Auto-Insurance Fraud Claim Detection by using Ensemble Learning Techniques

Wondemagegn Tilahun Kebede

A thesis Submitted to

Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2022

Adama, Ethiopia

Auto-Insurance Fraud Claim Detection by using Ensemble Learning Techniques

Wondemagegn Tilahun Kebede

Advisor: Bahiru Shifaw (Ph.D.)

A thesis Submitted to

Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2022

Adama, Ethiopia

APPROVAL PAGE OF MSC. THESIS

I hereby certify that the recommendation and suggestion given by the review committee are appropriately incorporated into the final thesis entitled **"Auto-Insurance Fraud Claim Detection by using Ensemble Learning Techniques"** by **Wondemagegn Tilahun**.

Bahiru Shifaw (Ph.D.) _____ _____

Major Advisor/Supervisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by **Wondemagegn Tilahun** have read and evaluated the thesis entitled "**Auto-Insurance Claim Fraud Detection by Using Ensemble Learning techniques"** and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

_____ _____ _____

Chairperson Signature Date

_____ _____ _____

Internal Examiner Signature Date

_____ _____ _____

External Examiner Signature Date

Final approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____ _____ _____

Department Head Signature Date

_____ _____ _____

School Dean Signature Date

_____ _____ _____

Office of Postgraduate Studies, Dean Signature Date

DECLARATION

I declare that this thesis entitled "**Auto-Insurance Fraud Claim Detection by using Ensemble Learning Techniques"** is my own work and has not been submitted to any university for similar purpose. The references used in this thesis are duly recognized by proper citations.

Wondemagegn Tilahun Kebede _____ _____

Name of student Signature Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled **"Auto-Insurance Fraud Claim Detection by using Ensemble Learning Techniques"** prepared under my guidance by **Wondemagegn Tilahun** submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science and Engineering. Therefore, I recommend the submission of revised version of the thesis to the department following the applicable procedures.

Bahiru Shifaw (PhD.)	_____	_____
Major Advisor/Supervisor	Signature	Date

ACKNOWLEDGEMENT

First and foremost, I thank the Almighty God for allowing me to complete this study. Second, I would like to thank Wolaita Sodo University and Adama Science and Technology University's Department of Computer Science Engineering for their assistance in adjusting, scheduling, and providing the opportunity to study here and complete this thesis work.

I am extremely grateful to my advisor, Bahiru Shifaw (Ph.D), for his approachable demeanor, invaluable advice, critical comments, and commitment to guide, as well as continuous support, which greatly improved this thesis work. I am grateful to Dr. Mesfin Abebe, the SIG supervisor and PG coordinator, for his ongoing commitment to guiding and supporting this research from its inception to completion.

I would like to thank the members of the CSE department's Smart Software SIG. Thank you to all of my master's thesis classmates for your cherished time spent together, ideas, and support until the end of my thesis. Special Thanks go to Ermiyas Nigatu who helped me during the time when my laptop broke by giving me his own laptop. Last but not least, I would like to acknowledge my family's support through this work; without them, this might not have been possible, and I am grateful for their support and advice in my life.

TABLE OF CONTENTS

ACKNOWLEDGEMENT iv
LIST OF TABLES ix
LIST OF FIGURES x
LIST OF EQUATIONS xii
LIST OF ACRONYMS AND ABBREVIATIONS xiii
ABSTRACT xv
CHAPTER ONE 1
1INTRODUCTION 1
1.1. Background of the study 1
1.2. Motivation 2
1.3. Statement of Problem 3
1.4. Research Question 3
1.5. Objective 3
1.6. Significance of the study 4
1.7. Scope 4
1.8. Limitation of the study 4
1.9. Organization of the Study 5
CHAPTER TWO 6
LITERATURE REVIEW AND RELATED WORKS 6
2.1. Overview of auto insurance fraud detection 6
2.2. Auto Insurance Fraud characteristics 6
2.3. Two Classes of Auto Insurance Fraud 7
2.3.1. Hard Insurance Fraud 7
2.3.2. Soft Insurance Fraud 7
2.4. Auto insurance fraud indicators 8
2.5. Insurance Fraud detection techniques 9
2.5.1. Social Network Analysis (SNA) 9
2.5.2. Fraud detection predictive analytics for big data 9
2.5.3. Social Customer Relationship Management (CRM) 10

2.6. Machine Learning Techniques 10
2.7. Machine Learning in Auto-insurance fraud detection 10
2.8. Ensemble Learning in Auto-insurance fraud detection 10
2.8.1. Technical Classification of Ensemble Methods 11
2.9. Feature Selection Methods 13
2.10. Related works 14
CHAPTER THREE 18
METHODOLOGY 18
3.1. Dataset 18
3.2. Dataset Description 18
3.3. Dataset labelling for fraud_type 19
3.4. Data pre-processing 19
3.4.1. Data Cleaning 19
3.4.2. Handling Missing Values 20
3.4.3. Data Processing 20
3.5. Handling Categorical Data using data encoding 20
3.6. Data Normalization 21
3.7. Feature Selection 22
3.7.1. Sequential feature selection 22
3.7.2. Recursive Feature Elimination 22
3.7.3. Embedded Methods 23
3.7.4. Pearson Correlation 23
3.7.5. Chi-Squared 24
3.7.6. Feature Selection by Combining Multiple Methods 24
3.8. Model building based on ensemble machine learning 25
3.9. Model Evaluation 25
3.9.1. Performance evaluators for the ensemble learning 25
3.9.2. Classification Accuracy 26
3.9.3. Cross Validation 27
3.10. Working Environment 27

3.11. Implementation Environment 28
CHAPTER FOUR 29
PROPOSED MODEL 29
4.1. Overview 29
4.2. Proposed Auto Insurance fraud claim detection model 29
4.3. Auto Insurance Fraud claim detection Model Building 31
4.3.1. Stacking classifier for claim detection model 31
4.3.2. RF classifier for fraud type classification model 32
4.4. Model Evaluation and Testing 32
4.5. Creating Classification Report 32
CHAPTER FIVE 33
IMPLEMENTATION AND EXPERIMENTATION 33
5.1. Dataset Description 33
5.2. Pre-processing Implementation 36
5.2.1. Missing value implementation 36
5.2.2. Handling Categorical Data (data encoding) implementation 37
5.2.3. Data Normalization 38
5.3. Implementation of Feature selection 38
5.3.1. Implementation of Sequential Feature Selector 38
5.3.2. Implementation of Superior Feature Selector 38
5.4. Splitting the dataset 39
5.5. Claim and fraud type classification models Implementation 39
5.5.1. Ensemble model implementation for both models 40
5.6. Model Testing and Evaluation 41
CHAPTER SIX 42
RESULTS, EVALUATION AND DISCUSSIONS 42
6.1. Dataset Class Distributions Result 42
6.2. Missing values result 43
6.3. Feature Selection Results for the claim detection model 43
6.4. Model Building for Proposed Models 44

6.4.1. Model building of claim detection model....44
6.4.2. Model building of fraud type classification model....44
6.5. Model Evaluation Results for claim detection model....45
6.5.1. Experiment 1 Models Evaluation Results on Original Dataset....45
6.5.2. Experiment 2 Models Evaluation Results after using SFS....47
6.5.3. Experiment 3 Models Evaluation Results after superior feature selection....48
6.5.4. Stacking classifier for the three Experiments....50
6.6. Model Evaluation Results for Fraud type classification model....51
6.6.1.Results on Fraud type classification model....51
6.6.2. HyperParameter Tuining....53
6.7. Comparison....53
6.8. Discussion....54
CONCULSION AND RECOMMENDATION....57
Conclusion....57
Findings from this research....57
Contribution....58
Recommendation....58
Future work....58
REFERENCES....59
APPENDIXES....I
Appendix A: Dataset Features....I
Appendix B: Sample Codes....V

LIST OF TABLES

Table 2.1 Auto insurance Fraud Indicators 8
Table 2.2: Summary of related works. 16
Table 3.1 Tools and Python Packages Used During the Implementation 28
Table 5.1 Dataset Features Description 34
Table 5.2 Sample Data 36
Table 6.1 Using fillna the missing value problem is solved 43
Table 6.2 Top features using sequential feature selection 43
Table 6.3 Models Accuracy of the Auto insurance claim before feature selection 45
Table 6.4 Models Accuracy of the Auto insurance 47
Table 6.5 Models Accuracy after superior feature selection 48
Table 6.6 Comparing all three Experiments in single table 50
Table 6.7 10-Folds Cross Validation accuracy (%) for fraud type classification model 52
Table 6.8 Hyperparameter tuning for claim detection model 53
Table 6.9 Hyperparameter tuning for fraud type classification model 53

LIST OF FIGURES

Figure 2.1 Random Forrest classifier.....11
Figure 2.2 Bagging classifier 12
Figure 2.3 Stacking classifier 12
Figure 2.4 Boosting classifiers 12
Figure 3.1 Research Design 18
Figure 3.2 Preprocessing steps.....19
Figure 3.3 The sequential forward selection 22
Figure 3.4 superior Feature selection by combining Multiple models.....24
Figure 3.5 Confusion Matrix with Binary-Classes 25
Figure 3.6 Confusion Matrix with 4-Classes.....26
Figure 4.1 workflow diagram of proposed model for auto insurance fraud detection.....29
Figure 4.2Proposed model for Auto Insurance fraud claim detection 30
Figure 4.3 Model Building Flow Diagram 31
Figure 4.4 proposed Stacking model for claim detection model 31
Figure 4.5 proposed RF model for fraud type classification model 32
Figure 5.1 Code snippet for importing multiple libraries 33
Figure.5.2 importing auto insurance claim dataset 36
Figure 5.5.3 checking for missing values in the Auto insurance claim dataset.....36
Figure 5.4 Missing Value Treatment using fillna 37
Figure 5.5 Frequency Encoding implementation 37
Figure 5.6 mean target encoding implementation.....37
Figure 5.7 label encoding implementation 37
Figure 5.8 Target guided ordinal Encoding implementation 38
Figure 5.9 ordinal encoding implementation 38
Figure 5.10 Min-Max Scaler.....38
Figure 5.11 Sequential Forward Selector for claim detection model 38
Figure 5.12 Sequential Forward Selector of fraud type classification model.....38
Figure 5.13 superior feature selection code snippet for claim detection model 39
Figure 5.14 Splitting the dataset claim detection model 39
Figure 5.15 Spliting the dataframe for fraud type classification model.....39
Figure 5.16 stacking classifier code snippet for claim detection model.....40
Figure 5.17 Code Snippet for Random Forest.....40
Figure 5.18 Code snippet for Ada Boosting 40
Figure 5.19 code snippet for bagging classifier 40
Figure 5.20 code snippet for XGBoosting classifier.....41
Figure 5.21 applying 10-fold cross validation 41
Figure 5.22 code snippet of classification report and confusion matrix 41
Figure 6.1 target class division in the claim dataset.....42
Figure 6.2 target class division in the fraud type data frame.....42
Figure 6.3 The 20 most important features based on superior feature selection.....44
Figure 6.4 Confusion Matrix before feature selection.....46
Figure 6.5 Confusion Matrix after using SFS.....47

Figure 6.6 Confusion Matrix after superior feature selection.....49
Figure 6.7 Accuracy of Stacking classifier for the three Experiments 51
Figure 6.8 Confusion Matrix for the Ada boost and RF for fraud type classification model..51
Figure 6.9 Comparison of performance metrics for the fraud type classification model.....52

LIST OF EQUATIONS

Equation 3.1 min-max normalization 21
Equation 3.1 Pearson's correlation Formula.....24
Equation 3.2 Chi-Squared Formula 24
Equation3.3 Accuracy Formula 26
Equation 3.4 Precision Formula.....26
Equation 3.5 Formula of recall 26
Equation 3.6 Specificity formula 27
Equation 3.7 F1 Score Formula 27

LIST OF ACRONYMS AND ABBREVIATIONS

Abbreviation	Definition
Ada boosting	Adaptive Boosting
AE	Autoencoder
AI	Artificial Intelligence
AUC	Area Under the ROC Curve
CART	Classification and Regression Trees
DL	Deep learning
DNN	Deep Neural Network
DT	Decision Tree
ELT	Ensemble Learning Techniques
ETB	Ethiopian Birr
FN	False Negatives
FP	False Positives
IFS	Intrinsic Feature Selection
KNN	K-nearest Neighbor
LASSO	Least Absolute Shrinkage and Selection Operator
LR	Linear Regression
ML	Machine Learning
NB	Naïve Bayes
NLP	Natural Language Processing
RF	Random Forest

RFE	Recursive Feature Elimination
RF	Random Forest
RF-MDA	Random Forest –Mean Decrease in Accuracy
RF-MDI	Random Forest –Mean Decrease in Impurity
ROC	Receiver Operator Characteristic Curve
SMOTE	Synthetic Minority Oversampling Technique
SFS	Sequential Feature Selector
SVM	Support vector machine
TN	True Negatives
TP	True Positives
UK	United Kingdom
USA	United States of America
VAE	Variational Autoencoder
XGBoost	Extreme Gradient Boosting

ABSTRACT

Insurance companies have recently suffered from dishonest insured claims. As a result, auto insurance fraud has become a major source of concern for both businesses and consumers. Multiple researchers have proposed auto insurance fraud claim detection systems that are implemented using machine learning techniques, but they lack a good feature engineering approach, they use too small features, and they do not identify which type of auto insurance fraud has occurred. The goal of this thesis is to develop an auto insurance fraud claim detector that improves the feature engineering approach by introducing a fraud type-based ensemble classifier. To achieve this goal, several ensemble learning models have been experimented with and tested with different methods to handle the auto-insurance claim dataset, which has 11,210 rows and 40 features and is publicly available. Different data preprocessing techniques were applied, such as data cleaning, missing value handling, data encoding, data scaling, and feature selection methods. Our study has proposed combine feature selection method, one detection model and one classification model with three functions. The first one is determining whether a claim is fraud or notfraud called a claim detection model, in which three experiments were carried out. Experiment 1 on the original dataset with no feature selection, Experiment 2 with sequential feature selection, and Experiment 3 with superior feature selection with ensembles such as RF, bagging classifiers, XGBoost, AdaBoost, and stacking classifiers. To avoid overfitting, the proposed model is tested using K-fold cross-validation. Other evaluation metrics such as precision, recall, f1 score, and specificity are also used. The stacking classifiers with sequential feature selection produced the highest accuracy of 99.31% and an F1 score of 0.98 across all three experiments. The second one determines the type of fraud, which has five classes called the fraud type classification model. The RF classifier with sequential feature selection produced the highest precision, recall, and f1-score of 97%. The third function is feature selection using a combined feature selector, which combines and votes on the top selected features that increase the classification performance of the auto insurance claim detection model. In the future, text analytics and image processing, as well as video processing, should be considered.

Key words: Auto insurance, Fraud claims, Ensemble Learnings, Stacking, Sequential Feature Selection, combine feature selection

CHAPTER ONE
INTRODUCTION

1.1. Background of the study

In this world, life is full of risks and surprises. As we all know risk is inevitable that is where the concept of insurance comes in. According to (Pfeffer, 1956) "A technique for decreasing the risk of one party, termed the insured, through the transfer of specific risks to another party, called the insurer, who promises a restoration, at least in part, of economic losses sustained by the insured," is called insurance.

Insurance fraud occurs when individuals attempt to get profit by violating the terms of the insurance agreement. Insurance fraudsters try to cause losses or damage rather than joining others who have no losses but want to be protected in case an unknown event occurs. As (Viaene & Dedene, 2004) described fraud can occur at any stage of an insurance transaction and can be committed by individuals seeking insurance, third-party claimants, policyholders, and professionals who provide claimant services.

Fraud accounts for 10% of the property casualty insurance industry's incurred losses and loss adjustment expenses, or approximately $3 billion per year.[1] The FBI estimates a $40 billion loss each year. We can definitely say it is a huge amount of money which is under scrutiny.

According to an indication of claims leaking in Ethiopia, claim costs may rise. For instance, the net claims of the Ethiopian insurance sector climbed from ETB 1.9 billion in 2013 to ETB 2.5 billion in 2015. Motor insurance claims, which can make up as much as 70% of the industry's claim costs, are subject to the possibility of being submitted fraudulently.[2]

(Morley et al., 2006) explains that AI algorithms can analyze large amounts of data quickly to find patterns. Then they can spot anomalies that don't fit the patterns. For example, by comparing new claims to existing data, AI can help detect claims values that are unusually high. Some companies use multiple types of insurance claim detection method to avoid insurance fraud.

Traditional and data analytics approaches are the two types. For the traditional Insurance companies and their field agents or loss assessors were assigned to fraud detection in insurance

[1] The Power of Analytics for Insurance Fraud Detection - Insurance Insights (lexisnexis.com)
[2] Insurance Fraud: – Ethiopian Business Review

claims in the traditional way. To detect a scam and authenticate or reject the claim, they relied on limited facts and often their intuition. They would normally use the existing infrastructure to verify the authenticity of documents, personally inspecting damaged premises or cars, and occasionally enlisting the assistance of professionals, surveyors, forensic accountants, or, in today's world, cyber specialists. Data analytics play a vital role in fraud detection. Classification modeling can be used to investigate fraud cases, filter obvious situations, and refer low-incidence cases for further investigation. Analytics aid in the development of a fully global perspective of the company's anti-fraud initiatives. In order to integrate data analytics is crucial. Unstructured data can be made more valuable with the use of analytics. This research proposes to use ensemble learning approaches to detect auto insurance fraud claims with the help of data analytics and AI.

1.2. Motivation

Machine learning and deep learning can solve a wide range of problems in a different of ways. As (Artís et al., 2002) explains that the biggest problem in the insurance industry is fraud claims. Detecting and preventing fraud is very expensive, time consuming and boring. According to (Abdallah et al., 2016) Insurance companies have recently suffered from dishonest insured claims. As a result, auto insurance fraud has become a major source of concern for both businesses and consumers. According to (Artís et al., 2002), the most common type of insurance fraud claim is auto insurance fraud. According to (Brockett et al., 1998) the most common fraudulent behavior in this area is staged vehicle fraud and rental car fraud. An increase in claim costs could be a sign that claims are leaking. For instance, the net claims of the Ethiopian insurance sector climbed from ETB 1.9 billion in 2013 to ETB 2.5 billion in 2015. Motor insurance claims, which can make up as much as 70% of the industry's claim costs, are subject to the possibility of being submitted fraudulently. As states Every year, insurance fraud costs the insurance industry billions of dollars, which hurts insurance company earnings and growth, as well as national economic growth, and insurance companies pass on the entire loss of fraud risk to the consumer by raising premium rates.[3] The reasons stated above are motivation for doing this research on auto insurance claim process to expose fraudulent claims thereby, helping the insurance industry to save human resource, time and money.

[3] Insurance Fraud: – Ethiopian Business Review

1.3. Statement of Problem

In today's world, fraudsters are more dynamic, and fraud is difficult to detect and prevent. According to (Viaene & Dedene, 2004), auto-insurance fraud is the most prevalent and dynamic in the insurance industry. Every day, new and inventive fraud methods are developed. Major auto insurance frauds that are causing problems for the industry are: staged auto accidents and false claims of injury; false reports of stolen vehicles; false claims that an accident occurred after a policy or coverage was purchased; false claims for damage that already existed; and claimants who concealed that a person excluded from coverage by their policy was driving at the time of the accident. To avoid this, fraud must be detected. Insurance fraud and, more broadly, insurance abuse endanger not just the insurer's profits but also the insurance industry's value chain and may seriously damage pre-existing social and economic systems. They are thought to significantly raise the cost of certain types of insurance (e.g., automobile, fire, and health insurance). (Viaene & Dedene, 2004) argue that they eventually become a threat to the very principle of solidarity that keeps the insurance concept alive. This study's goal is to develop a model that can detect auto insurance fraud. Fraud is unethical, and it costs the company money. This research is expected to demonstrate the use of ensemble learning techniques in reducing losses for the insurance company by building a model that can classify auto insurance fraud. Of course, fewer losses equate to more earnings.

1.4. Research Question

Following the completion of this research, the following research questions must be answered:

- What are the major attributes or features that allow one to predict the legal status of claim and types of auto insurance fraud claims?
- How to develop an auto insurance fraud claim detection model using ensemble machine learning techniques?
- Which ensemble learning technique provides the most accurate auto insurance fraud claim detection from the selected ensemble learners?

1.5. Objective

1.5.1. General Objective

The general objective of this study is to develop an auto insurance fraud claim detector that improves the feature engineering approach and introduces a fraud type-based ensemble classifier.

1.5.2. Specific Objective

- To gather and prepare datasets for training and testing ensemble models.
- To design a model that classifies auto insurance claims and auto insurance fraud types.
- To detect fraud from legal claims during the auto insurance claim process.
- To test the model on untrained data.
- To evaluate the performance of the designed model for better result.
- To state recommendation and conclusion based on the result of the experiment.

1.6. Significance of the study

The following are the significance of this study.

- Reliable design of an auto insurance fraud detection model for future researchers.
- Fast auto insurance claim process.
- Saving human resources, time, and money for the insurance industry improve consumer and provider satisfaction.
- Provide evidence for crimes done by the fraudsters for criminal prosecution when the type of identified fraud is hard fraud, because staged and vehicle theft frauds are criminal cases, so the claim will be evidence in the courtroom.

1.7. Scope

The scope of this study is

- Detect claim fraud for auto insurance claims; it classifies them as fraud or legal.
- Classify the type of auto insurance fraud that occurred: it classifies as Staged, overreporting, False claims, underwriting.
- The focus is on auto insurance fraud; it does not consider health, life, or other insurance types.

1.8. Limitation of the study

This study has the following limitations:

- The model will detect fraud from the prepared dataset do not detect from raw data.
- The physical damage of the automobile is not processed using image processing.
- Text and image analytics could be in hand and previous history of the insured are not considered.

1.9. Organization of the Study

This research work is organized into the following chapters.

Chapter 1 Introduction: explains the introduction and covers the motivation, problem statement, study objective, scope, limitation, and application results, as well as organization of the study.

Chapter 2 Literature Review: states the literature review to this study, reviews background information on several key concepts about Auto insurance fraud problems, and algorithms used in Auto insurance fraud detection and classification: feature selections and machine learning based Ensemble classifiers, and finally, the related works are discussed.

Chapter 3 Methodology: states, research methodology used in this study, methods used to preprocess the dataset, methods used to develop Auto insurance fraud claim detection system which are data preprocessing, feature selection methods, and ensemble machine learning classifier algorithms, and finally, it states the evaluation methods.

Chapter 4 Proposed Solution: discusses the proposed solution for Auto insurance fraud claim detection, which is the proposed data preprocessing, feature selection, machine learning based Ensemble classifiers, and evaluation methods.

Chapter 5 Implementation, shows the implementation and experimentation of the proposed solution. Including working environment, dataset description, also the implementation of preprocessing, feature selection implementation, model implementations, and evaluation of the models

Chapter 6: Results and Discussion: Result and Discussion part discusses the result of the dataset class distribution, feature selection, and discusses the major results obtained by comparing the models based on their performance.

Last chapter Conclusion and recommendation: describes conclusion and forwards recommendation and feature works for further investigation of this research study.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORKS

In this chapter, the relevant literature related to the main objective of the paper is reviewed to clearly state and further explore the research problem by providing a technical review on Auto-insurance Fraud detection and existing method for detection.

2.1. Overview of auto insurance fraud detection

According to (Milton, 2022) Auto insurance fraud is a serious crime that involves an individual or organization that deceives a carrier about a personal or commercial motor vehicle claim. They commit these acts for financial or other gains. One goal of data analytics is detecting patterns using machine learning techniques.

2.2. Auto Insurance Fraud characteristics

The most challenging aspect of detecting auto insurance fraud is that it is extremely difficult to do so since it is dynamic and imaginative. It gets harder as a result. The first step in examining any auto insurance claim is to separate it into the following five key fraud indicators: It is required for almost all claims (Benedek & László, 2019; Müller, 2013) . Those are:

- **Policy Holder characteristics**: policyholder is the insured person protected in case of a loss or claim it has the following age, sex, education level, hobbies, past number of claim and residence area (city, state, zip) so on.
- **Vehicle Characteristics:** age of car, auto make, auto model, year of manufacture, vehicle value, commercial, transport and other.
- **Policy Characteristics** the following are common auto liability coverage, uninsured and underinsured motorist coverage, comprehensive coverage, collision coverage, medical payments coverage, personal injury protection and more.
- **Claim characteristics** vehicle claim, injury claim, property claim
- **Incident characteristics** incident type, place state city, time, Collison type (rear front or side), witness available involved, authorities contacted (ambulance, lawyer or fire department involvement), police report, incident severity (minor, major and total loss), cars involved in the incident and others.

So, the claim will be examined and classified as fraud or non-fraud using the five primary features listed above.

2.3. Two Classes of Auto Insurance Fraud

Misrepresenting facts on insurance applications is an example of automotive insurance fraud (Dossett, 2022). It can also include staging accidents and filing claims for injuries that never occurred. Others spread false information about stolen vehicles.

2.3.1. Hard Insurance Fraud

Hard insurance fraud is easier to detect than soft insurance fraud. Hard insurance fraud is the deliberate staging of an automobile accident in order to defraud a carrier. To make a false claim for money, a criminal may stage injuries, arson, or theft.

- **Staged Auto Accidents** – These occur when a driver forces another driver into a collision. They use a phony witness to convince the cops that the victim is to blame.
- **Planned vehicle theft:** is when a motorist hires an accomplice to steal a car. After that, the couple sells the car. Before submitting a claim to the insurance company, they could also destroy it or disassemble it for parts.

2.3.2. Soft Insurance Fraud

Soft insurance fraud is a common criminal act in which individuals exaggerate parts of a legitimate claim. Personal data is used to calculate premium rates by an insurer underwriting your car insurance policy. An applicant provides false information about their residence or other information in order to obtain a lower rate on their auto insurance policy. Soft fraud raises policyholders' premiums more than hard fraud. The most well-known soft fraud examples:

- **Over-reporting** A driver might inflate the value of stolen goods. As part of the new insurance claim, other factors can include dented or previously damaged equipment.
- **Underwriting fraud** Providing false documentation for lower premiums Another type of fraud is providing false information to an insurance carrier. To obtain lower rates, a person may Misrepresenting or omitting information on a car insurance application.
- **False claims fraud** – Here, policyholders either file false claims or ask for money for things that are not covered by the policy. The tactics used by car insurance scammers may include making more than one claim for the same damage or filing a claim for repairs that were never made. Other strategies include including prior problems with their vehicle in their accident claim.

2.4. Auto insurance fraud indicators

The following table lists the most recognizable traits of frauds based on (Accidents, n.d.; *Fraud Indicators "Red Flags" – Organized Fraud*, n.d.; Insurance Fraud Bureau, 2012; Müller, 2014; Tennyson & Salsas-Forn, 2002). There is a very high possibility that the insurance claim is fraudulent if those signs are present.

Table 2.1 Auto insurance Fraud Indicators

Fraud Indicators	**Claim characteristics**	**Policy Holder characteristics**	**Incident characteristics**	**Vehicle Characteristics**
Staged	High claim amount	Age 20-45 Male Young policy holders	Witness little to zero Rear or side collision Minor damage or total loss No police report available fire report Vehicle lost	New or rental car for the victim side Old car for perpetuator
Overreporting	High claim amount	Age 20-45 Male Young policy holders	Minor damage No police report available	Windshield broken, Glass broken, old car
Underwriting	High claim amount	Age 20-45 Both Gender Young policy holders	Police report available Witness available	Old car Faulty windshield Faulty airbag
False claims	Low claim amount Repetitive claim False registration	Age 20-45 Both Gender Young policy holders	Minor damage No police report available Single vehicle involvement	Any car Faulty windshield Faulty airbag

According to (Benedek & László, 2019; Insurance Fraud Bureau, 2012; Müller, 2013) middle-aged individuals who prefer to drive high-valued cars and having signed a leasing contract more often than their honest counterparts. (Artís et al., 2002)With regard to their driving behavior, individuals engaging in fraudulent activities prove to be rather experienced and safe drivers. Another instance of auto insurance fraud is making tiny loss claims with the goal of avoiding detection. Similar to this, they attempt to avoid attracting attention by making claims after the insured loss occurrence has already happened. Type of damage is another key factor in indicating the auto insurance fraud(Tennyson & Salsas-Forn, 2002). According to (Artís et al., 2002; Müller, 2013) Older vehicles are more likely to be the subject of fraud since policyholders may view their money value as additional when purchasing a new vehicle.

2.5. Insurance Fraud detection techniques

2.5.1. Social Network Analysis (SNA)

The SNA method uses a hybrid strategy to find fraud. The hybrid approach incorporates organizational business rules, statistical techniques, pattern analysis, and network linkage analysis (Šubelj, Furlan, & Bajec, 2011). When looking for fraud in link analysis, look for clusters and how they relate to one another. A model can incorporate data from various sources such as records, judgments, and bankruptcies.

2.5.2. Fraud detection predictive analytics for big data

Data mining - (Gao & Ye, 2007)To find patterns and detect fraud, data mining for fraud detection and prevention classifies and segments data groups in which millions of transactions can be performed.

Neural networks - Suspicious patterns are discovered and used to detect future occurrences.

Machine Learning - analytics for fraud Machine Learning detects fraud characteristics automatically.

Pattern recognition - (Gao & Ye, 2007) detects patterns or clusters of suspicious behavior. Data mining and machine learning are used in fraud analytics use cases such as payment fraud analytics. Predictive analytics examines big data for fraud using text analytics and sentiment analysis. Data mining uncovers meaningful patterns, transforming raw, large datasets into useful information. The information is then submitted to algorithms by Machine Learning.

2.5.3. Social Customer Relationship Management (CRM)

Social CRM is a program for detecting fraud. It is critical for insurance companies to connect social media to their CRM these days. Connecting social media to CRM improves customer transparency. Customers gain trust in the organization as a result of this transparency. This customer-centric ecosystem benefits the business significantly while also ensuring that the customers are in control. (M, Y, Amrutha, Harsha, & M, 2020).

2.6. Machine Learning Techniques

Machines may now automatically learn from data to create predictions without human intervention. Machine learning use algorithms to recognize patterns and learn through an iterative process. Instead of depending on any preconceived equation that might serve as a model, ML algorithms use computation techniques to learn directly from data. Some of the most well-liked varieties of machine learning include the following: ML techniques include supervised, unsupervised, semi-supervised, and reinforcement learning. There are further machine learning algorithms that are capable of learning Such as ensemble learning.

2.7. Machine Learning in Auto-insurance fraud detection

Nowadays, machine learning is employed for almost all tasks that need detection. Machine learning applies parts of AI to analyse massive, labelled data sets, allowing systems to learn from experience without the need for new programming in order to detect insurance fraud. In the strategies described here, ML can enhance fraud detection methods. Data processing takes place quickly highlights the potential connections between numerous aspects that the human eye is unable to see uses a variety of data mining approaches to enable the discovery of fresh fraud schemes. Classification algorithms were more accurate in dealing with this issue. Machine Learning algorithms such as KNN, DTs, SVMs, and RF have been used on many previous researches for detecting fraudulent insurance claims.

2.8. Ensemble Learning in Auto-insurance fraud detection

Ensemble learning is a broad machine learning meta-approach that seeks to improve predictive performance by combining predictions from multiple models.(Bühlmann, 2012; Dietterich, 2000) Each of which solves the same original task in order to produce a better composite global model with more accurate and reliable estimates or decisions than a single model can.(Last et al., 2006). Health and auto insurance fraud claims are just two of the issues that ensemble

learning has been employed for. They have been used in different way in solving auto insurance frauds. The types of ensemble methods in machine learning.[4]

1. Sequential Methods: There are sequentially generated base learners that contain data dependency. Every other data in the base learner is dependent on previous data in some way. As a result, the previously mislabeled data is tuned based on its weight to improve the overall system performance. **Example**: Boosting

2. Parallel Method: The base learner is generated in parallel order, with no data dependency. Every piece of data in the base learner is generated on its own. **Example**: Stacking

3. Homogeneous Ensemble Methods: is a mix of the same kinds of classifiers. However, each classifier's dataset is unique. This will make the combined model work more precisely after the results of each model are aggregated. This ensemble method can handle a large number of datasets. The feature selection method in the homogeneous method is the same for different training data. It is computationally costly. **Example:** bagging and boosting

4. Heterogeneous Ensemble Methods: is the combination of different types of classifiers or machine learning models in which each classifier built upon the same data. This method works well with small datasets. For the same training data, the feature selection method differs in heterogeneous. This ensemble method achieves its overall result by averaging the results of each combined model. **Example**: Stacking

2.8.1.Technical Classification of Ensemble Methods

- **Random Forest Classifier** is a classifier that uses multiple decision trees on different subsets of a given dataset. Rather than relying on a single decision tree, the random forest uses the predictions from each tree to predict the final output based on the majority vote of predictions. Random forest (Breiman, 2001) is probably the most famous bagging method.

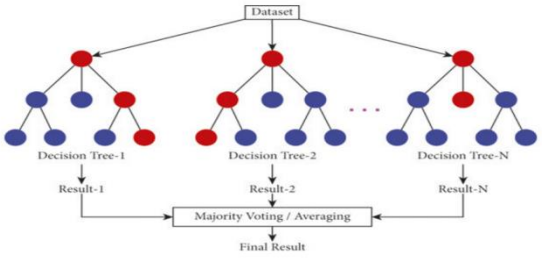


Figure 2.1 Random Forrest classifier

[4] Ensemble Methods in Machine Learning | 4 Types of Ensemble Methods (educba.com)

- **Bagging** (Breiman, 1996) also known as Bootstrap aggregation involves fitting many decision trees on different samples of the same dataset and averaging the predictions., It is used to deal with bias-variance trade-offs and reduces a prediction model's variance.

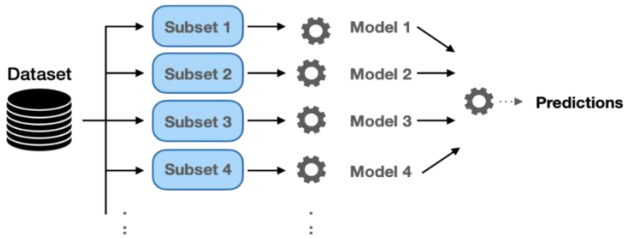


Figure 2.2 Bagging classifier

- **Stacking (Stacked Generalization)** (Wolpert, 1992) involves fitting many different models' types on the same data and using another model to learn how to best combine the predictions or the predictions of multiple classifiers are used as new features to train a meta-classifier.

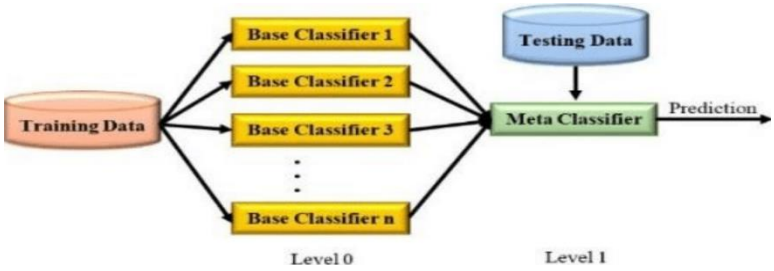


Figure 2.3 Stacking classifier

- **Boosting(Bootstrap Aggregating)**(Breiman, 1996)) involves adding ensemble members sequentially that correct the predictions made by prior models and outputs a weighted average of the predictions.

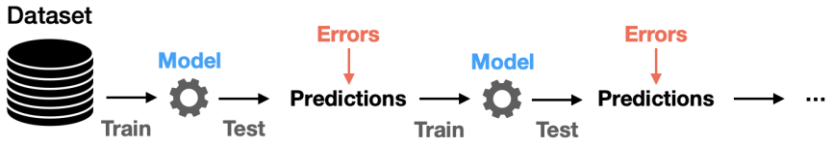


Figure 2.4Boosting classifiers

- **XGBoost** Extreme Gradient Boosting (XGBoost), a scalable and extremely accurate gradient boosting method, pushes the boundaries of computational power. Parallelization, tree trimming, hardware optimization, regularization, sparsity awareness, weighted quartile sketch, and cross validation are all combined in this method.

- **Adaboost** Adaptive Boost is a technique in Machine Learning used as an Ensemble Method. It can be used to improve the performance of machine learning algorithms. It is used best with weak learners, and these models achieve high accuracy above random chance. The most common algorithms with AdaBoost are decision trees with level one or decision stumps.

2.9. Feature Selection Methods

According to (García et al., 2015; Rückstieß et al., 2011; Salappa et al., 2007) Feature selection is the process of selecting the most appropriate subset of features to describe a given classification task. The goal of applying feature selection is to avoid noisy data, inconsistent data, and reduce the dimension of the dataset to get an efficient and accurate result.

Feature Selection: Feature subset selection is a method of selecting the subset feature that contributes the most to the prediction variable or output based on statistical analysis of the original dataset. It is a preprocessing technique that finds a minimum subset of features by removing as many irrelevant and redundant features as possible from the original data (Phyu & Oo, 2016). Filter approaches, wrapper approaches, and embedded approaches are the three types of feature selection approaches according to (Hoque, Bhattacharyya, & Kalita, 2014).

Filter Methods: It is the method of selecting and filtering features before implementing any machine-learning model and only after the best features are found, modeling algorithms can use them. Filter methods use statistical measures to score the dependencies between input features and target class so that most relevant features can be filtered. (Jović, K. Brkić, & N. Bogunović, 2015).

Wrapper Methods: The method considers the machine-learning model to randomly select a relevant feature and train the model with this feature subset to assess the model's accuracy on the dataset. The wrapper method takes considers feature independencies, the interaction between each feature, and model selection (Beniwal & Arora, 2012)

Embedded Methods: This method is used to select the best features during the modeling algorithm execution by interacting with the learning models at a lower computational cost than the wrapper approach. This method considers feature dependencies and searches locally for features that allow better local discrimination and it uses the independent criteria to recognize to choose the final optimal subset from among the optimal subsets for a known cardinality (George, Ron, & Karl, 1994)

Superior Feature Selection Method: Several feature subsets are created and then combined into a single subset. or a Hybrid Feature Selection Method for a combination of feature selection algorithms for a classification problem(Cateni et al., 2014; Rokach et al., 2006; Skurichina & Duin, 2005)

2.10. Related works

The study done by (Alamir et al., 2021) the aims were to develop a machine learning model that classifies and predicts the status of motor insurance claims using a machine learning approach. The missing value ratio, Z-Score, encoding methods, and entropy were used in this research as data set preparation strategies. K- Fold cross validation procedures were used to separate the final preprocessed data sets into training and testing sets. Finally, Multi Class-Support Vector Machine (SVM) and Random Forest (RF) were used to develop the prediction model (SVM). To assess the effectiveness of the models, RF, and Multi-Class SVM classifiers, accuracy, precision, recall, and F-measures were utilized. The model's prediction accuracy is 98.36 % by RF classifiers and 98.17 % by SVM classifiers, respectively, for predicting the status of a motor insurance claim. RF classifier hence outperforms Multi-Class Support Vector Machines.

For health insurance fraud detection, (Kunickaitė et al., 2020) use ensemble machine learning (Decision Trees, Baggsing, Random Forests, and Boosting). The model's performance was evaluated using accuracy, error rate, sensitivity, and specificity. After data cleaning, they used 2,662,308 records, leaving 1,997,076 records in the data set for subsequent processes. The following are the findings of the experiments: The bagging process has the highest accuracy (93.87 %). The decision tree predicted sensitivity of 99.46%. Boosting is the strategy that performs the best (92.91%). The bagging process has the highest accuracy (93.87 percent). Based on the error rate, bagging is the best performing approach (6.13%).

The research done by (Pranavi, 2020) use two machine learning algorithms to detect automobile insurance fraud. For the accurate identification of insurance fraud, they use the Random Forest and KNN algorithms. Performance is calculated by using the confusion matrix.

The study proposed by (Simfukwe et al., 2015) compares two techniques: artificial neural networks (ANN) and the Nave Bayes classifier. The applicability of Neural Networks and the Nave Bayes classifier to the dataset is discussed, and the results are compared and contrasted using a variety of performance measures such as ROC curves, Accuracy, AUC, Precision, and

Sensitivity. The Nave Bayes classifier performed slightly better than the artificial neural network.

To improve the overall performance of prediction models, (Kalwihura & Logeswaran, 2020) proposes a data pre-processing technique, specifically a fraud behavior feature engineering approach. The assessed behavior is based on the RFM model, as well as an additional behavior analysis related to policy expiration. The feature engineering approach's high dimensionality issues and class imbalance issues are also addressed using an ensemble feature selection and modeling approach. The suggested strategy raises the F1-measure by 56.2 % as compared to earlier published state-of-the-art results.

(Caruana & Grech, 2021) They compare two techniques in this article: artificial neural networks (ANN) and the Nave Bayes classifier. The relevance of Neural Networks and the Nave Bayes classifier to the dataset is discussed, and the results are compared and contrasted using a variety of performance measures such as ROC curves, Accuracy, AUC, Precision, and Sensitivity. The Naïve Bayes classifier outperformed the artificial neural network slightly.

(Moon et al, 2019) In this study, parametric and nonparametric statistical learning algorithms are considered in order to reduce uncertainty and increase the likelihood of detecting appropriate claims. An important set of features for a model is determined by measuring variable importance based on observed claim characteristics via cross-validation and testing improvement in the accuracy with which automobile fraudulent claims are classified using the Akaike Information Criterion.

The paper proposed by (Hassan & Abraham, 2016) which is an innovative insurance fraud detection model based on ensemble combining classifiers on previously designed base-classifier models. The ten-fold cross validation method of testing is used throughout the paper. Its uniqueness stems from the use of several ensembles combining classifiers and comparing them to select the best model. A publicly available dataset for the detection of vehicle insurance fraud demonstrates that DTIFD outperforms all other proposed models, including ensembles of classifier-designed IFD models with good recall, but it is still the best.

The study done by (Hanafy & Ming, 2021) investigates how automotive insurance providers incorporate machine learning into their operations and how ML models can be applied to insurance big data. To forecast the likelihood of a claim, they employ a range of ML techniques,

including as logistic regression, XGBoost, random forest, decision trees, naive Bayes, and K-NN. They evaluate and contrast these models' efficacy as well. With accuracy, kappa, and AUC values of 0.8677, 0.7117, and 0.840, respectively, the results demonstrated that RF performed better than other approaches.

Table 2.2: Summary of related works.

Authors & Year	**Title**	**Gaps**	**Algorithms used**	**Best Algorithm**
(Pranavi, 2020)	Analysis of Vehicle Insurance Data to Detect Fraud using Machine Learning	➢ Only two ML algorithms were utilized. ➢ The used features were too small. ➢ Poor feature engineering methods ➢ Do not categorize the type of auto insurance fraud that has taken place. ➢ being of only binary class since fraud comes in several forms	RF and SVM	RF
(Hanafy & Ming, 2021)	Machine Learning Approaches for Auto Insurance Big Data	➢ Poor Feature Engineering Approach ➢ Too small features were used ➢ Being of only binary class because fraud has multiple types	RF, C50, XGBoost, J48, KNN, CART, Naïve Bayes	RF
(Caruana & Grech 2021)	Automobile insurance fraud detection	➢ Poor Feature Engineering Approach ➢ Too small features were used ➢ Do not classify which type of auto insurance fraud has occurred ➢ Being binary class because fraud has multiple types	ANN and Naïve Bayes	Naïve Bayes

(Moon et al. 2019)	A Classification modeling for Detecting Fraudulent Automobile Insurance Claims	➢ Poor Feature Engineering Approach ➢ small features were used ➢ Do not classify which type of auto insurance fraud has occurred ➢ Being binary class because fraud has multiple types	LR, LASSO, RF-MDI & RF-MDA	LASSO
(Hassan & Abraham, 2016)	Modeling Insurance Fraud Detection Using Ensemble Combining Classification	➢ Does not focus on the feature engineering approach. ➢ Do not classify which type of auto insurance fraud has occurred ➢ Being binary class because fraud has multiple types	Grading, Stacking and Vote	Grading
(Alamir et al., 2021)	Motor Insurance Claim Status Prediction using Machine Learning Techniques	➢ There are just two ML algorithms employed. ➢ Does not focus on the feature engineering approach. ➢ Do not classify which type of auto insurance fraud has occurred	RF and SVM	RF

Determining the sorts of fraud that took place throughout the claiming process and identifying fraudulent claims for auto insurance are the two main objectives of this study. The majority of the authors in the related work that are indicated in the above table explained that it is challenging to recognize and detect fraudulent behavior. The majority of the existing publications categorize auto insurance as being either legal or fraudulent; they do not take into account the type of fraud that occurs once auto insurance fraud is discovered. This study's findings ameliorate the shortcomings of earlier research, particularly the related work listed in the table above. The goal of this study is to develop an ensemble model that detect auto insurance fraud claim and type of fraud that happened.

CHAPTER THREE
METHODOLOGY

This chapter discusses the methodology and the process of developing auto insurance fraud claim detection using Ensemble learning based classifiers using auto insurance claim dataset.

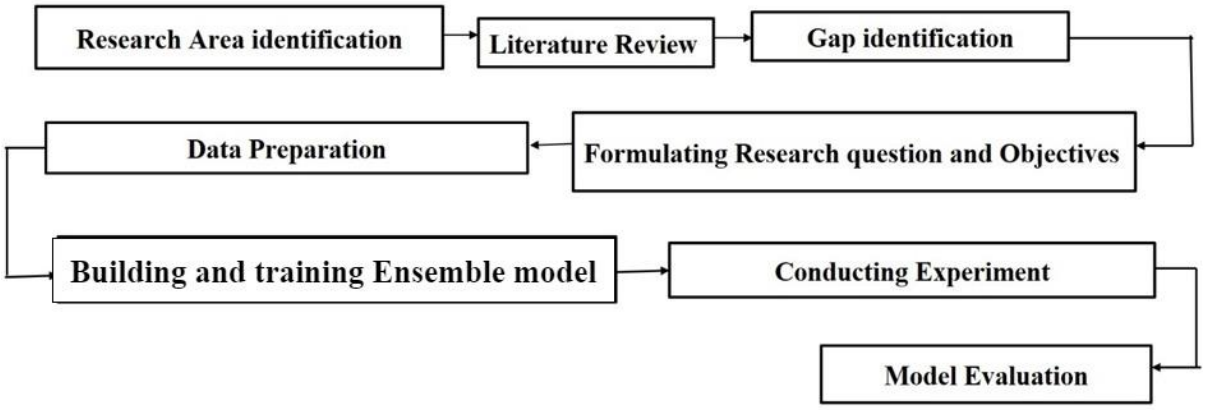


Figure 3.1 Research Design

3.1. Dataset

The most important thing in machine learning or deep learning is neither the algorithm nor the model; it is the quality of data that matters. So, understanding and collecting data is an important part of machine learning technology. Even if the model is good, it will not learn anything unless the data is good and valid.(Rosli et al., 2016)

3.2. Dataset Description

The datasets are taken from GitHub. The URL to the datasets are here: Auto-Insurance-Claims-Fraud-Detection-with-ML/insurance_claim_updated.csv at master · ezzaimsoufiane/Auto-Insurance-Claims-Fraud-Detection-with-ML · GitHub. and Auto-Insurance-Fraud-Claim-Detection/Insurance Fraud Detection.ipynb at master · adibhattar95/Auto-Insurance-Fraud-Claim-Detection · GitHub. The goal of this study is to create an ensemble model that can identify the type of fraud that occurred when it comes to auto insurance claims. The dataset originally had a binary class, but after examining the features of the frauds, a new column (fraud_type) was added with five classes. All analyses in this study are performed on this dataset which includes claims for a car insurance company in the United States the data set can be categorized as a global dataset with improvement of 1 column. The data set includes 10,211 individual claims and 1000 claim merged together for total of 11,210. There are two target

variable the first one is reported_fraud. If a claim is reported to be fraudulent, this variable is labeled Y; otherwise, it is labeled N. the next target variable is fraud_type which has 5 classes by name staged, overreporting, underwriting, False Claim, notfraud. Each claim in the data is described by 40 different attributes, which are represented as columns. The data consists of attributes of both numeric and categorical nature. Some examples of attributes are the age of the insured, the insured amounts and premiums, the job of the insured, the number of vehicles involved in the incident and the brand of the car for which the claim was made.

3.3. Dataset labelling for fraud_type

The dataset target-variable fraud_type is labelled based on table 2.1. which are auto insurance fraud indicators.

3.4. Data pre-processing

A data mining approach called preprocessing entails putting raw data into a format that may be used. Real-world data is typically riddled with inaccuracies and is inconsistent, lacking in specific behaviors or trends, and frequently incomplete. A tried-and-true technique for solving such issues is data preprocessing (Roy et al., 2018). As a result, particular procedures are followed and completed in order to transform the data into a manageable and tidy data set. This set of techniques is known as data preprocessing which are: import the libraries, import the dataset, check out the missing values, data normalization and see the categorical values, encoding categorical values to numerical values using feature selection and extraction methods, splitting the data-set into training and test.

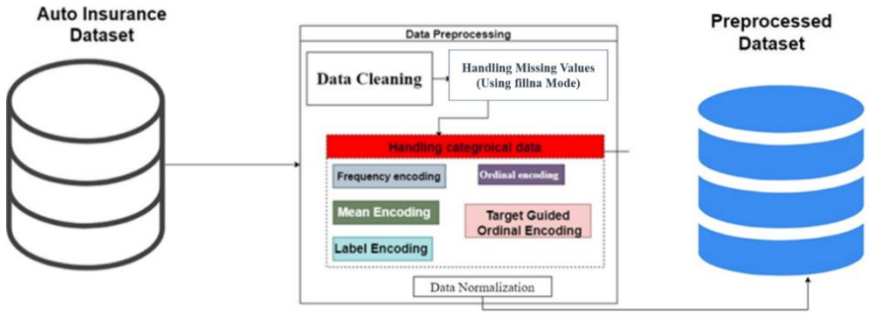


Figure 3.2 Preprocessing steps

3.4.1. Data Cleaning

Data cleaning is the process of editing, correcting, and structuring data within a data set. This includes removing corrupted or irrelevant data and formatting it in a computer-readable

language. D.ata cleaning technique used to authenticate data and ensure that it is of high quality.(Herbert & Wang, 2007). It entails dealing with missing data, noisy data, and so on.

3.4.2.Handling Missing Values

This situation occurs when some data is missing from the data. There are several approaches that can be taken to complete this task. Some results had missing values due to some mistakes while collecting data and some unfilled values of auto insurance claim data. The missing values have an impact on the accuracy of the classifier model. Missing values can lead to serious issues, whether it is poor representation of the overall dataset in terms of its distribution, or bias in the results obtained (Kang, 2013).

3.4.3.Data Processing

Converting data from one form to a more useable and desired form is the goal of data processing. (Qiu et al., 2016; Roy et al., 2018) Making it more meaningful and informative by simplifying it. Depending on the task at hand and the machine's specifications, the output of this entire process can take any desired form, such as graphs, videos, charts, tables, images, and so on.

3.5. Handling Categorical Data using data encoding[5]

- **Frequency Encoding:** It is a method of using the frequency of the categories as labels. In cases where the frequency is somewhat related to the target variable, it aids the model in understanding and assigning weight in direct and inverse proportion, depending on the nature of the data. The following features are encoded using mean encoding [insured_occupation , incident_severity , authorities_contacted]

incident_type = **{**Single Vehicle Collision: 4818; Multi-vehicle Collision: 4775; Parked Car: 761; Vehicle Theft: 857}

- **Mean target encoding:** this refers to the mean value of the target variable on training data is used to determine the feature label for each category. Mean Encoding for columns that didn't have any ordinal order. The following features are encoded using mean encoding

[5] A Complete Guide to Categorical Data Encoding - (analyticsindiamag.com)

[policy_state, policy_csl, insured_hobbies, insured_relationship ,collision_type, incident_state, incident_city, auto_make , auto_model]

collison_type = **{**Front Collision: 0.473; Rear Collision: 0.507; Side Collision: 0.444}

- **Label encoding** refers to converting the labels into a numeric form so as to convert them into the machine-readable form. for all columns that have two categories and also better for labeling and target variables. The following features are encoded using Label Encoding [insured_sex, property_damage, police_report_available , fraud reported, fraud_type]

fraud_reported = **{**Y: 1; N: 0}

- **Target guided Ordinal Encoding** In this technique, we will transform our categorical vari able by comparing it to the target or output variable. Choose a categorical variable. Apply t he aggregated mean of the categorical variable to the target variable. Give the category wit h the highest mean higher integer values or a higher rank. These features have some ordin al order, but we are not sure of it; so, we rank them in order of how well they predict t he Target Variable.

incident_severity = **{**Trivial Damage: 5; Minor Damage: 6; Total Loss: 7; Major Damage: 8}

- **An ordinal encoding** involves mapping each unique label to an integer value. This type of encoding is really only appropriate if there is a known relationship between the categories.

insured_education_level = **{**High School :1; College:2; Associate:3; Masters:4; JD:5; MD: PhD:7}

3.6. Data Normalization

All features are in different range so feature scaling can assist us in having features with the same scale that contribute equally to the result. The min-max scaler, also known as the normalization method, is used in this study among other feature scaling techniques. Where the minimum value gets transformed into 0 and maximum value gets transformed in to 1. The formula of minmax normalization is as follows:

$$x_n = \frac{x_{i-} x_{min}}{x_{max} - x_{min}}$$ ……Equation 3.1 **min-max normalization**

Where, Xn is normalized value of a variable X

Xi is current value of variable X

X_{Min} minimum value of variable X and,

X_{Max} is the maximum value of variable X

3.7. Feature Selection

The feature selection method is used to select the relevant features from the original dataset in order to build the model. In this study, six feature selection methods are used to select relevant features to train the models. After each selection methods prefers their best features, the sequential feature selection method and combine feature selection method which counts all best feature according

3.7.1. Sequential feature selection

The sequential forward selection procedure involves carrying out the subsequent steps to find the N features that will fit in the K-features subset that are the most appropriate out of all of them. First and foremost, the best feature out of all the features is chosen (i.e., using some criterion function). The best pair of features is then chosen by combining this greatest feature with one of the remaining features. The best feature triplet is then constructed by combining the two best features with one of the remaining features. This process is repeated until K features, the predetermined number of features, are chosen.

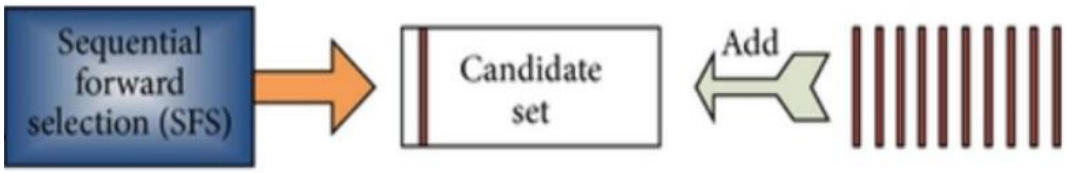


Figure 3.3 The sequential forward selection

Based on the cross-validation score of an estimator, this estimator determines the optimum feature to add or remove at each stage.(Marcano-Cedeño et al., 2010; Salappa et al., 2007)

3.7.2. Recursive Feature Elimination

Recursive feature elimination (RFE) (Misra & Yadav, 2020) is a feature selection method that fits a model and eliminates the weakest feature (or features) until the desired number of features is reached RFE's objective is to choose features by repeatedly taking into account smaller and smaller sets of features. The relevance of each feature is then determined after the estimator has been trained on the first set of features. The least significant features are then removed from the existing list of features. On the pruned set, this procedure is continued recursively until the

desired number of features to pick is attained. Then, recursively, the model's dependencies and collinearities are eliminated by removing a small number of features per loop.

3.7.3.Embedded Methods

The feature selection process is integrated into the learning or model building phases in the embedded method. It requires less computation than the wrapper method and is less prone to overfitting. Embedded methods combine the best features of Filter and Wrapper methods(Guyon, n.d.).

Tree-based Methods such as Decision Tree, Random Forest, Extra Tree, XGBoost are some of the tree-based methods one can use to get the feature importance. Most favorite ones are Random Forest and Boosted Trees (XGBoost, LightGBM, CatBoost) as they are improving the variance of simple Decision Trees by incorporating randomness.

3.7.3.1.Intrinsic Feature Selection - Random Forest

Random Forest estimates a feature's importance using mean decrease impurity. Mean decrease in impurity (MDI) is a measure of feature importance for decision tree models**.** They are computed as the mean and standard deviation of accumulation of the impurity decrease within each tree. The greater the importance of the feature, the lower the value is.

3.7.3.2. Intrinsic Feature Selection – XGBoost

Scikit-learn can use Feature Selection with XGBoost Feature for feature selection. This is accomplished by utilizing the SelectFromModel class, which accepts a model and can transform a dataset into a subset with selected features. This class can accept a pre-trained model, for example, one trained on the entire training dataset. By using a threshold, it may then pick which features to choose. This threshold is used to consistently select the same features on the training and test datasets when executing the transform function on the SelectFromModel instance.

3.7.4. Pearson Correlation

This approach uses filters. Look at the target and numerical features in our dataset's Pearson's correlation in terms of absolute value. On the basis of this criterion, we retain the top n features. It is applied to measure the linear relationship between the continuous variables X and Y. Its value varies from -1 to +1. Pearson’s correlation is given as: formula

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{ Equation 3.2}$$

Pearson's correlation Formula

r = Pearson correlation coefficient

x = values of the x-variable in a sample

$\bar{x}$ = mean of the values of the x-variable

y = values of the y-variable in a sample

$\bar{y}$= mean of the values of the y-variable

3.7.5.Chi-Squared

Another filter-based approach. In this method, we compute the chi-square metric between the target and the numerical variable and select only the variable with the highest chi-squared values. We can get observed count O and expected count E from the data of two variables. Chi-Square calculates the difference between the expected count E and the observed count O.

$$X^2c = \sum \frac{(Oi-Ei)^2}{Ei} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{ Equation 3.3 Chi-}$$

Squared Formula C=degrees of freedom, O= observed value(s), E= expected value(s)

3.7.6.Feature Selection by Combining Multiple Methods

Superior Feature Selection by Combining Multiple Models Even though it takes some work to get to the final results, it will be worth it because the combined power of multiple models can surpass any other feature selection method(Cateni et al., 2014; Rokach et al., 2006; Skurichina & Duin, 2005). Choosing the models. Combining the votes, the result will be an array with counts of how many times all models chose each feature for better result. (Lee, 2002)

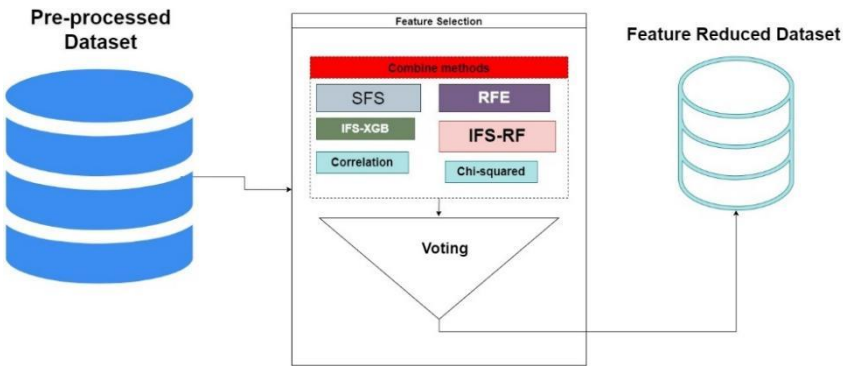


Figure 3.4 superior Feature selection by combining Multiple models

3.8. Model building based on ensemble machine learning

This study is intended to detect fraudulent auto insurance claims and their types using ensemble learning techniques using an already prepared and labeled dataset. The Classification models are built using ensemble learning algorithms. The goal is to get the insurance industry, specifically the auto insurance industry, to deal with fraudulent claims. stacking classifier is proposed for claim classification, RF is implemented for the type of fraud classification.

3.9. Model Evaluation

The evaluation of performance is a critical step in developing an accurate machine model. To ensure that the model fits the dataset and works well on new unseen input data, the classification model must be evaluated using model evaluation techniques.

3.9.1. Performance evaluators for the ensemble learning

Confusion Matrix: A confusion matrix is simply a two-dimensional table. Furthermore, both dimensions have True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)**.**

- **True Positives (TP): When both the actual and predicted classes are true.**
- **True Negatives (TN): When both the actual and predicted classes are false.**
- **False Positives (FP) occur when the actual class is false but the predicted class is true.**
- **False Negatives (FN): When the actual class is true but the predicted class is false.**

The confusion matrix shows how many correct and incorrect predictions the model made in comparison to the actual classifications in the test data. This study uses the confusion matrix with binary and four class model.

confusion matrix		Predicted values	
		Non-Fraud claim	Fraud claim
Actual values	Fraud claim	True Non-fraud claim TP	False Fraud claim FP
	Non-Fraud claim	False Non-fraud claim FN	True Fraud claim TN

Figure 3.5 Confusion Matrix with Binary-Classes

Confusion Matrix for the 5 classes		Predicted Values				
		overreporting	Staged	False claim	underwriting	notfraud
Actual Values	overreporting	True overreporting	false Staged	false claim	false underwriting	false notfraud
	Staged	false overreporting	True Staged	false claim	false underwriting	false notfraud
	claim	false overreporting	false Staged	True claim	false underwriting	false notfraud
	underwriting	false overreporting	false Staged	false claim	True underwriting	false notfraud
	notfraud	false notfraud	false notfraud	false notfraud	false notfraud	True notfraud

Figure 3.6 Confusion Matrix with 5-Classes

3.9.2.Classification Accuracy

It is defined as the proportion of correct predictions made to all predictions made. We can easily calculate it using the confusion matrix and the following formula:

$$\textbf{Accuracy} = \frac{TP+TN}{\text{TP + TN+FP + FN}} \quad \text{Equation3.4}$$

Accuracy Formula

Classification Report: - This report consists of the scores of Precisions, Recall, F1 and Support. They are explained as follows:

Precision: - The number of correct documents returned by our ML model, as used in document retrieval, can be defined. We can easily calculate it using the confusion matrix and the following formula:

$$\textbf{Precision} = \frac{\text{TP}}{\text{TP + FP}} \quad \text{Equation 3.5}$$

Precision Formula

Recall or Sensitivity: - may be defined as the number of positives returned by our ML model. Using the confusion matrix and the formula below, we can easily calculate it:

$$\textbf{Recall} = \frac{TP}{\text{TP + FN}} \quad \text{Equation 3.6}$$

formula of recall

Specificity: - in contrast to recall, may be defined as the number of negatives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula:

$$\textbf{Specifitity} = \frac{TN}{\textbf{TN+FP}}$$ ……………………………………………….…... Equation 3.7

Specificity formula

Support: - may be defined as the number of samples of the true response that fall into each class of target values.

F1 Score: - We will get the harmonic mean of precision and recall from this score. The F1 score is calculated as the weighted average of precision and recall. The best F1 value is 1 and the worst is 0. The following formula can be used to calculate the F1 score:

$$\boldsymbol{F1} = \frac{\mathbf{2} * (\textbf{Precision} * \textbf{Recall})}{(\textbf{Precision} + \textbf{Recall})}$$ ……………………………………………… Equation 3.8

F1 Score Formula

The precision and recall contributions to the F1 score are equal.

3.9.3.Cross Validation

Cross-Validation is a performance evaluation method for evaluating and comparing models that divide data into partitions: one for training a model and the rest for testing or validating the model. The cross-validation technique is the most commonly used evaluation technique to avoid overfitting. As a result, the most widely used K-fold cross validation evaluation method is used to ensure that the models correctly identified the pattern in the training dataset for this study. The basic form of the cross-validation technique is K-fold cross-validation. The results of each evaluation are averaged for a final score, and the final model is fitted to the entire dataset for operationalization. (Raschka, 2018) Cross Validation Scores In general, we can tell if a model is optimal by looking at its F1, precision, recall, and accuracy for classification.

3.10. Working Environment

Personal computer with the following descriptions is used. Processor Intel(R) Core (TM) i5-6500U CPU @ 2.50GHz, 2601 Mhz, 2 Core(s), 4Logical Processor(s), 8 Gigabyte of physical memory, 1TB hard disk storage capacities and Microsoft Windows 10 Enterprise,64bit operating system.

3.11. Implementation Environment

Table 3.1 Tools and Python Packages Used During the Implementation

Tools and packages	Version	Description
Anaconda navigator	1.9.12	Assist us in launching development applications and managing Anaconda packages, environments, and channels without the need for command-line tools.
Jupyter Notebook	6.0.1	It is a free web application that enables us to create and share documents. It is useful for data cleaning and transformation, modeling, data visualization, and machine learning.
Matplotlib	3.1.1	Publication quality figures in python. This study uses it for data and results visualization.
Microsoft Excel	2019	For dataset preparation CSV format
Microsoft Word	2019	For document preparation purpose
Numpy	1.16.5	Array processing for numbers, strings, and objects. This study uses it for handling converting the text to numeric data for features, training, and testing the model.
Pandas	0.25.1	Pandas is a useful package that provides fast, flexible, and expressive data structures for working with "relational" or "labeled" data. It supports data integration and filtering, as well as loading data from external sources such as Excel and managing the dataset.
Python	3.8	Python is a popular platform easy to learn programming language that can be used to create a ML application.
Scikit-learn	0.21.3	A python module used in machine learning for handling basic ML tasks clustering, regression, and classification.
Seaborn	0.9.0	Statistical data visualization
Mendeley desktop		It's a handy reference manager that also functions as a scholarly social network for finding related works.

CHAPTER FOUR
PROPOSED MODEL

4.1. Overview

This chapter discusses the proposed solution for auto insurance fraud claim detection using ensemble-learning techniques to solve the identified problem by using an auto insurance claim dataset with its severity for claim detection model (binary class) and fraud type classification model (five class). The data for this study was collected from GitHub and one column is added by the researcher for identifying the claim, and then prepared for detection based on the methodology that is discussed in the previous chapter. The general process flow is shown in the following figure.

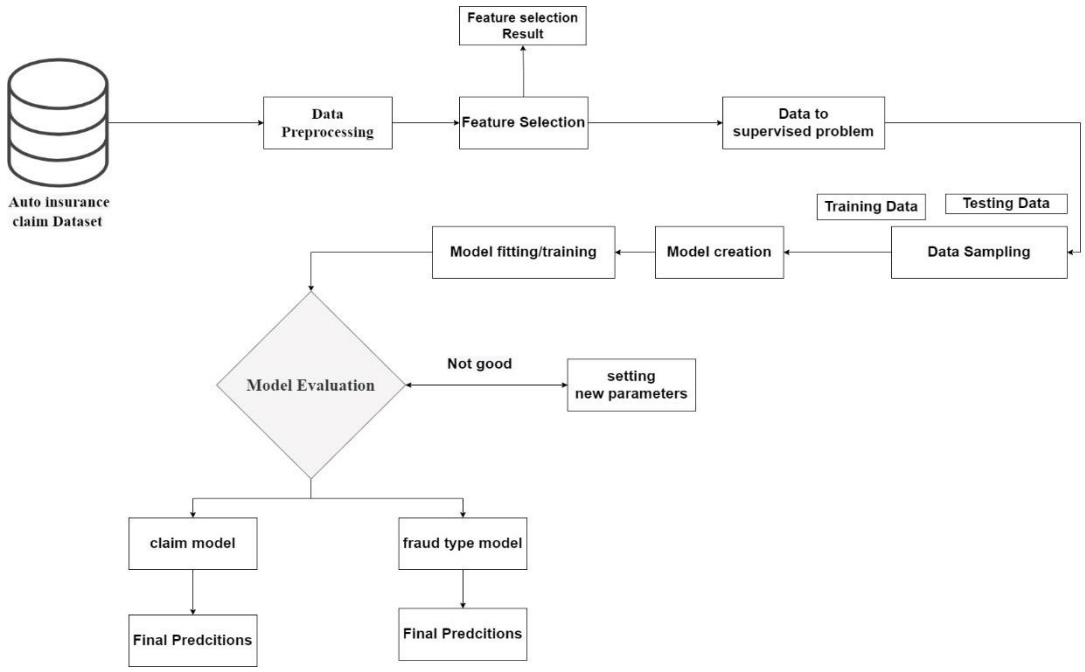


Figure 4.1 workflow diagram of proposed model for auto insurance fraud detection

4.2. Proposed Auto Insurance fraud claim detection model

Following the identification of the relevant features for the classification of Auto insurance claim and fraud type classification model for auto insurance claim data, the classification models were created by combining machine learning algorithms such as Random Forest (RF), bagging classifiers, XGBoost and Ada-boost, and stacking classifiers for claim detection model and rf for fraud type classification model.

The proposed model is used to classify auto insurance claim as a fraud or legal claim and identifying the fraud type using the fraud type classification model. The method takes the dataset as input and then the dataset is preprocessed to make the method complete and competent. Dataset preprocessing process includes data cleaning, handling missing values, handling categorical data. After preprocessing the dataset feature selection that has the following feature selection methods (Sequential Feature Selector, Recursive Feature Elimination, Intrinsic Feature Selection - Random Forest, Intrinsic Feature Selection – XGBoost, Pearson Correlation and Chi-Squared) to select relevant features for the classification of are applied. after the application of the superior feature selection using combining feature selection method, the model developed using Random Forest (RF), bagging classifier boosting classifiers, and stacking classifiers which are ensemble learning algorithms.

The proposed model was evaluated using K-fold cross-validation to avoid overfitting. Other evaluation metrics such as precision, recall, f-measure, sensitivity, and specificity are used. The evaluation result is important to select the best model to classify auto insurance claims. So, the best model was selected based on the above-mentioned metrics level for the classification of the claims. Finally, for the model development, python programming language is used. Anaconda navigator provides the IDE Jupyter Notebook. The notebook is used for building models and testing purpose.

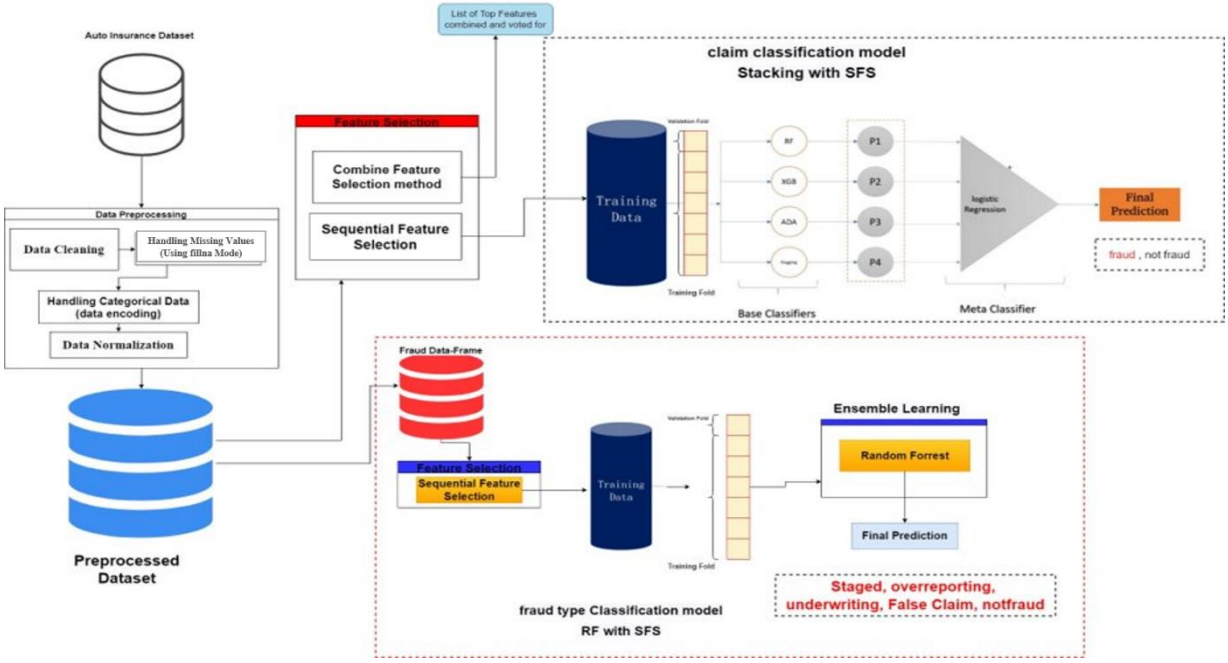


Figure 4.2Proposed model for Auto Insurance fraud claim detection

4.3. Auto Insurance Fraud claim detection Model Building

This study aims to develop a model that enables to classify whether the claim is legal or fraud as well as the type of the fraud that occurred. This study builds a classification model mainly by using ensemble machine-learning models, RF, bagging stacking boosting classifiers. For creating a classifiers model, a total instant of 11,210 records were collected for training and testing using learning algorithms. In this study, the classification process outcome is predicted from a given input or futures relation to class.

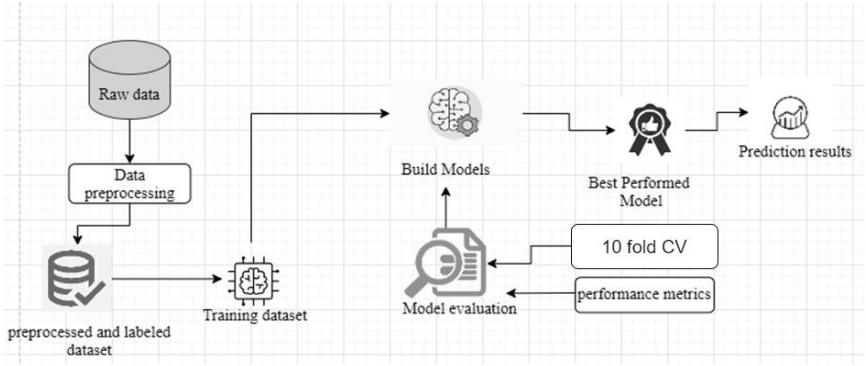


Figure 4.3 Model Building Flow Diagram

4.3.1. Stacking classifier for claim detection model

The most efficient technique to combine predictions from numerous ensembles' member models is learned using a machine learning model. Stacking is the name of this procedure. The architecture of a stacking model consists of two or more base models (also known as level-0 models) and a meta-model (also known as a level-1 model) that combines the predictions of the base models. Forecasts using out-of-sample data from base models are used to train the meta-model. Level-0 Models are models that are fitted to training data and have predictions compiled for them (Base-Models). Model at level 1 that learns the best method for combining predictions from base models

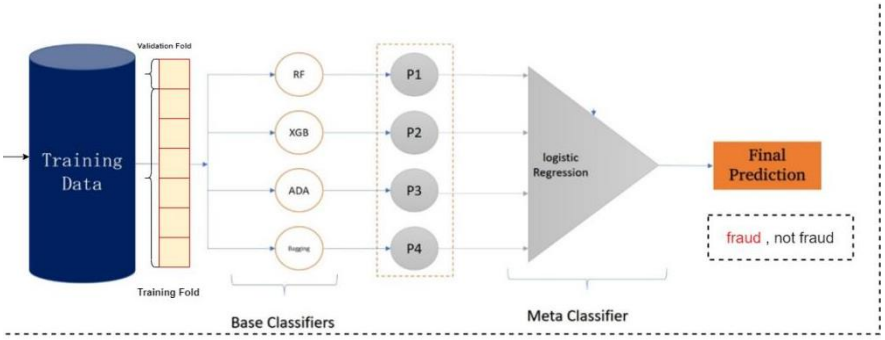


Figure 4.4 proposed Stacking model for claim detection model

Base models in this study are RF, XGBoosting, Adaboosting, and bagging while the meta model is logistic regression. The prediction from the base models fed in to the LR model the final prediction improves the predictions from the 4 models.

4.3.2.RF classifier for fraud type classification model

Random Forrest is a classifier that uses multiple decision trees on different subsets of a given dataset. Rather than relying on a single decision tree, the random forest uses the predictions from each tree to predict the final output based on the majority vote of predictions.

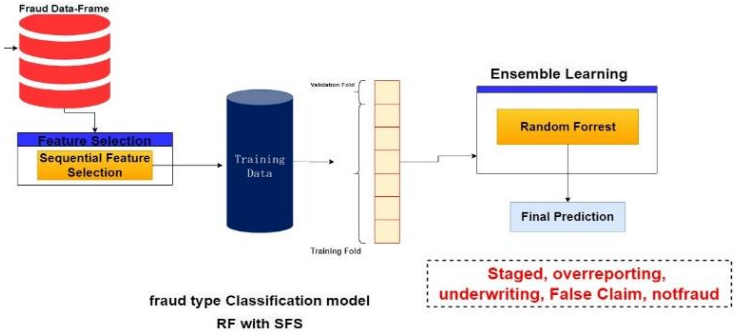


Figure 4.5 proposed RF model for fraud type classification model

4.4. Model Evaluation and Testing

The most important component of the study is evaluating and testing the Ensembel learning models developed on the auto insurance claim dataset to measure their performance. Because five algorithms were employed in this study, comparing and selecting the optimal model for further prediction is critical. The success of the classifier model is assessed in this study using 10-fold cross-validation and other performance evaluation measures appropriate for models, including accuracy, recall, F1 score, sensitivity, specificity, and confusion matrix, as mentioned in the previous chapter. One of the important aspects of machine learning is evaluating the performance of learning algorithms. In this study, we used the k-fold cross-validation of K = 10 that divided data equally to 10 folds, where 9-folds are used for training, and the remaining 1- fold is used for evaluation or testing. The evaluation technique implement using the sklearn cross_val_score() and cross_val_predict() methods using K value ten. These methods use the models and dataset to validate the learning skill of each model.

4.5. Creating Classification Report

A classification report is a metric for evaluating machine learning performance. It displays the precision, recall, F1 Score, and support of the trained classification model.

CHAPTER FIVE

IMPLEMENTATION AND EXPERIMENTATION

The chapter describes how to implement the proposed solution for auto insurance claim detection using the methodology and to implement the proposed solution presented discussed in the last two chapters. Three experiments were carried out in this study and described below.

```
#importing the library package for modeling and metrics for model evaluation.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

Figure 5.1 Code snippet for importing multiple libraries

5.1. Dataset Description

The goal of this study is to create an ensemble model that can identify the type of fraud that occurred when it comes to auto insurance claims. The dataset originally had a binary class, but after examining the features of the frauds, a new column (fraud_type) was added with five classes. All analyses in this study are performed on this dataset which includes claims for a car insurance company in the United States the data set can be categorized as a global dataset with improvement of 1 column. The data set includes 10,211 individual claims and 1000 claim merged together for total of 11,210. There are two target variable the first one is reported_fraud. If a claim is reported to be fraudulent, this variable is labeled Y; otherwise, it is labeled N. the next target variable is fraud_type which has 5 classes by name staged, overreporting, underwriting, False Claim, notfraud. Each claim in the data is described by 40 different attributes, which are represented as columns.

Table 5.1 Dataset Features Description

No	Features name	Dtypes	Variable	Missing data	Uniqueness
1	months_as_customer	int64	Independent	0%	No
2	Age	int64	Independent	0%	No
3	policy_number	int64	Independent	0%	Yes
4	policy_bind_date	object	Independent	0%	No
5	policy_state	object	Independent	0%	No
6	policy_csl	int64	Independent	0%	No
7	policy_deductable	float64	Independent	0%	No
8	policy_annual_premium	float64	Independent	0%	No
9	umbrella_limit	int64	Independent	0%	No
10	insured_zip	object	Independent	0%	No
11	insured_sex	object	Independent	0%	No
12	insured_education_level	object	Independent	0%	No
13	insured_occupation	object	Independent	0%	No
14	insured_hobbies	object	Independent	0%	No
15	insured_relationship	int64	Independent	0%	No
16	capital-gains	int64	Independent	0%	No
17	capital-loss	object	Independent	0%	No
18	incident_date	object	Independent	0%	No
19	incident_type	object	Independent	0%	No
20	collision_type	object	Independent	14.4%	No

21	incident_severity	object	Independent	0%	No
22	authorities_contacted	object	Independent	0%	No
23	incident_state	object	Independent	0%	No
24	incident_city	object	Independent	0%	No
25	incident_location	int64	Independent	0%	Yes
26	incident_hour_of_the_day	int64	Independent	0%	No
27	number_of_vehicles_involved	object	Independent	0%	No
28	property_damage	int64	Independent	37.7%	No
29	bodily_injuries	int64	Independent	0%	No
30	Witnesses	object	Independent	0%	No
31	police_report_available	float64	Independent	34.9%	No
32	total_claim_amount	int64	Independent	0%	No
33	injury_claim	int64	Independent	0%	No
34	property_claim	int64	Independent	0%	No
35	vehicle_claim	object	Independent	0%	No
36	auto_make	object	Independent	0%	No
37	auto_model	object	Independent	0%	Yes
38	auto_year	int64	Independent	0%	No
39	fraud_reported	object	Target	0%	No
39	fraud_type	object	Target	0%	No
40	_C39	Object	Independent	100%	

The following sample data can be seen before any process is initiated by implementing import data for the auto insurance claim dataset.

```
# let's import the data
data = pd.read_csv('dataset/insurance_claim_merged.csv')
# let's take a look at the data
pd.set_option('display.max_columns', None)
data.head(10)
```

Figure.5.2 importing auto insurance claim dataset

Table 5.2 Sample Data

witnesses	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year	fraud_reported
1	YES	96200.0	3000	500	58870	Saab	92x	1997	N
5	?	31200.0	3830	7370	32130	Saab	95	2006	N
3	?	14500.0	0	0	5690	Chevrolet	Malibu	1996	N
1	NO	7500.0	0	0	420	Saab	95	1990	N
1	YES	16500.0	5400	4300	8270	Toyota	Highlander	1998	N
1	NO	37710.0	14780	5400	48800	Honda	Civic	1995	N
1	YES	36400.0	6100	190	27630	Chevrolet	Tahoe	2005	N
3	?	83000.0	16000	770	58000	Accura	RSX	2009	N
2	?	34510.0	11000	5880	24800	BMW	M5	2005	N
2	?	91100.0	11410	14160	59520	Audi	A5	2000	N

5.2. Pre-processing Implementation

Preprocessing implementation includes handling missing values, handling categorical values, and Feature Selection. All of them are implemented using a python programming language as discussed in chapters three and four to make it suitable for the ensemble learning algorithm to build classifying models.

5.2.1. Missing value implementation

The following code snippet for checking missing value implemented. For missing value

```
data = data.replace('?',np.NaN)

data.isnull().any()
```

Figure 5.5.3 checking for missing values in the Auto insurance claim dataset

In order to fill the missing values in the dataset, we have used the fillna() method that replaces the missing values by calculating the mode value.

```
# missing value treatment using fillna

# we will replace the '?' by the most common collision type as we are unaware of the type.
data['collision_type'].fillna(data['collision_type'].mode()[0], inplace = True)

# It may be the case that there are no responses for property damage then we might take it as No property damage.
data['property_damage'].fillna('NO', inplace = True)

# again, if there are no responses fpr police report available then we might take it as No report available
data['police_report_available'].fillna('NO', inplace = True)

data.isnull().any().any()
```

Figure 5.4 Missing Value Treatment using fillna

5.2.2.Handling Categorical Data (data encoding) implementation

After checking out for the missing values which is found none and the next step is to change the categorical values to numerical.

- **Frequency Encoding:** The following features are encoded using mean encoding [insured_occupation , incident_severity , authorities_contacted]

```
incident_type_map=data.incident_type.value_counts().to_dict()
data.incident_type=data.incident_type.map(incident_type_map)
```

Figure 5.5 Frequency Encoding implementation

- **Mean target encoding:** The following features are encoded using mean encoding [policy_state, policy_csl, insured_hobbies, insured_relationship ,collision_type, incident_state, incident_city, auto_make , auto_model]

```
policy_state_map=data.groupby(['policy_state'])['fraud_reported'].mean().to_dict()

data.policy_state=data.policy_state.map(policy_state_map)
```

Figure 5.6 mean target encoding implementation

- **Label encoding** the following features are encoded using. [insured_sex , property damage,police_report_available , fraud reported, fraud_type]

```
fraud_map={'Y':1,"N":0}
data.fraud_reported=data.fraud_reported.map(fraud_map)
```

Figure 5.7 label encoding implementation

- **Target guided Ordinal Encoding** In this technique, we will transform the categorical variable by comparing it to the target or output variable.

```
authorities_map={j:i+5 for i,j in enumerate(data.groupby(["authorities_contacted"])
                ['fraud_reported'].mean().sort_values().index)}

data.authorities_contacted=data.authorities_contacted.map(authorities_map)
```

Figure 5.8 Target guided ordinal Encoding implementation

- **An ordinal encoding** involves mapping each unique label to an integer value. This type of encoding is really only appropriate if there is a known relationship between the categories.

```
education_map={'High School':1,'College':2,'Associate':3,'Masters':4,'JD':5,'MD':6,'PhD':7}
data.insured_education_level=data.insured_education_level.map(education_map)
```

Figure 5.9 ordinal encoding implementation

5.2.3. Data Normalization

Using min max normalization technique, we used this implementation below.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
# transform data
scaled = scaler.fit_transform(data)
print(scaled)
data= scaled
```

Figure 5.10 Min-Max Scaler

5.3. Implementation of Feature selection

After dealing with the data preprocessing steps such as handling missing values, and handling categorical values the next step is to select features.

5.3.1. Implementation of Sequential Feature Selector

Implementation of sequential feature selection method in order to reduce the features in the data set corresponding the target variables (fraud_reported and fraud_type).

```
sfs=sfs(KNeighborsClassifier(n_neighbors=10,weights='distance',metric='euclidean'),k_features=20,forward=True,floating=False
sfs.fit(data.drop(['fraud_reported'],axis=1),data.fraud_reported)
plot_sfs(sfs.get_metric_dict(),kind='std_dev')
```

Figure 5.11 Sequential Forward Selector for claim detection model

```
sfs=sfs(KNeighborsClassifier(n_neighbors=10,weights='distance',metric='euclidean'),k_features=20,forward=True,floating=False
sfs.fit(data.drop(['fraud_type'],axis=1),data.fraud_type)
plot_sfs(sfs.get_metric_dict(),kind='std_dev')
```

Figure 5.12 Sequential Forward Selector of fraud type classification model

5.3.2. Implementation of Superior Feature Selector

The features selection process using the combination of six models to identify the best features using superior feature selection method from the dataset,

```
feature_selection=pd.DataFrame(data={'Features':data.drop(['fraud_reported'],axis=1).columns,
                                     "Person_corr":cor_support,'CHI2':chi_support,
                                     "SFS":sfs_support,'RFE':rfe_support,'Rand-Forest':embeded_rf_support,
                                     "XGB":xgb_support})
feature_selection['Total'] = np.sum(feature_selection, axis=1)

feature_selection_df = feature_selection.sort_values(['Total'] , ascending=False)
feature_selection_df.index = range(1, len(feature_selection_df)+1)
feature_selection_df
# We take the top 20 features
feature_selection_df.loc[:20,'Features'].values
```

Figure 5.13 superior feature selection code snippet for claim detection model

5.4. Splitting the dataset

After dealing with the data preprocessing steps such as handling missing values, and handling categorical values the next step is to divide the dataset into features, and target which means independent, and dependent features respectively in order to build the proposed models.

```
x = data.drop(['fraud_reported'], axis = 1)
y = data['fraud_reported']
print("Shape of x :", x.shape)
print("Shape of y :", y.shape)
```

Figure 5.14 Splitting the dataset claim detection model

```
# let's split the data into dependent and independent sets
x = data.drop(['fraud_type'], axis = 1)
y = data['fraud_type']
```

Figure 5.15 Spliting the dataframe for fraud type classification model

5.5. Claim and fraud type classification models Implementation

Multiple ensembles learning models have been experimented and evaluated for both models and the stacking classifier outperforms the rest models. In both classifications claim (binary class) as well fraud type (four class) models. The stacking classifier takes the prediction from the four: RF, XGB, Bagging and ada boosting ensembles during claim detection model and three ensembles: RF, Bagging and Ada boosting during fraud type classification model then using the logistic regression as meta classifier to predicts the result. The benefit of stacking is that it can harness the capabilities of a range of well-performing models and make predictions that have better performance than any single model in the ensemble

```
lr = LogisticRegression()

# Starting from v0.16.0, StackingCVRegressor supports
# `random_state` to get deterministic result.
sclf = StackingCVClassifier(classifiers=[adb_dtree,rf,bgc,xgb],
                            meta_classifier=lr,
                            random_state=RANDOM_SEED)

print('10-fold cross validation:\n')

for clf, label in zip([adb_dtree, rf, sclf],
                      ['Ada boosting',
                       'Random Forest',
                       'bagging',
                       'XGBoosting',
                       'StackingClassifier']):

    scores = model_selection.cross_val_score(clf, x, y,
                                             cv=10, scoring='accuracy')
    print("Accuracy: %0.2f (+/- %0.2f) [%s]"
          % (scores.mean(), scores.std(), label))
```

Figure 5.16 stacking classifier code snippet for claim detection model

5.5.1.Ensemble model implementation for both models

RF: It is implemented with the RandomForestClassifier () method from the ensemble package's sklearn, which is used to fit and train the classifier.

```
rf=RandomForestClassifier(random_state=30,
                          max_depth=30, min_samples_leaf=1, min_samples_split=2, n_estimators=50)
rf.fit(X, y)
```

Figure 5.17 Code Snippet for Random Forest

Ada Boosting: Adaboost is less prone to overfitting as the input parameters are not jointly optimized. It is implemented with the AdaBoostClassifier() method from the ensemble package's sklearn which is used to fit and train the classifier.

```
adb_dtree=AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_depth=10,min_samples_leaf=10),
                             n_estimators=50, learning_rate=0.3, algorithm="SAMME.R")
adb_dtree.fit(X, y)
```

Figure 5.18 Code snippet for Ada Boosting

Bagging: It is implemented with the BaggingClassifier () method from the ensemble package's sklearn which is used to fit and train the classifier.

```
bgc=BaggingClassifier(base_estimator=KNeighborsClassifier(n_neighbors=1,weights='distance',metric='euclidean'),
                      n_estimators=20, max_samples=0.7)
bgc.fit(X, y)
```

Figure 5.19 code snippet for bagging classifier

XGBoost: It is implemented with the XgbClassifier () method from the ensemble package's sklearn which is used to fit and train the classifier.

```
xgb=XGBClassifier(use_label_encoder=True ,eval_metric='error'
                  max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=120)
xgb.fit(X,y)
```

Figure 5.20 code snippet for XGBoosting classifier

5.6. Model Testing and Evaluation

One of the important aspects of machine learning is evaluating the performance of learning algorithms. In this study, we used the k-fold cross-validation of K = 10 that divided data equally to 10 folds, where 9-folds are used for training, and the remaining 1- fold is used for evaluation or testing. The evaluation technique implement using the sklearn cross_val_score() and cross_val_predict() methods using K value ten. These methods use the models and dataset to validate the learning skill of each model.

```
print('10-fold cross validation:\n')

for clf, label in zip([adb_dtree, rf, sclf],
                      ['Ada boosting',
                       'Random Forest',
                       'bagging',
                       'XGBoosting',
                       'StackingClassifier']):

    scores = model_selection.cross_val_score(clf, x, y,
                                             cv=10, scoring='accuracy')
    print("Accuracy: %0.2f (+/- %0.2f) [%s]"
          % (scores.mean(), scores.std(), label))
```

Figure 5.21 applying 10-fold cross validation

In this study, we used the classification_report() and confusion_matrix() methods, which are used to assess how well built models perform by estimating the value of 10-fold CV for the auto insurance claim dataset.

```
print(classification_report(y, prediction))
rf.score(X, y)
plot_confusion_matrix(rf, X, ytest ,cmap='Blues')
```

Figure 5.22 code snippet of classification report and confusion matrix

These methods give a result of performance evaluation metrics such as precision, recall, f1-score and specificity of the model.

CHAPTER SIX

RESULTS, EVALUATION AND DISCUSSIONS

The results of an experiment employing ensemble machine-learning techniques to detect auto insurance fraud claims are presented in this chapter. The class distribution of the dataset is explored in this part, as well as studies to create classification models for detection using binary and four classes. Then we wrap up by going over the outcomes of each experiment in this research. This chapter discusses the findings in detail.

6.1. Dataset Class Distributions Result

The dataset used for this study contains 11,210 rows with 40 columns, including two target classes for claim classification, which is a binary class target variable that is labeled Y for fraud and N for notfraud. notFraud row size is 5870 and the fraud size is 5340.

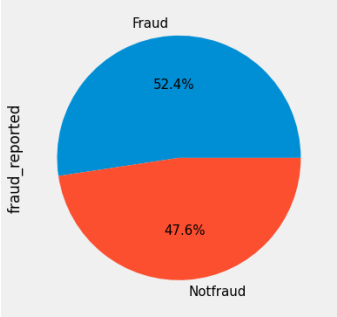


Figure 6.1 target class division in the claim dataset

The auto insurance claim dataset for fraud type now we will see the class representation of the five classes that are Staged, Overreporting, Underwriting, False Claim and not fraud. The size of each instance is 1407,1768, 879, 1286,5870 respectively. The details of the data frame class distribution are shown in the next figure.

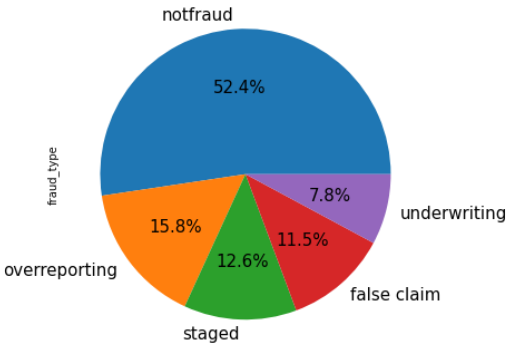


Figure 6.2 target class division in the fraud type data frame

6.2. Missing values result

Checking for null or missing value in the features is the first step in the pre-processing then the following result has been found.

Table 6.1 Using fillna the missing value problem is solved

Feature	isnull()	percentage	isnull()	percentage
collision_type	True	14.4%	False	0%
property_damage	True	37.7%	False	0%
police_report_available	True	34.9%	False	0%

6.3. Feature Selection Results for the claim detection model

The feature selection process using the combination of six models to identify the best features using Superior feature selection method and Sequential Feature Selector are applied. The superior feature selection has two steps which are Choosing the feature selection models (six for this study) and combining the votes, the result will be an array with counts of how many times all models chose each feature for better result. Here is the result for both feature selection methods.

Table 6.2 Top features using sequential feature selection

'policy_state', 'policy_csl', 'insured_sex', 'insured_education_level', 'insured_occupation', 'insured_hobbies', 'insured_relationship', 'incident_type', 'collision_type', 'incident_severity', 'authorities_contacted', 'incident_state', 'incident_city', 'number_of_vehicles_involved', 'property_damage', 'bodily_injuries', 'witnesses', 'police_report_available', 'auto_make', 'claim_duration'

The implementation of the combined feature selection method on the auto insurance dataset has given the following result and those features influences the predictions. Combine feature selection determines the best features during the auto insurance fraud claim process and adding the performance of ensemble learners it improves the detection of auto insurance fraud. Because this feature is selected based on the following six features selection method. Sequential Feature Selector, Recursive Feature Elimination, IFS-RF, IFS-XGBoost, Pearson Correlation, and Chi-Squared.

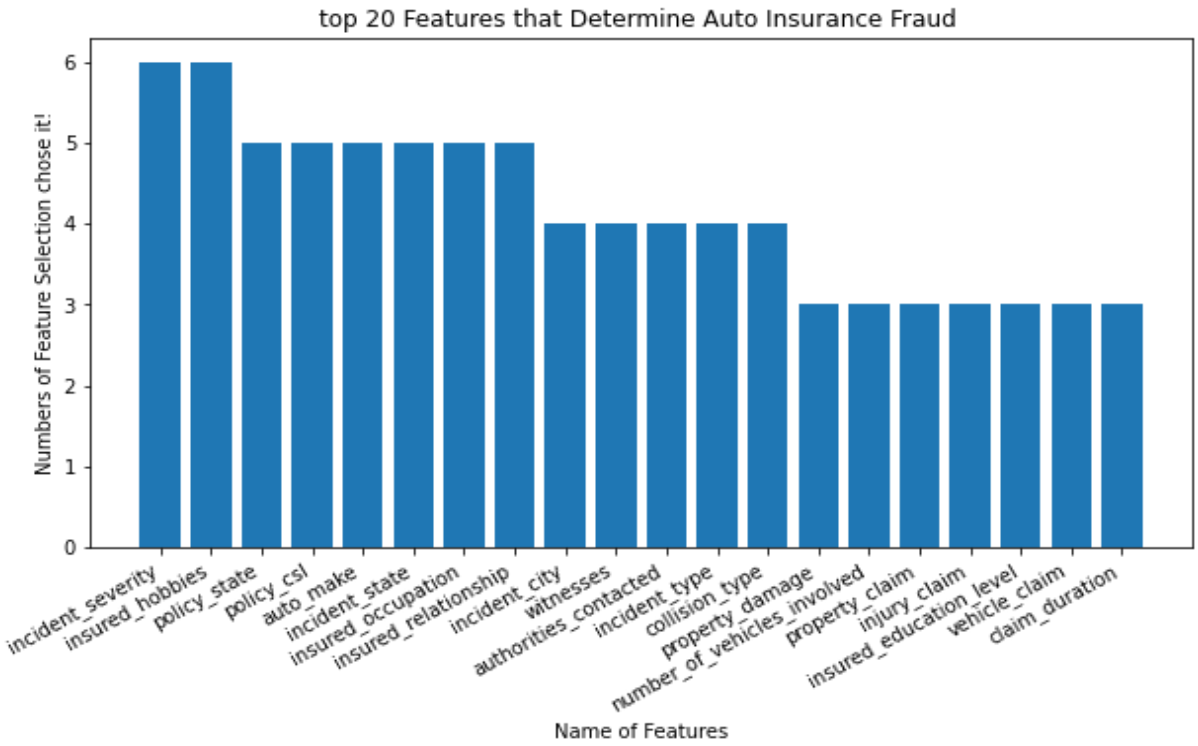


Figure 6.3 The 20 most important features based on superior feature selection

Combine selection methods yield a better result compared to single feature selection methods as well most of the time.

6.4. Model Building for Proposed Models

6.4.1. Model building of claim detection model

For claim detection model there are five models we proposed, namely: Random Forest, stacking, bagging, Ada boosting and XGBoosting. First, the data set is split into independent and dependent categories. Independent features are the features whose values are not determined by others, and the dependent feature is the target or the class whose value depends on the independent features. Then, the models were built on binary class datasets without and with the implementation of feature selection methods, which are sequential feature selection and superior feature selection method.

6.4.2. Model building of fraud type classification model

For the fraud type classification model, three models, namely, Random Forest, stacking, and Ada boosting, are proposed. First, the data set is split into independent and dependent categories. Independent features are the features whose values are not determined by others, and the dependent feature is the target or the class whose value depends on the independent

features. Then, the models were built on five classes (Staged, Overreporting, Underwriting, False Claim, and notfraud) of data frame with the implementation of Sequential Feature Selection method.

6.5. Model Evaluation Results for claim detection model

The experiment resulted in 15 models based on two feature selection methods and five ensemble classifiers. Testing and training of these models were performed using 10-fold cross-validation with other performance evaluation metrics as discussed in chapters three and four. The 10-fold cross-validation method randomly splits the dataset into ten equal-sized sets. Train the models using 10-1 folds and test them using one remaining fold. The process is iterated for each fold. The obtained results are presented in binary classification models. Three experiments were done and the results were divided into two sets: first, conducting the models on the original dataset, and second, applying the two feature selection methods (SFS and Superior feature selection method) with models on the auto insurance claim dataset.

6.5.1. Experiment 1 Models Evaluation Results on Original Dataset

These classification models built using the two-class auto insurance claim merged dataset that means the target classes are notfraud and fraud. The models are trained and tested using the 10-fold CV and other performance evaluation metrics of CV. In this experiment, first, the data is trained and tested model on the original dataset without applying feature selection methods.

Table 6.3 Models Accuracy of the Auto insurance claim before feature selection

Models	Ensemble Learners 10-Folds Cross Validation accuracy (%)										
	1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**	**10th**	**Mean**
RF	94.47	94.89	94.47	91.49	94.89	91.91	89.36	92.77	92.77	90.4	92.71
XGB	95.32	98.3	96.17	93.19	96.17	95.32	94.04	95.74	96.17	94.02	95.44
Ada	94.74	96.17	94.47	94.04	94.47	88.51	94.04	95.32	94.04	91.45	93.70
Bagging	60.85	60.0	54.04	58.3	62.13	58.72	54.47	60.43	62.55	52.99	58.45

The above table indicates the CV accuracy score for 4 classifiers with each fold of the dataset. The lowest average accuracy is 58.45 % from the Bagging model, and the highest average accuracy of 95.44 % resulted in the XGB boosting model. Ada Boosting and Random Forest

has 93.7% and 92.72% average accuracy respectively in addition to the accuracy of 10-fold cross-validation, this study uses other performance evaluation metrics such as precision, recall, f1_score, specificity, and confusion matrix as discussed in previous chapters. Table 6.3 below shows the result of each performance evaluation metrics for each model with the 10-fold CV. Confusion matrix for each model described in the following figure to show how correctly the models classify each class.

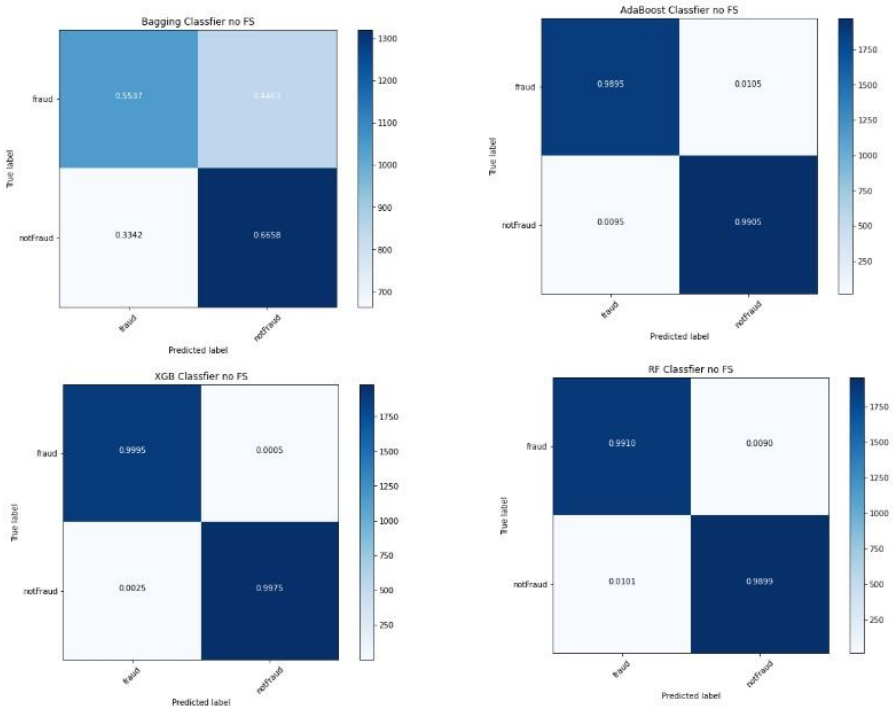


Figure 6.4 Confusion Matrix before feature selection

The above confusion matrix represents the binary confusion matrix for Bagging model, Ada boosting model, XGBoost and Random Forest. Bagging model classifies 54% of notFraud and 66 % of Fraud correctly and 46% of notFraud as Fraud and 34% of fraud as notfraud are misclassified. Ada boosting model classifies 99.91% of notFraud and 99.93 % of Fraud correctly and 0.09% of notFraud as Fraud and 0.07% of fraud as notfraud XGBoost model classify 100% of notFraud, 99.7% of Fraud correctly, and misclassifies 0.03% fraud as notFraud. Random Forest model classify 99.98% of notFraud, 98.8% of fraud correctly, and misclassifies 0.02% of notFraud as Fraud and 1.2 % fraud as notFraud. Finally, the result of binary models before applying the feature selection method demonstrates that the XGB model shows better accuracy, f1_score, and sensitivity than the Bagging, Ada boost and RF model.

6.5.2.Experiment 2 Models Evaluation Results after using SFS

Table 6.4 Models Accuracy of the Auto insurance

Models	Ensemble Learners 10-Folds Cross Validation accuracy (%)										
	1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**	**10th**	**Mean**
RF	98.97	98.45	98.71	98.2	97.68	98.45	98.71	98.71	98.71	98.97	98.56
XGB	99.48	98.71	99.48	98.71	98.45	98.71	98.71	98.71	98.97	99.22	98.97
Ada boosting	98.45	98.45	98.71	98.71	98.2	98.97	98.97	98.71	99.48	98.71	98.74
Bagging	92.78	90.46	89.43	90.21	92.01	91.99	92.25	94.57	93.28	94.83	92.18

The above table indicates the CV accuracy score for 4 classifiers with each fold of the dataset. The lowest average accuracy is 92.18% from the Bagging model, and the highest average acc uracy of 98.97% resulted in the XGB boosting model. Ada Boosting and Random Forest has 9 8.74% and 98.56% average accuracy respectively in addition to the accuracy of 10-fold cross-validation, this study uses other performance evaluation metrics such as precision, recall, f1_s core, specificity, and confusion matrix as discussed in previous chapters. Table 6.3 below sho ws the result of each performance evaluation metrics for each model with the 10-fold CV. Co nfusion matrix for each model described in the following figure to show how correctly the mo dels classify each class.

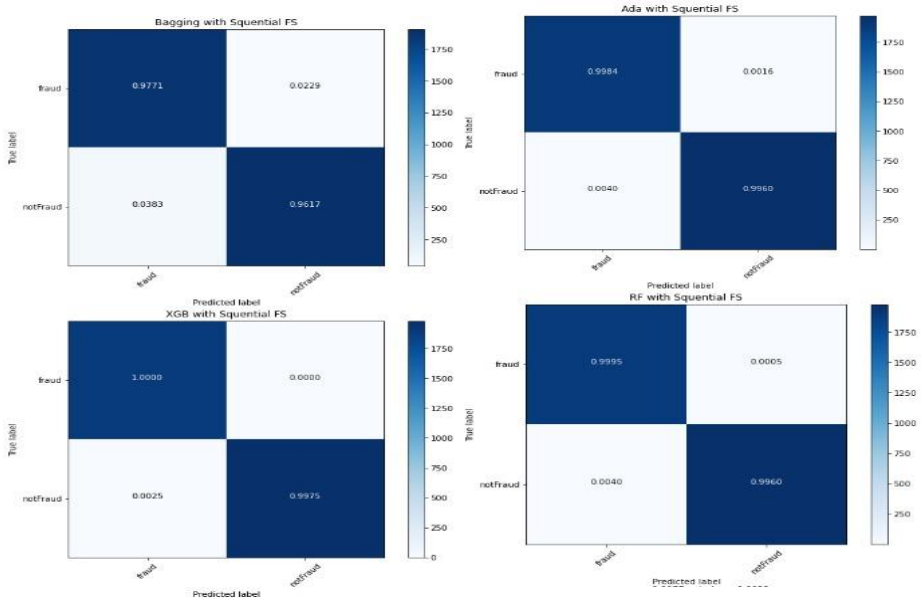


Figure 6.5 Confusion Matrix after using Sequential Feature Selection

The above confusion matrix represents the binary confusion matrix for Bagging model, Ada boosting model, XGBoost and Random Forest. Bagging model classifies 98% of notFraud and 96 % of Fraud correctly and 2% of notFraud as Fraud and 4% of fraud as notfraud are misclassified. Ada boosting model classifies 100% of notFraud and 99.99 % of Fraud correctly and misclassifies 0.01% of fraud as notfraud. XGBoost model classify 100% of notFraud, 99.75% of Fraud correctly, and misclassifies 0.25% fraud as notFraud. Random Forest model classify 100% of notFraud, 99.34% of fraud correctly, and misclassifies 0.76 % fraud as notFraud. Finally, the result of binary models before applying the feature selection method demonstrates that the XGB model shows better accuracy, f1_score, and sensitivity than the Bagging, Ada boost and RF model.

6.5.3.Experiment 3 Models Evaluation Results after superior feature selection

Table 6.5 Models Accuracy after superior feature selection

Models	**Ensemble Learners 10-Folds Cross Validation accuracy (%)**										
	1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**	**10th**	**Mean**
RF	99.23	97.68	97.94	98.71	97.42	95.87	97.67	97.42	97.42	97.42	97.68
XGB	98.45	97.42	98.97	98.97	98.71	98.45	98.19	97.67	98.19	98.97	98.4
Ada boosting	98.71	99.23	98.71	99.23	97.94	98.45	98.45	98.97	98.19	97.67	98.55
Bagging	61.34	56.19	56.44	62.11	52.84	58.14	61.76	58.91	60.21	60.47	58.84

The above table indicates the CV accuracy score for 4 classifiers with each fold of the dataset. The lowest average accuracy is 58.84 % from the Bagging model, and the highest average accuracy of 98.55 % resulted in the Ada boosting model. XGBoost and Random Forest has 98.4% and 97.68% average accuracy respectively in addition to the accuracy of 10-fold cross-validation, this study uses other performance evaluation metrics such as precision, recall, f1_score, specificity, and confusion matrix as discussed in previous chapters. Table 6.3 below shows the result of each performance evaluation metrics for each model with the 10-fold CV. Confusion matrix for each model described in the following figure to show how correctly the models classify each class.

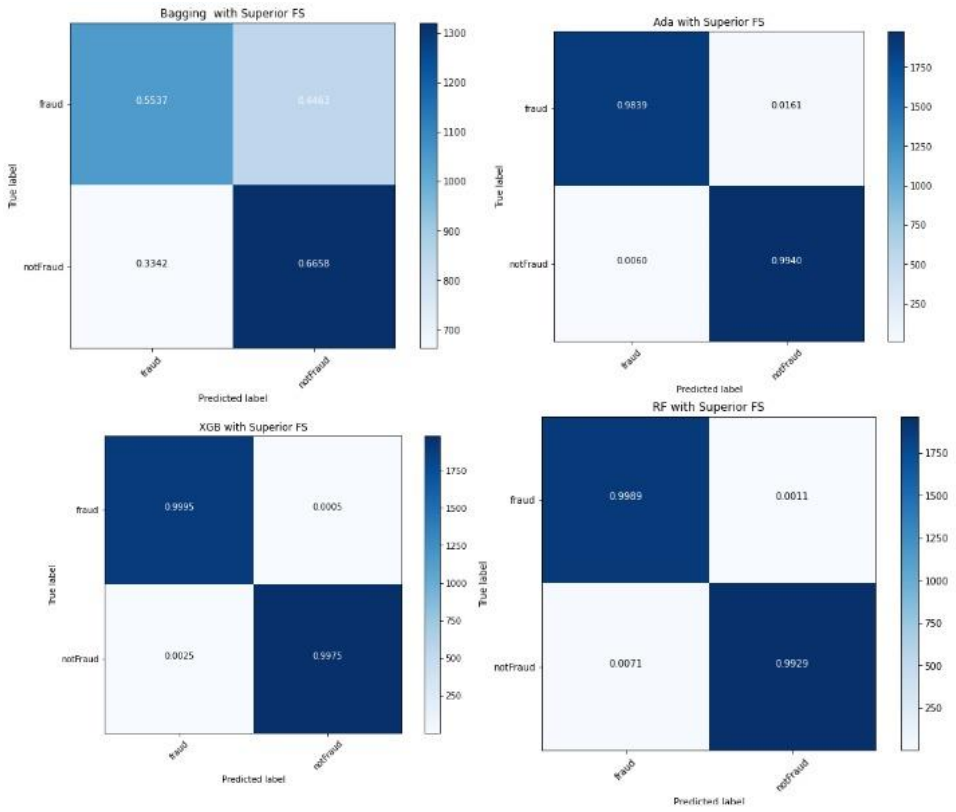


Figure 6.6 Confusion Matrix after superior feature selection

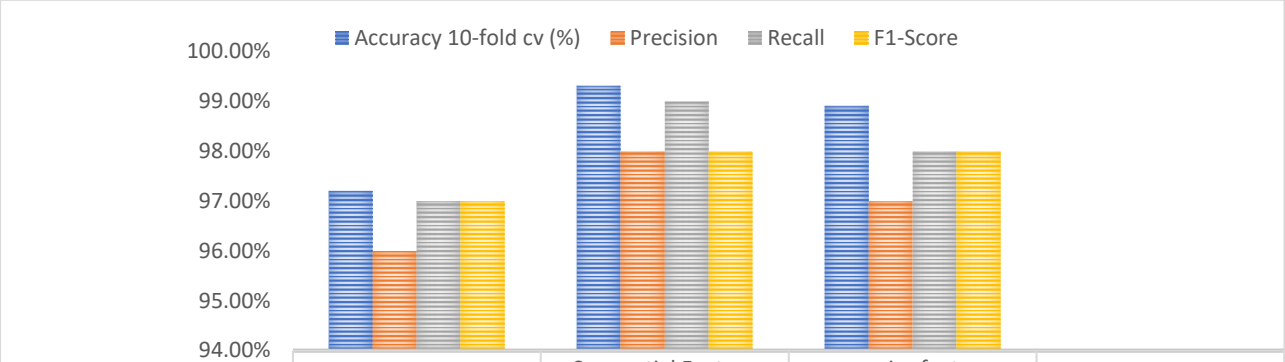
The above four confusion matrix represents the binary confusion matrix for Bagging model, Ada boosting model, XGBoost and Random Forest. Bagging model classifies 55% of notFraud and 68 % of Fraud correctly and 45% of notFraud as Fraud and 32% of fraud as notfraud are misclassified. Ada boosting model classifies 99.98% of not Fraud and 99.98 % of Fraud correctly and 0.02% of not Fraud as Fraud and 0.02% of fraud as not fraud XGBoost model classifies 100% of not Fraud, 99.7% of Fraud correctly, and misclassifies 0.33% fraud as not Fraud. Random Forest model classify 99.74% of not Fraud, 99.33% of fraud correctly, and misclassifies 0.26% of not Fraud as Fraud and 0.33 % fraud as not Fraud. Finally, the result of binary models after applying superior combined feature selection method demonstrates that the XGB model shows better accuracy, f1_score, and sensitivity than the Bagging, Ada boost and RF model.

Table 6.6 Comparing all three Experiments in single table

Case and models		Performance evaluation metrics				
		Precision	**Recall**	**F1_score**	**Specificity**	**Accuracy 10-CV**
Case 1: Original dataset after preprocessing	**RF**	0.98	0.97	0.98	0.96	92.71 %
	XGB	0.99	0.99	0.99	0.99	95.44 %
	Ada	0.96	0.99	0.98	0.96	93.70 %
	Bagging	0.61	0.67	0.64	0.67	58.45 %
Case 2: after Applying SFS feature selection method	**RF**	1.00	0.99	0.99	1.00	98.56 %
	XGB	1.00	0.99	0.99	1.00	98.97 %
	Ada	1.00	0.99	0.99	1.00	98.74 %
	Bagging	0.97	0.96	0.96	0.97	92.18 %
Case 3: After applying superior feature selection	**RF**	0.99	1.00	0.99	0.99	97.68 %
	XGB	0.99	0.99	0.99	0.98	98.4 %
	Ada	1.00	0.99	0.99	1.00	98.55 %
	Bagging	0.57	0.59	0.58	0.61	58.84 %

6.5.4.Stacking classifier for the three Experiments

The stacking classifier, or stacked generalization, consists of stacking the output of an individual estimator and using a classifier to compute the final prediction. Stacking allows us to capitalize on the strengths of each individual estimator by feeding their output into a final estimator. In this study the rf, XGB, bagging and Ada boost base classifier and logistic regression as meta classifier.



	Original Dataset	Sequential Feature Selector	superior feature selection method	
Accuracy 10-fold cv (%)	97.20%	99.31%	98.90%	
Precision	96.00%	98.00%	97.00%	
Recall	97.00%	99.00%	98%	
F1-Score	97.00%	98.00%	98%	

Figure 6.7 Accuracy of Stacking classifier for the three Experiments

6.6. Model Evaluation Results for Fraud type classification model

6.6.1. Results on Fraud type classification model

These classification models built using the two-class auto insurance claim merged dataset that means the target classes are notfraud and fraud. The models are trained and tested using the 10-fold CV and other performance evaluation metrics of CV. In this experiment, first, the data is trained and tested model on the original dataset without applying feature selection methods.

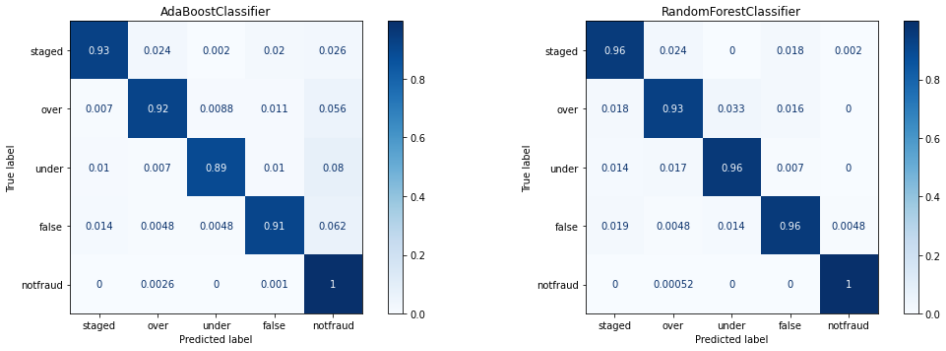


Figure 6.8 Confusion Matrix for the Ada boost and RF for fraud type classification model

The above confusion matrix represents the five-class confusion matrix for Ada boosting model and Random Forest. Accordingly, the model Ada boost classifier correctly classifies instances of 93%, 92%, 89%, 91% and 0.99% as Staged, Overreporting, Underwriting, False Claim, and notfraud respectively. And misclassifies 0.07%, 0.08%,0.11%, 0.09% and 0.01% in total against their classes respectively. Accordingly, the model random forest classifier correctly

classifies instances of 96%, 93%, 96%, 96% and 0.99% as Staged, Overreporting, Underwriting, False Claim, and notfraud respectively. And misclassifies 0.04%, 0.07%,0.04%, 0.04% and < 0.01% in total against their classes respectively.

Table 6.7 10-Folds Cross Validation accuracy (%) for fraud type classification model

Models	**10-Folds Cross Validation accuracy (%)**										
	1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**	**10th**	**Mean**
Stacking	73.78	76.89	75.56	71.43	75.45	75.0	74.55	77.68	78.57	73.21	75.21
Ada	85.78	90.67	87.11	86.16	89.73	85.71	87.05	90.18	88.84	87.05	87.83
RF	91.56	92.89	89.78	91.52	96.43	94.64	91.96	90.62	91.52	92.41	92.33

The 10 folds as shown in the table 6.7, among the three models, RF outperforms Stacking and ADA in Accuracy but for multi class classification like this with imbalanced data accuracy is not reason able comparison metric so checking the performance metrics such as precision, recall, and F1 score is necessary. Figure 6.9 compares the performance metrics precision, recall, and F1 score.

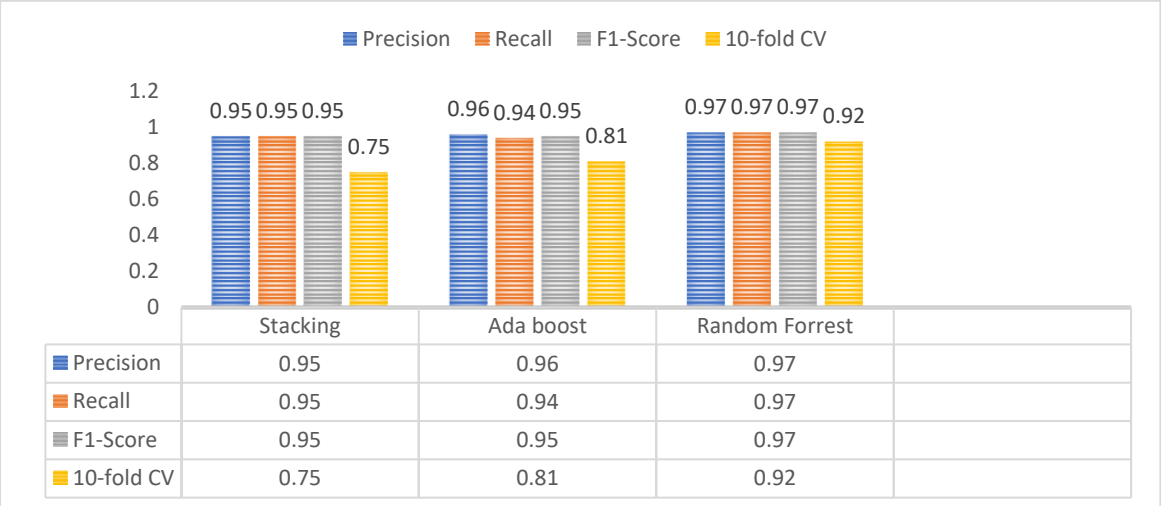


Figure 6.9 Comparison of performance metrics for the fraud type classification model

The result of five class models after applying the sequential feature selection method demonstrates that the RF model shows better 10-fold CV accuracy, precision, recall, and f1 score than the Ada boost and stacking models. The accuracy of multi class is liable to determine better performance when the data is imbalanced, so we choose to compare using precision

recall and f1 score. The random forest outperforms other the algorithms based on precision, recall and f1 score.

6.6.2.Hyperparameter Tuning

Controlling the behavior of a machine learning model requires hyperparameter tuning. Grid search is a kind of "brute force" approach to hyperparameter tuning. In order to fit the model, first we construct a grid of potential discrete hyperparameter values.(Yang & Shami, 2020) We keep track of the model output for each set and then choose the combination that has delivered the best output. To achieve a better outcome, we applied hyperparameters based on grid search for this binary and multiclass model.

Table 6.8 Hyperparameter tuning for claim detection model

Hyperparameter tune	**Best hyper parameters**
GridSearchCV RF	`{'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}`
GridSearchCV Adaboost	`{max_depth=10, min_samples_leaf=5, n_estimators = 50, learning_rate=0.3}`
GridSearchCV BGC	`{n_estimators=20, max_samples=0.7}`
GridSearch XGB	`{'min_samples_split': 2, 'n_estimators': 120}`

Table 6.9 Hyperparameter tuning for fraud type classification model

Hyperparameter tune	**Best hyper parameters**
GridSearchCV RF	`{'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 120}`
GridSearchCV Ada boost	`{max_depth=20, min_samples_leaf=5, n_estimators = 30, learning_rate=0.33}`

6.7. Comparison

The proposed model is compared to similar models. Several related studies have been done using machine learning, ensemble learning, and deep learning to detect auto insurance fraud claims globally, our proposed model, which is the claim detection model based on the stacking

classifier, has an accuracy of 99.31%. The proposed model also has another part called the fraud type classification model that classifies which kind of fraud has happened, which has precision, recall, and an f1 score of 97% based on RF using a sequential feature selector.

When our proposed claim detection model is compared to (Pranavi, 2020), our model is based on ensemble learning for classification of claims (claim detection model). This type of fraud model improves the lack of recognition of types of auto insurance fraud that do not exist in the related work paper. Their class is binary while ours is multiclass. That is also another difference between their paper and ours. The other thing our models select is the best predictive features of auto insurance claims. The model proposed by (Hassan & Abraham, 2016) compared to our models lacks the feature selection property and detection of what type of auto insurance fraud occurred. When compared to (Hanafy & Ming, 2021), our study performs better in terms of the accuracy of the machine learning algorithms. While they use machine learning, ours uses Ensemble and no feature selection method is implemented, and fraud type is also not considered in this related work. Our model, which is called auto insurance fraud detection in general, has the following three main functions that are incorporated at once: an ensemble-based stacking classifier with a sequential feature selector method to determine the binary class classification of a claim detection model with an accuracy of 99.31%; and another fraud type classification model for classification of the type of auto insurance fraud based on an RF classifier that is trained after the implementation of feature selection with precision, recall, and an f1-score of 97%. The third function is feature selection using a combined feature selector based on the following six methods, which are: sequential Feature Selector, Recursive Feature Elimination, IFS-RF, IFS-XGBoost, Pearson Correlation, and Chi-Squared feature selection methods, which combine and vote the best features that increase the classification performance of the auto insurance claim detection model.

6.8. Discussion

Auto insurance fraud claim is a major problem that surrounds the insurance industry with billions of moneys at stake(Insurance Fraud, 2015). Insurers and insureds are losing tons of time and wealth through fraudulent insurance claims. Thus, detection of auto insurance fraud claims is helpful to avoid the loss of wealth and time for both parties. The proposed auto insurance fraud claim detection using ensemble learning techniques has one dataset that is merged together from two datasets. The dataset has 40 features with different types of numerical and categorical values along with the class to which each instance belongs. The

researcher added one column based on the characteristics of the fraud based on an indication of auto insurance fraud claims (Accidents, n.d.; Fraud Indicators “Red Flags” – Organized Fraud, n.d.; Insurance Fraud Bureau, 2012; Müller, 2014; Tennyson & Salsas-Forn, 2002; Weinstock, 1975) .

The auto insurance dataset had missed some values, and these missing values were handled using fillna. The model used five distinct types of data encoding techniques to encode various kinds of categorical data to numerical data, such as frequency encoding, mean target encoding, label encoding, target guided ordinal encoding, and ordinal encoding. The prepared dataset has two target variables (fraud_reported and fraud_type), with a total dataset size of 11,210. Combining feature selection determines the best features during the auto insurance fraud claim process and adding the performance of ensemble learners will definitely improve the detection of auto insurance fraud. Because this feature is selected based on the following six features' selection method. Sequential Feature Selector, Recursive Feature Elimination, IFS-RF, IFS-XGBoost, Pearson Correlation, and Chi-Squared. The ensemble classifier models form the basis of the entire work. Five ensemble-learning models, including Random Forest (RF), bagging classifiers, XGBoost, AdaBoost, and stacking classifiers, are built and assessed for the claim detection model using three separate experiments.

The claim detection model was created using two different types of feature selection approaches. Precision, recall, F1 score, and specificity are performance evaluation metrics used in the model evaluation task, along with 10-fold cross-validation. When evaluating the model's performance, the confusion matrix is utilized to display the classes that were correctly and wrongly categorized. Three experiments were prepared before model building for the claim dataset. Experiment 1: model building before applying any feature selection method. Experiment 2: applying the Sequential Feature Selector, then model building. Experiment 3: model construction after using the superior feature selection method with combined votes from six feature selection methods. So, the following discussion is based on the results from the three experiments. The stacking classifier is the most accurate ensemble learning algorithm. a stacking classifier is based on the other four ensemble classifiers' results and most of the time it outperforms them because it uses a meta-learning algorithm to learn how to best combine the predictions from two or more base machine learning algorithms. After the stacking classifier in this study, the following ensemble classifiers are best for the above three experiments. XGBoosting, AdaBoosting, Random Forest, and Bagging are the best, respectively.

Three ensemble-learning models, including Random Forest (RF), AdaBoost, and stacking classifiers, are built and assessed for the fraud type classification model using Sequential Feature Selector, followed by model building. Precision, recall, F1 score, and specificity are performance evaluation metrics used in the model evaluation task, along with 10-fold cross-validation. When evaluating the model's performance, the confusion matrix is utilized to display the classes that were correctly and wrongly categorized. The Random Forest classifier is the most accurate ensemble learning for the fraud type classification model. Then Ada's boosting and stacking follow. There are features frequently selected in each feature set, which indicates that they are the most predictive features and have a strong predictive relationship to the class. The most frequently selected features using all feature selection methods in all models are displayed in figure 6.2.

CONCULSION AND RECOMMENDATION

The experiments carried out on proposed auto insurance fraud claim detection using ensemble learning techniques will be concluded, recommended, and future research directions will be forwarded in this chapter.

Conclusion

Fraud claim detection is very crucial for both the insurers and the insureds to prevent loss of money and time, as well as affect the whole credibility of the insurance industry. In this study, an ensemble-learning technique is proposed in order to detect auto insurance fraud claims from the legal ones and the type of the fraud that has been occurred. The proposed study also employs Superior Feature Selection in order to select the most relevant and predictive features before building the model.

The stacking classifiers with sequential feature selection, which delivered an accuracy of 99.31% for the claim detection model (binary class) across all three Experiment, is the best-performing ensemble for the claim detection model. Random Forest classifier with sequential feature selection, is the best classifier for the fraud type classification model (five classes) Based on precision, recall, and f1-score which all scored 97%.

Findings from this research

- The pattern of auto insurance fraud around the world is very similar and insurance fraud is worldwide problem where every customer is a victim even if one person commits fraud, insurance companies increase premium for every customer.
- There is little to no integration of the insurance claim system and technology, which leaves a huge burden on insurers' shoulders in Ethiopia and finding organized data is difficult, especially locally where there is no auto insurance claim data organized.
- The experiment finding shows that the proposed approach for claim detection model stacking classifier shows better accuracy as it is compared to other classification methods.
- The experiment finding shows that the proposed approach for fraud type classification model RF shows better accuracy as it is compared to other classification methods.
- Finding pattern where less fraud presents can be tackled using combined feature selection to increase the ability of auto insurance fraud detector.

- Finding highly predictive features using combine feature selector is better from single feature selector because, the combined vote of those six methods shows the most prevalent features compared to single feature selector method for the auto insurance claim dataset.
- Ensemble models improves when the sequential feature selection method implemented for detection of auto insurance datasets.

Contribution

- Identifying fraud is common but what type of fraud is the additional improvement this study provides.
- Implementing combine feature selection method to determine fraud characteristics that matter during claim process and identifying those feature that enables one to predict fraud with multiple experiments.
- Comparison of the proposed model with other different ensemble models such as RF, XGBoost, Ada boost, bagging and stacking for claim detection and fraud type classification model.

Recommendation

These days Auto Insurance fraud claims have become a challenge worldwide and many people are losing tons of wealth and time globally as well as locally in Ethiopia. The application of ensemble learning techniques to auto insurance claim data is an important research area to improve the services provided and come up with solutions to prevent the loss. Insurance companies should store insurance claim data in a computerized system that enables the experts and concerned parties to retrieve the data easily when it is needed. Though the dataset used for this study purpose is small in size when compared to the real-world auto insurance claim dataset, it is better to implement the models using a dataset with more instances.

Future work

- Text analytics, image and video processing as well as web or mobile based system should be considered for researcher who is interested in the auto insurance fraud detection.

* * *

This research was sponsored by Adama science and Technology University with a grant number

ASTU/SM-R/489/22 Adama, Ethiopia

REFERENCES

Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, *68*, 90–113. https://doi.org/10.1016/J.JNCA.2016.04.007

Accidents, C. (n.d.). *NICB Scene Indicators for Law Enforcement*. 1–12.

Alamir, E., Urgessa, T., Hunegnaw, A., & Gopikrishna, T. (2021). Motor Insurance Claim Status Prediction using Machine Learning Techniques. *International Journal of Advanced Computer Science and Applications*, *12*(3), 457–463. https://doi.org/10.14569/IJACSA.2021.0120354

Artís, M., Ayuso, M., & Guillén, M. (2002). Detection of Automobile Insurance Fraud With Discrete Choice Models and Misclassified Claims. *Journal of Risk and Insurance*, *69*(3), 325–340. https://doi.org/https://doi.org/10.1111/1539-6975.00022

Benedek, B., & László, E. (2019). Identifying Key Fraud Indicators in the Automobile Insurance Industry Using SQL Server Analysis Services. *Studia Universitatis Babes-Bolyai Oeconomica*, *64*(2), 53–71. https://doi.org/10.2478/subboec-2019-0009

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1007/bf00058655

Brockett, P. L., Xia, X., & Derrig, R. A. (1998). Using Kohonen's Self-Organizing Feature Map to Uncover Automobile Bodily Injury Claims Fraud. *The Journal of Risk and Insurance*, *65*(2), 245–274. https://doi.org/10.2307/253535

Bühlmann, P. (2012). Bagging, boosting and ensemble methods. In J. E. Gentle, W. K. Härdle, & Y. Mori (Eds.), *Handbook of Computational Statistics: Concepts and Methods: Second Edition* (pp. 985–1022). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-21551-3__33

Cateni, S., Colla, V., & Vannucci, M. (2014). A hybrid feature selection method for classification purposes. *Proceedings - UKSim-AMSS 8th European Modelling Symposium on Computer Modelling and Simulation, EMS 2014*, 39–44. https://doi.org/10.1109/EMS.2014.44

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1857 LNCS*, 1–15. https://doi.org/10.1007/3-540-45014-9_1

Fraud Indicators " Red Flags " – Organized Fraud. (n.d.).

García, S., Luengo, J., & Herrera, F. (2015). Feature selection. In C. Sammut & G. I. Webb (Eds.), *Intelligent Systems Reference Library* (Vol. 72, pp. 163–193). Springer US. https://doi.org/10.1007/978-3-319-10247-4_7

Gomes, C., Jin, Z., & Yang, H. (2021). Insurance Fraud Detection with Unsupervised Deep Learning. *Journal of Risk & Insurance*. https://doi.org/10.1111/jori.12359

Guyon, I. (n.d.). Lecture 9: Embedded Methods. *Search*, 1–12. http://files/96/Guyon and Elisseeff - Lecture 9 Embedded Methods.pdf

Hanafy, M., & Ming, R. (2021). Machine Learning Approaches for Auto Insurance Big Data. In *Risks* (Vol. 9, Issue 2). https://doi.org/10.3390/risks9020042

Hassan, A. K. I., & Abraham, A. (2016). *Modeling Insurance Fraud Detection Using Ensemble Combining Classification*.

Herbert, K. G., & Wang, J. T. L. (2007). Biological data cleaning: A case study. *International Journal of Information Quality*, *1*(1), 60–82. https://doi.org/10.1504/IJIQ.2007.013376

Insurance Fraud Bureau. (2012). *Crash for Cash: Putting the brakes on fraud*. 1–24.

Kang, H. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, *64*(5), 402–406. https://doi.org/10.4097/kjae.2013.64.5.402

Kunickaitė, R., Zdanavičiūtė, M., & Krilavičius, T. (2020). Fraud detection in health insurance using ensemble learning methods. *CEUR Workshop Proceedings*, *2698*.

Marcano-Cedeño, A., Quintanilla-Domínguez, J., Cortina-Januchs, M. G., & Andina, D. (2010). Feature selection using Sequential Forward Selection and classification applying Artificial Metaplasticity Neural Network. *IECON Proceedings (Industrial Electronics Conference)*, 2845–2850. https://doi.org/10.1109/IECON.2010.5675075

Misra, P., & Yadav, A. S. (2020). Improving the classification accuracy using recursive feature elimination with cross-validation. *International Journal on Emerging Technologies*, *11*(3), 659–665. http://files/151/Misra and Yadav - 2020 - Improving the Classification Accuracy using Recurs.pdf

Morley, N. J., Ball, L. J., & Ormerod, T. C. (2006). How the detection of insurance fraud succeeds and fails. *Psychology, Crime & Law*, *12*(2), 163–180. https://doi.org/10.1080/10683160512331316325

Müller, K. (2013). *THE IDENTIFICATION OF INSURANCE FRAUD-AN EMPIRICAL ANALYSIS-KATJA MÜLLER WORKING PAPERS ON RISK MANAGEMENT AND INSURANCE NO. 137 EDITED BY HATO SCHMEISER CHAIR FOR RISK MANAGEMENT AND INSURANCE The Identification of Insurance Fraud-An Empirical Analysis*.

Müller, K. (2014). The Identification of Insurance Fraud - An Empirical Analysis. *I.VW-HSG Working Paper*.

Pfeffer, L. (1956). Issues That Divide. *Journal of Social Issues*, *12*(3), 21–39. https://doi.org/https://doi.org/10.1111/j.1540-4560.1956.tb00376.x

Pranavi, P. (2020). Analysis of Vehicle Insurance Data to Detect Fraud using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*, *8*, 2033–2038. https://doi.org/10.22214/ijraset.2020.30734

Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *Eurasip Journal on Advances in Signal Processing*, *2016*(1), 67. https://doi.org/10.1186/s13634-016-0355-x

Rokach, L., Chizi, B., & Maimon, O. (2006). Feature selection by combining multiple methods. In M. Last, P. S. Szczepaniak, Z. Volkovich, & A. Kandel (Eds.), *Studies in Computational Intelligence* (Vol. 23, pp. 295–304). Springer-Verlag. https://doi.org/10.1007/3-540-33880-2_30

Rosli, M. M., Tempero, E., & Luxton-Reilly, A. (2016). What is in our datasets? Describing a structure of datasets. *ACM International Conference Proceeding Series*, *01-05-Febr*, 1–10. https://doi.org/10.1145/2843043.2843059

Roy, S., Sharma, P., Nath, K., Bhattacharyya, D. K., & Kalita, J. K. (2018). Pre-processing: A data preparation step. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* (Vols 1–3, pp. 463–471). Elsevier. https://doi.org/10.1016/B978-0-12-809633-8.20457-3

Rückstieß, T., Osendorfer, C., & van der Smagt, P. (2011). Sequential feature selection for classification. In D. Wang & M. Reynolds (Eds.), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 7106 LNAI* (pp. 132–141). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-25832-9_14

Salappa, A., Doumpos, M., & Zopounidis, C. (2007). Feature selection algorithms in classification problems: An experimental evaluation. *Optimization Methods and Software*, *22*(1), 199–212. https://doi.org/10.1080/10556780600881910

Simfukwe, M., Kunda, D., & Chembe, C. (2015). *Comparing Naive Bayes Method and Artificial Neural Network for Semen Quality Categorization*. *2*(7), 689–695.

Skurichina, M., & Duin, R. P. W. (2005). Combining feature subsets in feature selection. *Lecture Notes in Computer Science*, *3541*, 165–175. https://doi.org/10.1007/11494683_17

Tennyson, S., & Salsas-Forn, P. (2002). Claims Auditing in Automobile Insurance: Fraud Detection and Deterrence Objectives. *The Journal of Risk and Insurance*, *69*(3), 289–308. http://www.jstor.org/stable/1558679

Viaene, S., & Dedene, G. (2004). Insurance Fraud: Issues and Challenges. *The Geneva Papers on Risk and Insurance - Issues and Practice*, *29*(2), 313–333. https://doi.org/10.1111/j.1468-0440.2004.00290.x

Wang, Y., & Xu, W. (2018). Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. *Decis. Support Syst.*, *105*, 87–95.

Weinstock, F. J. (1975). Introduction to insurance. *International Ophthalmology Clinics*, *15*(2), 189–190. https://doi.org/10.1097/00004397-197501520-00022

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Yang, L., & Shami, A. (2020). *On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice*.

APPENDIXES

Appendix A: Dataset Features Description

1. months_as_customer: It denotes the number of months for which the customer is associated with the insurance company.
2. age: continuous. It denotes the age of the person.
3. policy_number: The policy number.
4. policy_bind_date: Start date of the policy.
5. policy_state: The state where the policy is registered.
6. policy_csl: - combined single limits. How much of the bodily injury will be covered from the total damage?
7. policy_deductable: The amount paid out of pocket by the policy-holder before an insurance provider will pay any expenses.
8. policy_annual_premium: The yearly premium for the policy.
9. umbrella_limit: An umbrella insurance policy is extra liability insurance coverage that goes beyond the limits of the insured's homeowners, auto or watercraft insurance. It provides an additional layer of security to those who are at risk of being sued for damages to other people's property or injuries caused to others in an accident.
10. insured_zip: The zip code where the policy is registered.
11. insured_sex: It denotes the person's gender.
12. insured_education_level: The highest educational qualification of the policy-holder.
13. insured_occupation: The occupation of the policy-holder.
14. insured_hobbies: The hobbies of the policy-holder.
15. insured_relationship: Dependents on the policy-holder.
16. capital-gain: It denotes the monitory gains by the person.
17. capital-loss: It denotes the monitory loss by the person.
18. incident_date: The date when the incident happened.
19. incident_type: The type of the incident.
20. collision_type: The type of collision that took place.
21. incident_severity: The severity of the incident.
22. authorities_contacted: Which authority was contacted?
23. incident_state: The state in which the incident took place.
24. incident_city: The city in which the incident took place.

25. incident_location: The Street in which the incident took place.
26. incident_hour_of_the_day: The time of the day when the incident took place.
27. property_damage: If any property damage was done.
28. bodily_injuries: Number of bodily injuries.
29. Witnesses: Number of witnesses' present.
30. police_report_available: Is the police report available.
31. total_claim_amount: Total amount claimed by the customer.
32. injury_claim: Amount claimed for injury
33. property_claim: Amount claimed for property damage.
34. vehicle_claim: Amount claimed for vehicle damage.
35. auto_make: The manufacturer of the vehicle
36. auto_model: The model of the vehicle.
37. auto_year: The year of manufacture of the vehicle.
38. fraud_reported: Whether the claim is fraudulent or not.
39. _C39: empty column named c39 to represent the column as 39^{th}
40. fraud_type : different types of auto insurance fraud

Terminologies in auto insurance[6]

The following are the terminologies generally used in the auto or motor insurance areas.

Automobile Insurance: Coverage for the risks of driving or owning a car. Collision, liability, comprehensive, medical, and uninsured motorist coverage are all options.

Claim: Notice to an insurer that under the terms of a policy, a loss may be covered.

Claimant: Either the first or third party. This includes anyone who claims a right of recovery.

Collision (Auto): Reimburses you for damage to your vehicle caused by a collision with another vehicle or with any other movable or fixed object (for example, you accidentally backed into another object while pulling out from a parking stall and causing damage to the bumper and fender of your covered automobile).

Collision Deductive Waiver: If you are hit by an uninsured motorist who is negligent, this coverage will waive your collision deductible.

Deductible: The amount of the loss that the insured must pay before the insurance company will pay benefits. Increase your deductible to lower your premium.

Expiration Date: The end date of the policy.

Insured: The policyholder the person protected in case of a loss or claim.

Insurer: The insurance company.

Policy: The written contract of insurance.

Policy Limit: The maximum amount a policy will pay, either overall or under a specific coverage.

Premium: The amount of money charged by an insurance company for insurance coverage.

Underwriting:is the process of selecting applicants for insurance and categorizing them base d on their degrees of insurability in order to charge the appropriate premium rates. Unaccepta ble risks are rejected as part of the process.

[6] https://dhhinsurance.com/

Table A.1 Feature selection for six different feature selection methods

Feature Selection	The 20 top features
SFS	'policy_state', 'policy_csl', 'insured_sex', 'insured_education_level', 'insured_occupation', 'insured_hobbies', 'insured_relationship', 'incident_type', 'collision_type', 'incident_severity', 'authorities_contacted', 'incident_state', 'incident_city', 'number_of_vehicles_involved', 'property_damage', 'bodily_injuries', 'witnesses', 'police_report_available', 'auto_make', 'claim_duration'
RFE	'months_as_customer', 'policy_state', 'policy_csl', 'policy_annual_premium', 'insured_education_level', 'insured_occupation', 'insured_hobbies', 'insured_relationship', 'incident_type', 'collision_type', 'incident_severity', 'authorities_contacted', 'incident_state', 'incident_city', 'total_claim_amount', 'injury_claim', 'property_claim', 'vehicle_claim', 'auto_make', 'claim_duration'
IFS Random Forest	'months_as_customer', 'policy_state', 'policy_csl', 'policy_annual_premium', 'insured_education_level', 'insured_occupation', 'insured_hobbies', 'insured_relationship', 'incident_type', 'collision_type', 'incident_severity', 'authorities_contacted', 'incident_state', 'incident_city', 'total_claim_amount', 'injury_claim', 'property_claim', 'vehicle_claim', 'auto_make', 'claim_duration'
IFS XGBoost	'policy_csl', 'insured_hobbies', 'collision_type', 'incident_severity', 'incident_state', 'witnesses', 'police_report_available'
Filter Method – Correlation	'policy_csl', 'umbrella_limit', 'insured_occupation', 'insured_hobbies', 'insured_relationship', 'capital.gains', 'capital.loss', 'incident_type', 'collision_type', 'incident_severity', 'authorities_contacted', 'incident_state', 'incident_city', 'number_of_vehicles_involved', 'witnesses', 'total_claim_amount', 'injury_claim', 'property_claim', 'vehicle_claim', 'auto_make'
Filter Method - Chi2	'policy_csl', 'umbrella_limit', 'insured_occupation', 'insured_hobbies', 'insured_relationship', 'capital.gains', 'incident_type', 'collision_type', 'incident_severity', 'authorities_contacted', 'incident_state', 'incident_city', 'number_of_vehicles_involved', 'property_damage', 'police_report_available', 'total_claim_amount', 'injury_claim', 'property_claim', 'vehicle_claim', 'auto_make'

Appendix B: Sample Codes

 # Label Encoding

```
sex_map={'MALE':1,'FEMALE':0}

data.insured_sex=data.insured_sex.map(sex_map)

yes_no_map={'YES':1,"NO":0}

data.property_damage=data.property_damage.map(yes_no_map)

data.police_report_available=data.police_report_available.map(yes_no_map)

fraud_map={'Y':1,"N":0}

data.fraud_reported=data.fraud_reported.map(fraud_map)
```

fraud_type = Staged fraud_type = Overwriting fraud_type= Underwriting fraud_type = False claim fraud_type = notfraud	fraud_type =1 fraud_type = 2 fraud_type= 3 fraud_type = 4 fraud_type = 5
fraud_reported = N fraud_reported = Y	fraud_reported = 0 fraud_reported = 1

 # Wrapper Method - Recursive Feature Elimination

```
from sklearn.feature_selection import chi2 , f_classif ,mutual_info_classif,RFE

rfe=RFE(RandomForestClassifier(),n_features_to_select=20)

rfe=rfe.fit(data.drop(['fraud_reported'],axis=1),data.fraud_reported)

rfe_support=rfe.get_support()

data.drop(['fraud_reported'],axis=1).columns[rfe_support]
```

INTRINSIC FEATURE SELECTION – RANDOM FOREST

```
from sklearn.feature_selection import SelectFromModel

from sklearn.ensemble import RandomForestClassifier

embeded_rf_selector = SelectFromModel(RandomForestClassifier(n_estimators=100),
max_features=20)

embeded_rf_selector.fit(data.drop(['fraud_reported'],axis=1),data.fraud_reported)

embeded_rf_support = embeded_rf_selector.get_support()

data.drop(['fraud_reported'],axis=1).columns[rfe_support]
```

#Wrapper method – Sequential Feature Selector

```
from sklearn.ensemble import RandomForestClassifier

from mlxtend.feature_selection import SequentialFeatureSelector as sfs

from mlxtend.plotting import plot_sequential_feature_selection as plot_sfs

from sklearn.neighbors import KNeighborsClassifier

sfs=sfs(KNeighborsClassifier(n_neighbors=10,weights='distance',metric='euclidean'),

    k_features=20,forward=True,floating=False,scoring='recall',cv=3)

sfs.fit(data.drop(['fraud_reported'],axis=1),data.fraud_reported)

plot_sfs(sfs.get_metric_dict(),kind='std_dev')

sfs_support=[True if list(sfs.k_feature_names_).count(col)==1  else False for col in
data.drop(['fraud_reported'],axis=1).columns ]

data.drop(['fraud_reported'],axis=1)[list(sfs.k_feature_names_)].columns
```

 # Superior Feature Selection combining multiple methods

```
feature_selection=pd.DataFrame(data={
                        'Features':data.drop(['fraud_reported'],axis=1).columns,
                            "Person_corr":cor_support,'CHI2':chi_support,
                           "SFS":sfs_support,'RFE':rfe_support,"XGB":xgb_support,
                            'Rand-Forest':embeded_rf_support})
                            feature_selection['Total'] = np.sum(feature_selection, axis=1)
feature_selection_df = feature_selection.sort_values(['Total'] , ascending=False)
feature_selection_df.index = range(1, len(feature_selection_df)+1)
feature_selection_df
# We take the top 20 features
feature_selection_df.loc[:20,'Features'].values
```

 # Sample code for Random Forest classifier

```
rf=RandomForestClassifier(random_state=42)
rf= rf.fit(X_train, y_train)
prediction = rf.predict(X)
# Evaluation
print(classification_report(y, prediction))
rf.score(X, y)
plot_confusion_matrix(rf, X, y ,cmap='Blues')
```

#Sample code for cross validation-based Evaluation

```
from sklearn.model_selection import cross_val_score

acc=cross_val_score(bgc,X_test,y_test,scoring='accuracy',cv=10,n_jobs=-1)

print("Accuracy Scores of each Fold per iteration")

print("================================================")

fold=1

for i in acc:

    print("Accuracy of Fold",fold,"is:",round(i*100,2))

    print("------------------------------------------------")

    fold=fold+1

print("Mean Accuracy of all Folds",round(acc.mean()*100,2))s
```