Insurance Fraud Detection

# Acknowledgments

We would like to express sincere gratitude to our professor chair Dr. \*\*\*\*\*\* and our supervisor Dr. \*\*\*\*\*\* for providing their invaluable guidance, comments and suggestions throughout the course of this project. We offer our appreciation for the learning opportunities provided by the committee. We should also thank all our course instructors throughout the program without whose guidance and support it would not be possible to undertake solving of a complex analytical problem like fraud detection.

Contents

[Abstract 2](#_Toc103947959)

[Chapter 1: Introduction to the project 3](#_Toc103947960)

[1.1 Introduction: 3](#_Toc103947961)

[1.2 Aim and Objective 4](#_Toc103947962)

[1.3 Problems 5](#_Toc103947963)

[1.4 Research Methodology 6](#_Toc103947964)

[Chapter 2 Related works 7](#_Toc103947965)

[Chapter 3: Methodology 9](#_Toc103947966)

[3.1 Background 10](#_Toc103947967)

[3.2 What is Machine learning? 10](#_Toc103947968)

[3.2.1 Why is ML Important? 11](#_Toc103947969)

[3.2.2 How Does ML Work? 11](#_Toc103947970)

[3.2.3 Machine learning Methods 13](#_Toc103947971)

[What is Active learning? 26](#_Toc103947972)

[3.2.4 The Fraud Detection Problem in ML 27](#_Toc103947973)

[Chapter 4: Data Analysis and Implementation 30](#_Toc103947974)

[4.1 Data Set: 30](#_Toc103947975)

[4.2 Tools used: 30](#_Toc103947976)

[4.3 Data preprocessing 31](#_Toc103947977)

[4.3.1 One hot encoding: 31](#_Toc103947978)

[4.3.2 Missing data handling: 31](#_Toc103947979)

[4.3.3 Normalization: 31](#_Toc103947980)

[4.3.4 Tuning 32](#_Toc103947981)

[4.3.5 Confusion matrix: 32](#_Toc103947982)

[4.4 Implementation 33](#_Toc103947983)

[4.4.1 Import library 33](#_Toc103947984)

[4.4.2 Import dataset 33](#_Toc103947985)

[4.4.3 Dropping missing data 34](#_Toc103947986)

[4.4.4 Performing initial EDA: 34](#_Toc103947987)

[4.4.5 Model 38](#_Toc103947988)

[4.6 Discussion 42](#_Toc103947989)

[Chapter 5: Result analysis 44](#_Toc103947990)

[5.1 Results 44](#_Toc103947991)

[Chapter 6: Conclusion and Future work: 45](#_Toc103947992)

[6.1 Conclusion: 45](#_Toc103947993)

[6.2 Future Work: 45](#_Toc103947994)

Abstract

Payments-related fraud is a major worry for cyber-crime communities, and current analysis has shown that ml algorithms may detect fraudulent activity in massive amounts of data. Such algorithms are capable of detecting fraudulent activity that people auditors may miss, and they can do it in real-time. Using publicly known insurance transaction data, we devote different supervised ml algorithms to the fraud detection problem in this research. We want to demonstrate how supervised ml techniques may accurately classify data with significant class imbalance. Our research shows that LDA is performing well compared to all the classifiers from the ROC curves and results. KNN is performing the worst out of all the classifiers.

*Keywords:* Insurance fraud detection; Machine Learning; Cyber security; Random Forest; KNN

# Lists of Figures

[Figure 1 Methodology structure 7](#_Toc103948084)

[Figure 2 logistic regression curve example 15](#_Toc103948085)

[Figure 3 logistic regression threshold example 15](#_Toc103948086)

[Figure 4 LR working 16](#_Toc103948087)

[Figure 5 LR sbend 16](#_Toc103948088)

[Figure 6 LR equation on graph 17](#_Toc103948089)

[Figure 7 sigmoid curved equation 18](#_Toc103948090)

[Figure 8 Naive bayes structure 19](#_Toc103948091)

[Figure 9 Decision tree framework 22](#_Toc103948092)

[Figure 10 Decision tree example 23](#_Toc103948093)

[Figure 11 LDA 26](#_Toc103948094)

[Figure 12Methodology phases 29](#_Toc103948095)

[Figure 13 screen shot of the dataset 31](#_Toc103948096)

[Figure 14 import libraries 34](#_Toc103948097)

[Figure 15 import dataset 35](#_Toc103948098)

[Figure 16 dataset after missing data 35](#_Toc103948099)

[Figure 17 EDA 36](#_Toc103948100)

[Figure 18 fraud analysis 36](#_Toc103948101)

[Figure 19 heatmap 37](#_Toc103948102)

[Figure 20 categorical data encoding 38](#_Toc103948103)

[Figure 21 one hot encoding 38](#_Toc103948104)

[Figure 22 oversampling using SMOTE 39](#_Toc103948105)

[Figure 23 LR model with output 39](#_Toc103948106)

[Figure 24 KNN model with output 40](#_Toc103948107)

[Figure 25 Decision tree implementation with output 41](#_Toc103948108)

[Figure 26 Random forest implementation with output 42](#_Toc103948109)

[Figure 27 LDA implementation with output 43](#_Toc103948110)

[Figure 28 Result Analysis with ROC 45](#_Toc103948111)

# Lists of Abbreviations

|  |  |
| --- | --- |
| KNN | k-Nearest Neighbors |
| Roc | Receiver Operating Characteristic curve |
| ML | Machine Learning |
| LR | Logistic regression |
| tp | True Positive |
| a | Accuracy |
| tn | True negative |
| NB | Navie Bayes |
| dl | Deep learning |
| fp | False positive |
| SVM | Support Vector Machines |
| Fn | False positive |
| CM | Confusion Matrix |
| LDA | Linear Discriminant Analysis |

# Chapter 1: Introduction to the project

## 1.1 Introduction:

Detection of fraud affects a wide range of businesses, particularly the financial sector, insurance, government organizations, and law enforcement. Fraudulent efforts have grown considerably in recent years, emphasizing the importance of fraud detection (Ko, 2020). Despite the efforts of several institutions, fraud results in the loss of enormous sums of money each year. So the number of fraudulent operations is so tiny that detecting these scams is challenging. (Benedek, 2022).

Fraud may take many forms, from inflated losses to unintentional spending. In the insurance industry, around 10% of claims contain various types of dishonesty, resulting in at least 25% of insurance expenditures (Haneesh, 2018). Because there are so many distinct types of fraud, it's harder to spot them.

Online wallets in varying forms are on the rise all around the entire globe. The amount of transactions handled by payment providers is rapidly increasing. In 2021, PayPal, for example, took $579 billion in total payments. Along with this change, there has been a significant surge in securities fraud in various banking systems. (Al-Hashedi, 2021).

The role of cyberspace & cyber-crime teams is to prevent unwanted online financial fraud. Numerous banks and investment organizations have specialized teams of hundreds of analysts working on automated systems to monitor transaction information via their products and flag those that are possibly fraudulent. As a result, to be effectively and efficiently equipped to handle computer hackers cases, it is critical to investigate how to face the issues of identifying fraudulent required fields in enormous volumes of data (Sheshasaayee, 2019).

## 1.2 Aim and Objective

The objective is to develop a model that can predict which payments are likely to be fraudulent. The standard method of detecting fraud is to create algorithms based on fraud signs. A fraud judgment might be made in two ways based on these heuristics.

In other instances, guidelines would be defined to determine if the matter needed to be investigated.

In other circumstances, a checklist with ratings for the different fraud indications might be created. The sum of these ratings and the claim's worth will decide if the case must be investigated.

This research involved a few months of work creating a framework for detecting fraud in financial transactions. We want the research's conclusion to make it easier to analyze & notice fraudulent transactions.

The project's three essential aims are as follows:

* Research the publications on fraudulent financial detection to better understand the many facets of the problem.
* Employing supervised ML procedures, tackle the concern of financial fraud detection using a publicly available selection dataset.
* To evaluate several categorization approaches to resolve the most appropriate for this application.

The eventual aim is to develop a system and programs that include the analytics and machine learning principles covered in the curriculum. The quality of the categorization findings and the scope of the analysis are critical to the research's triumph. We anticipate that the latest research will offer a baseline for future research & development in this area and a knowledge foundation for students interested in learning about the complexities of fraud prevention.

.

## 1.3 Problems

The most serious problem that insurance providers confront is fraud, which results in massive losses that are often irreversible. Because combating fraud cases, particularly in insurance firms, is difficult, the key objective is to avoid fraudulent acts at all costs. According to reports (Itri, 2019), 22% and 34% of auto insurance claims are believed to be fraudulent, although only 3% of situations are punished. The first step in minimizing fraudulent situations is to discover them, which is complex and just not "expense" because extensive and time-consuming examinations may irritate genuine clients. (Gupta, 2022).

Increasing investigative expenses also make it more difficult to discover fraud. As a result, businesses fail to conduct necessary investigations, resulting in a slew of potential hazards. Because manual fraud detection is costly and inefficient, we now need to examine the fraud before approving the claim. Several artificial intelligence & ML approaches are effective in identifying fraud (Hanafy, 2022).

This project aims to create a predictive model for fraudulent transactions. The difficulty with detecting ML fraud is that scams are far less common than reasonable insurance costs.

Financial fraud detection is difficult due to the variety of fraud types or the lesser amount of declared frauds in standard sampling. Approaches based on machine learning improve prediction accuracy, enabling loss controllers to cover more ground with fewer false positives. The cost of false alerts must be considered in terms of loss avoidance when creating detection methods. (Itri, 2019).

It encompasses a wide range of unethical behaviors that a person may participate in to obtain a fair result from an insurance company. This could include preparing the event, misinterpreting the contexts, including the notable main character and the cause of the incident, & emphasizing the size and scope of the harm.

## 1.4 Research Methodology

In this study, the conventional ML method was used. The tagged class variable in the discovered dataset was used as the machine learning models' prediction variable.

* We examined the data set using creative analysis to discover possible fraud factors.
* We witnessed the distinction between fraud and non-fraud transactions using several visualization tools.
* We used two supervised ML approaches – Logistic Regression and Random Forest – to address the fraud detection problem.
* To remedy the dataset's class imbalance, we also explored under-sampling.
* The models were constructed via cross-validation to minimize overfitting and provide consistent performance.
* Performance metrics such as the Confusion Matrix and ROC were utilized to compare the models' performance.

Jupyter notebook was used to conduct this analysis, which was done in Python. The ML models were run using built-in libraries and techniques. Functions were developed when needed to make specific analyses or visualizations easier.

## 1.5 Thesis Structure

This research is structured as follows:

Introduction to Chapter 1

Chapter 2: Review of Literature: reviews the authors' early investigations and exposes problems in their methodologies.

Methodology (Chapter 3): The paper presents the research approaches for removing dynamic and memory components from analysis findings to do ML. Following a discussion of the ML techniques and evaluation criteria employed,

Chapter 4: Implementation investigates the findings of the investigation.

Chapter 5: Findings and Discussion

Chapter 6: Final Thoughts/ conclusions and Future Plans

.

# Chapter 2 Related works

Fraud detection is an issue that affects numerous businesses, including banking and finance, insurance, government organizations, and law enforcement. It is challenging because the number of fraudulent actions is relatively low. Fraudulent efforts have grown considerably in recent years, emphasizing the importance of fraud detection, regardless of various actions. Annually, substantial quantities of money are lost due to fraud in financial institutions.

In this (Vyas, 2022), authors offers a blockchain-based architecture for a level of readiness and data sharing across the insurance network's numerous interacting actors, Distributed ledgers peer-to-peer interface that allows for validating treatment claims in a secure, immutable, and transparent manner. Additionally, discuss how technology and intelligent models might be combined to better business operations. It will demonstrate how these capabilities may be used to construct a system that eliminates fraud in automotive, healthcare, including life insurance payments, among several other areas. There study will examine the many forms of fraud prevention and detection within insurance claim systems and how they are classified using various machine learning algorithms. In addition, it discusses the future of Fraud Detection inside the Insurance Payout System.

In this (Hansson, 2022), authors categorize the sequence of insurance claims, we propose two versions of LSTM-based ML algorithms. Autoencoders can produce feature significance that provides knowledge into the models' predictions whenever models are put into reality. This method is based on the notion that data outliers are fake. The procedures were developed and tested using a dataset we created using data from just a Swedish insurance firm, with the few flagged frauds used exclusively for analysis and testing. The experiments demonstrate legislature capability, and additional research reveals that the combined effect of autoencoders and LSTMs is efficient yet performs similarly to the baselines used. This research serves as a starting point for practitioners interested in learning essential areas of anomaly detection in fraud detection.

In this (Roy, 2017), authors employing machine learning techniques aims at detecting auto vehicle fraud. In addition, the accuracy will be examined using a confusion matrix. It could aid in the calculation of specificity, recall, & accuracy.

In this (Bhowmik, 2011) , authors study describes a detecting fraud approach that uses data to forecast & evaluate fraud tendencies. They use the Nave Bayesian Identification & Judgement Tree-Based techniques to create classifiers. The method is described briefly and how it may be used to identify fraud. Both strategies make use of the same data. The classifier predictions are analyzed and interpreted. Bayesian Nave Visualization, Decision Tree Visualization, & Rule-Based Categorization are all used to assist prediction accuracy. They test strategies for detecting fraud in the automotive insurer.

In this (Nian, 2016) author Illustrate that a Laplacian matrix's primary non-principal eigenvector is related to a bi-class classification strength metric that can be used to rank anomalies. They proposed SRA delivers an aberration ranking, whether about the class labels or concentrating on two primary patterns by presently only the first non-principal eigenvalues of the Laplacian matrix. The ranking connection can be chosen established on whether the smaller class's cardinality (positive or negative) is sufficiently big. They show that our suggested SRA outperforms unusual conventional case fraud detection approaches using a vehicle insurance claim data set and disregarding labels while producing ranking. Eventually, they offer that, while the proposed SRA performs well for a few resemblance metrics in the car insurance industry.

In this (Itri, 2019), authors focuses on determining the efficacy and verifiability of the most well-known ML algorithms for fraudulent prediction. An anonymously insurance company's automotive data claims were used to test the supervised approach. We want to suggest a method for improving the validity of AI  outcomes. The research concluded that Random-forests outperforms all other techniques tested.

In this (Gomes, 2021), authors This research aims to offer a unique DL approach that uses unstructured variable significance to acquire pragmatic insights into the behaviour of an insured individual. It provides the framework for understanding how minimal effort may be used to get insights into an authorized person's dishonest conduct. They present a new latent inconsistent design using two important unsupervised DL  algorithms, notably fully convolutional or the nonlinear autoencoder, after a preliminary assessment of the limits of conventional fraud investigative techniques. The characteristics of each model are addressed to help the reader understand how models may be customized for fraud protection and how the findings should be interpreted. Achievement assessments are done qualitatively and quantitatively, with a more significant focus on critical review. Various measures are employed to evaluate better the area of knowledge of the fraud detection setup.

# 

# Chapter 3: Methodology

We go over the methodology in detail in this chapter. Section 3.1 background of fraud in an insurance company. Section 3.2 Details analysis of Machine learning algorithms. Section 3.3 describes the steps of the proposed approach and the mechanism for detecting fraud in insurance.

## 3.1 Background

To comprehend the challenge, you must first understand insurance claims forecasting, big data, machine learning, and classification. We investigate the following terms.

There are several supervised machine learning applications. They have focused on the methods required to achieve this in the present study. This is only a brief on categorization issues where the output of instances can only be discontinuous, unordered data (Gilda, 2017).

Machine learning creates a classification that is used to generalize from activities (explanations in a training set) to learn a set of rules. The method of applying supervised ML to solve a massive problem. A large amount of data calculates the likelihood of claims incidence. A claim prediction problem necessitates the use of large data models. Consequently, an efficient approach & a more accurate ML algorithm are required to predict the driver's risk. A program that takes into account the insurance company and the likelihood of applying for a patent in the following year can read and analyze massive datasets comprising tens of thousands of customer details by the insurance provider (Pranavi, 2020).

Kia is a major provider of vehicle insurance. As per Kia (Anon., n.d.), its automobile section aims to create insurance prices relying on the driver's abilities. They think that flourishing methodologies may be used to estimate the incidence of claims in the future year, resulting in more accurate findings. As a result, they created a sample with 70 parameters & 1,888,026 observations 6. These observations contain client data gathered out over many years by the firm.

## 3.2 What is ML?

Many ML modeling methodologies are covered in Machine Learning (ML). A machine learning model predicts outcomes from any digital data by identifying relevant features using statistics. Models may be trained on massive, complex datasets and can improve independently without the need for software upgrades or supervision (Bi, 2019).

Although machine learning has been used in data analysis and related fields for decades, it has never been relevant to languages and social sciences until recently. It usually requires high-level knowledge of computing and mathematics, especially statistics and probability. Many individuals confuse machine learning with data mining since data-mining industries are rife with using learning algorithms to address issues. The learning algorithms are used in the building of various domain models. These learning algorithms, for example, detect abnormalities inside a system, a network, computer vision, e-mail spam, and so on.

ML is a subset of man-made brainpower zeroed in on building frameworks that can gain from authentic information, recognize examples, and pursue legitimate choices with almost no human mediation. An information investigation strategy robotizes the structure of insightful models by utilizing information enveloping assorted types of advanced data, including numbers, words, snaps, and pictures.

The nature of an ML model is reliant upon two significant angles:

**1. The nature of the information**: ML applications gain from the information and consistently work on the precision of results utilizing robotized enhancement techniques. A typical expression for creating ML calculations is "trash in, trash out." The platitude implies that if you put in bad quality or muddled information, the result of your model will be, to a great extent, mistaken.

**2. The model decision itself:** There are plenty of calculations in ML that an information researcher can pick, all with their own particular purposes. It is essential to pick the right calculation for each utilization case. Brain networks are a calculation type with huge publicity around them due to the high precision and flexibility they can convey. Be that as it may, for low information measures, picking an easier model will frequently perform better.

### 3.2.1 Why is ML Important?

ML is filling insignificance because of progressively colossal volumes and various information, the entrance and moderateness of computational power, and the accessibility of the fast Internet. These advanced change factors enable one to quickly and consequently foster models that can rapidly and precisely dissect remarkably huge and complex informational collections (Rajkomar, 2019).

There are many purpose cases that ML can be applied to reduce expenses, relieve gambles, and work on general personal satisfaction, including suggesting items/administrations, recognizing network protection breaks, and empowering self-driving vehicles. With more noteworthy admittance to information and calculation power, ML is becoming more pervasive consistently and will before long be incorporated into numerous features of human existence (Harrison, 2021).

## 3.2.2 How Does ML Work?

There are four key advances you would follow while making an ml model.

**1. Pick and Prepare a Training Data Set**

Preparing information will illustrate the information the AI application will ingest to tune model boundaries. Preparing information is once in a while marked, meaning it has been labeled to get down on orders or expected values the AI mode is expected to foresee. Other preparation information might be unlabeled, so the model should extricate and allot bunches independently.

Information should be partitioned into a preparation and testing subset for names. The previous is utilized to prepare the model, and the last option is to assess the adequacy of the model and track down ways of further developing it.

**2. Select an Algorithm to Apply to the Training Data Set**

The sort of AI calculation you pick will principally rely upon a couple of viewpoints (ltex, 2021):

Whether the utilization case is the expectation of a worth or grouping, which uses marked preparing information or the utilization case is bunching or dimensionality decrease, which utilizes unlabeled preparation information

How much information is in the preparation set? The idea of the issue the model tries to address

You would typically utilize relapse calculations like normal least-square relapse or strategic relapse for expectation or order use cases. You will probably depend on bunching calculations like k-implies or closest neighbors with unlabeled information. A few calculations like brain organizations can be designed to work with bunching and forecast use cases.

**3. Train the Algorithm to Build the Model**

Preparing the calculation involves tuning model factors and boundaries to anticipate the proper outcomes more precisely (Kathleen, 2021). Preparing the AI calculation is normally iterative and uses an assortment of enhancement techniques relying on the picked model. These improvement strategies don't need human intercession, which is important for the force of AI. The machine gains from the information you give it with practically zero explicit courses from the clien.

**4. Use & Improve the Model**

The last advance is to take care of new information to the model for working on its viability and precision over the long haul (admin, 2020). Where the new data will come from relies upon the idea of tackling the issue. For example, an ml model for self-driving vehicles will ingest genuine data on street conditions, articles, and transit regulations.

### 3.2.3 ML Methods

These learning algorithms, for example, are used to detect abnormalities inside a system, a network, computer vision, e-mail spam, and so on (Schrider, 2019).

Below are different types of learning in machine learning? It consists of;

#### What Is Supervised ml?

Administered ML calculations utilize marked information as preparing information, where the relevant results to include information are known. The ML calculation ingests many sources of info and relates the right results (jason, 2016). The calculation contrasts its own anticipated results and the right results to work out model exactness and enhances model boundaries to further develop precision.

Administered ML depends on examples to foresee values on unlabeled information. It is most normally utilized in computerization, over many information records, or in situations where there are many information inputs for people to successfully process. For instance, the calculation can get charge card exchanges that are probably fake or recognize the protection client who presumably will record a case.

In directed learning, the objective is to memorize the mapping (the rules) between a set of inputs and yields. This ordinarily isolates the information into two sets: Preparing and showing prepared testing information (David, 2017). For example, the inputs can be the climate figure, and the yields would be the visitors to the shoreline. The objective of directed learning would be to memorize the mapping that portrays the relationship between temperature and the number of shoreline guests. It is also called the task-driven approach. When installing the model, specific characteristics and classes are chosen for simple learning and recognition, unlike unsupervised learning approaches, which use the supplied data for training to predict future incoming clustering databases.

A side impact to be mindful of in directed learning is that our supervision presents an inclination to the learning. The show can, as it were, copy precisely what it appeared, so it is vital to appear it with solid, impartial illustrations. Moreover, directed learning ordinarily requires a part of information sometime recently it learns. Getting sufficient dependably labeled information is regularly the hardest and most costly portion of utilizing administered learning.

##### Logistic regression:

LR is a measurable strategy utilized to build AI models where the reliant variable is dichotomous: for example, paired. Strategic relapse is utilized to portray information and the connection between one ward variable and at least one autonomous factor. The autonomous factors can be ostensible, ordinal, or of the stretch sort.

The name "Logistic regression" is gotten from the idea of strategic capacity. The calculated capacity is otherwise called the sigmoid capacity. The worth of this calculated capacity lies somewhere in the range of nothing and one (Shipe, 2019).

Coming up next is an illustration of a calculated capacity we can use to find the likelihood of a vehicle stalling, contingent upon how long it has been since it was adjusted last.

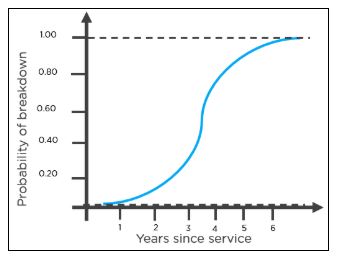


Figure logistic regression curve example .

This is how you can decipher the outcomes from the chart to conclude regardless of whether the vehicle will stall.

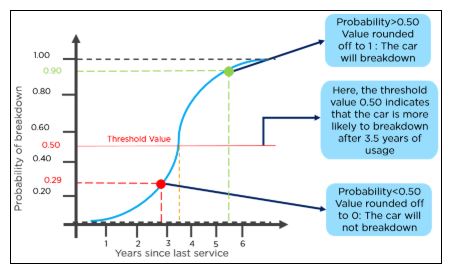


Figure 3 logistic regression threshold example .

**How Does the Logistic Regression LR Algorithm Work?**

Think about the accompanying model: An association needs to decide a representative's compensation increment given their exhibition. For this reason, a straight relapse calculation will assist them with choosing. Plotting a relapse line by thinking about the representative's exhibition as the free factor and the compensation increment as the reliant variable will make their errand more straightforward (Sur, 2019).



Figure LR working .

Presently, imagine a scenario where the association wants to find out whether a representative would get an advancement or not in light of their presentation. The above direct chart won't be reasonable for this situation. Like this, we cut the line at nothing and one and convert it into a sigmoid bend (S bend).

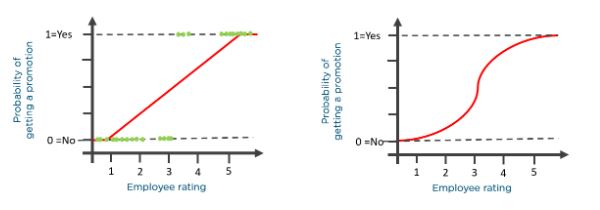


Figure LR sbend .

Given the edge esteems, the association can conclude regardless of whether a representative will get a compensation increment.

How about we go over the chances of progress to figure out calculated relapse.

Chances (𝜃) = Probability of an occasion occurring/Probability of an occasion not occurring

𝜃 = p/1 - p

The upsides of chances range from zero to ∞ , and the upsides of likelihood lie somewhere in the range of nothing and one.

Think about the situation of a straight line:

𝑦 = 𝛽0 + 𝛽1\* 𝑥

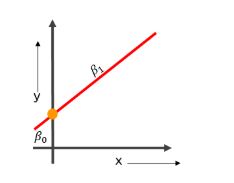
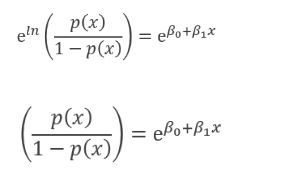
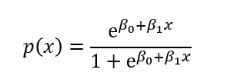


Figure 6 LR equation on graph

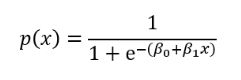
Presently to anticipate the chances of accomplishment, we utilize the accompanying recipe:

log





For sigmoid function the below equation is:



Now sigmoid curved equation:

Figure sigmoid curved equation .

##### Naïve bayes algorithm

It is a technique based on Bayes' theory, the presumption of autonomy amongst indicators.

A Classification Algorithm assumes that the existence of one element in a class is unrelated to the presence of another. It is, indeed, naive! (dmin, 2022).

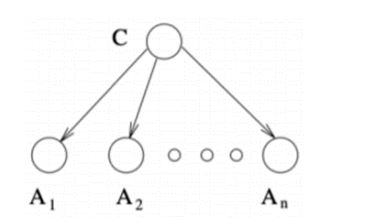


Figure Naive bayes structure .

**How does the Naive Bayes calculation function?**

The calculation initially makes a recurrence table (like earlier likelihood) of all classes and afterward makes a probability table. Then, at long last, it computes the back likelihood.

**Allow us to check out the issue explanation viable:**

The real borders of three varieties of flowers are represented in the Iris informative collection: Versicolor, Setosa, and Virginia. The dataset includes quantitative bounds such as Sepal width, Sepal length, Petal width, and petal length. Given these boundaries, we may predict the classes of the flowers using this information (sunil, 2017). The information consists of continual quantitative attributes depicting individual highlights' components. Regardless of whether these characteristics depend on one another or on other highlights, a Naive Bayes classifier would use these properties to autonomously increase the chance that the bloom belongs to specified animal groups.

An exploratory information examination was performed to enter the significant factors, and one such result is displayed underneath.

##### **KNN**

K-closest neighbor depends on regulated learning strategies. It is viewed as one of the clearest ML models. K-NN expects the examination between the accessible and new information; it remembers the new information for the class closest to the classes accessible. It very well may be utilized for characterization relapse. However, it is predominantly utilized for order (Pranavi, 2020). It is otherwise called a sluggish model since it doesn't gain from the preparation dataset quickly. K-NN modes store the dataset through preparing; when K-NN gets new information, it characterizes this new information to the closest accessible classification given the preparation information in the K-NN model. It can likewise be wasteful computationally. Notwithstanding, K-NNs have prevailed in a few market issues.

The KNN calculation is an expert arrangement strategy for estimating the likelihood that an information point will become an individual from some collection, provided what bunch the information focuses nearest to it belong (onel, 2018).

Then the analysis is a directed ML calculation used to address characterization and relapse difficulties. However, it is largely used for order difficulties.

KNN is a languid learning and non-parametric calculation. It's known as a lethargic learning calculation or the languid student since it plays out no preparation when you supply the preparation information. All things being equal, it simply stores the information during the preparation time and plays out no estimations. It doesn't assemble a model until a question is performed on the dataset.

It's viewed as a non-parametric technique since it makes no suppositions about the hidden information dissemination. Basically, KNN attempts to figure out what bunch an information point has a place with by taking a gander at the data of interest around it.

Consider there are two gatherings, An and B.

To decide if an information point is in bunch An or bunch B, the calculation takes a gander at the conditions of the data of interest close to it. Assuming most of the information focuses are in bunch A, almost certainly, the data of interest is referred to in bunch An and the other way around (admin, 2019).

So, KNN includes grouping a piece of information by looking at the closest clarified data of interest, otherwise called the closest neighbor.

Try not to befuddle K-NN order with K-implies bunching. KNN is a regulated order calculation that characterizes new information and focuses on the closest data of interest. Then again, K-implies bunching is an unaided bunching calculation that bunches information into a K number of groups.

**How does KNN function?**

As referenced over, the KNN calculation is overwhelmingly utilized as a classifier. We should investigate how KNN attempts to group concealed input data of interest.

Unlike arrangements utilizing fake brain organizations, the KNN order is detailed & comfortable to run. It's ideal in possibilities where the statement guides are clear or non-straight.

KNN uses a democratic mechanism to determine the class of a hidden perception. This suggests that the class with the most votes will become the class of the data of significance.

If the value of K is equal to one, we will use just the nearest neighbor to determine the class of data of importance. If the value of K is ten, we'll use the ten nearest neighbors, and so on.

To place that into viewpoint, consider an unclassified information point X. There are a few data of interest with known classes, An and B, in a dispersed plot.

Assume the information point X is set close to bunch A.

As you most likely are aware, we arrange a piece of information by checking the closest clarified calls attention.

For this situation, information point X has a place with bunch An as its closest neighbor is in a similar gathering. If bunch A has more than ten data of interest and the worth of K is equivalent to 10, then, at that point, the information point X will, in any case, have a place with bunch An as all its closest neighbors are in a similar gathering.

Assume one more unclassified information point Y is set between bunch An and bunch B. Assuming K is equivalent to 10, we pick the gathering that gets the most votes, implying that we order Y to the gathering in which it has the most number of neighbors. For instance, if Y has seven neighbors in bunch B and three neighbors in bunch A, it has a place with bunch B. The way that the classifier allocates the classification with the biggest number of votes is valid no matter the number of classes present (Avainash, 2018). You may be considering how the distance metric is determined to decide if an information point is a neighbor or not.

There are four methods for measuring the distance between the data of interest and its closest neighbor: Euclidean distance, Manhattan distance, Hamming distance, and Minkowski distance. Euclidean distance is the most ordinarily utilized distance capacity or metric out of the three.

##### **Decision tree DT**

Decision Tree in ML is a piece of grouping calculation that additionally gives answers for the relapse issues utilizing the order rule (starting from the root to the leaf hub); its design resembles the flowchart where every one of the inward hubs addresses the test on an element (e.g., regardless of whether the varying number is more prominent than a number), each leaf hub is utilized to address the class name( results that should be processed after taking every one of the choices). The branches address combination conjunctions of highlights that lead to the class marks.

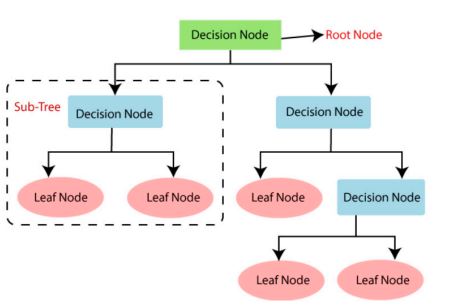


Figure 9 Decision tree framework

DT in Machine Learning has a wide field in the cutting edge world. There are a lot of calculations in ML which is used in our everyday life. One of the significant calculations is the Decision Tree utilized for characterization and an answer for relapse issues. As it is a prescient model, Decision Tree Analysis is done through an algorithmic methodology where an informational index is parted into subsets according to conditions. The actual name says it is a tree-like model on the off chance that else articulations. The more profound the tree and the more the hubs, the better is the model (admin, 2020).

**Kinds of Decision Tree in Machine Learning**

DT is a tree-like diagram where the arranging begins from the root hub to the leaf hub until the objective is accomplished. It is the most well-known one for choice and characterization in light of directed calculations. It is developed by recursive apportioning, where every hub goes about as an experiment for certain properties, and each edge, getting from the hub, is a potential response in the experiment. Both the root and leaf hubs are two substances of the calculation.

We comprehend with the assistance of a little model as follows:

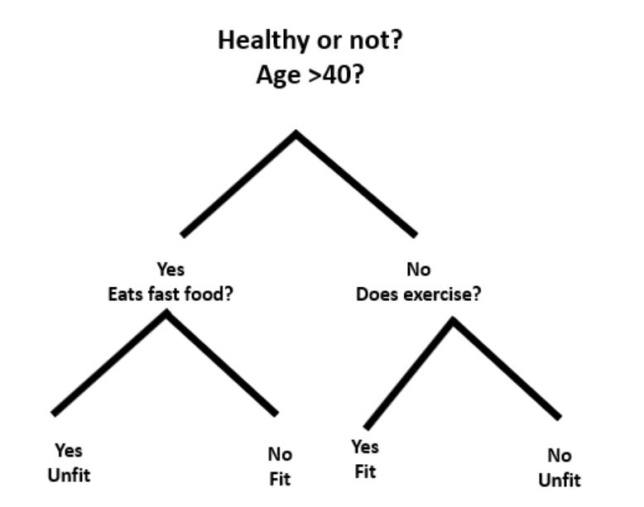
****

Figure 10 Decision tree example

Here, the root hub is regardless of whether you are under 40. Provided that this is true, then, at that point, do you eat cheap food? If you are unsuitable or disaster will be imminent, you are fit. Also, on the off chance that you are more than 40, do you do work out? Assuming this is the case, you are fit, or something bad might happen, you are unsuitable. This was essentially a paired arrangement.

There are two sorts of Decision Trees:

**Classification Trees:** The above model is a categorical-based Classification Tree.

**Regression Trees:** In this calculation, the choice or result is persistent. It has a solitary mathematical result with additional information sources or indicators.

In the Decision tree, the normal test recognizes the quality at every hub. The interaction is called property choice and has a few measures to recognize the trait.

##### **Random Forest RF**

RF is a continuous ML computation that may be used for various assignments such as relapse and order. It is a group approach, meaning that an arbitrary forest model is made up of a large number of little choice trees, termed assessors, each producing its own expectations. The arbitrary timberland model merges the projections of the judges to offer a more exact expectation (tony, 2019)

Standard choice tree classifiers have the drawback that they have inclined to overfit the preparation set. The arbitrary woodland's gathering configuration permits the irregular timberland to make up for this and sum up well to concealed information, incorporating information with missing qualities (admin, 2020). Arbitrary timberlands are likewise great at taking care of huge datasets with high dimensionality and heterogeneous component types (for instance, assuming one section is straight out and another is mathematical).

Irregular backwoods are generally excellent for order issues yet are somewhat less great at relapse issues. Rather than direct relapse, an irregular woodland regressor can't make expectations outside the scope of its preparation information.

Arbitrary timberlands are additionally secret elements: it is hard to peer inside an irregular woodland classifier and figure out the thinking behind its choices rather than some more conventional AI calculations. Likewise, they can be delayed in preparing and running and produce enormous record sizes.

Since they are incredibly vigorous, simple to begin with, great at heterogeneous information types, and have not very many hyperparameters, irregular woodlands are much of the time an information researcher's most memorable port of call while fostering another AI framework, as they permit information researchers to get a quick outline of what sort of precision can sensibly be accomplished on an issue, regardless of whether the prior arrangement may not include arbitrary timberland.

Random forests mirror a shift to the packed away choice trees that make a vast number of de-corresponded trees with the goal that prescient proficiency can be worked on further. They are an extremely famous "off-the-crate" or "off-the-rack" learning calculation with great prescient

execution and few hyper-boundaries. There are numerous arbitrary timberland executions. Nonetheless, the Leo Breiman calculation is generally legitimate. Arbitrary timberlands make a prescient worth because of the relapse of individual trees. It makes plans to over-fit.

There are various variations of the RF calculation. However, the most generally involved adaptation today depends on Leo Breiman's 2001 paper, so we will follow Breiman's execution.

Allow us to expect we have a preparation set of N preparing models, and for every model, we have N highlights. An arbitrary backwoods will comprise of Ntree choice trees or assessors.

**1. Sacking**

Given an N-model sample, we repeatedly generate example subsets of the informational index of size n, where n is less than N. Examining is done indiscriminately but with substitution. This method of subsampling a prepared set is known as bootstrap totaling or dumping.

**2. Arbitrary subspace strategy**

Assuming each preparing model has M elements, we take a subset of them of size m < M to prepare every assessor. So no assessor sees the full preparation set; every assessor sees just m elements of n preparing models.

**3. Preparing assessors**

We make Ntree choice trees or assessors and train everyone on an alternate arrangement of m elements and n preparing models. The trees are not pruned, as they would be on account of preparing a straightforward choice tree classifier.

**4. Perform derivation by totaling forecasts of assessors**

To make an expectation for another approaching model, we pass the important highlights of this guide to every one of the three assessors. We will get Ntree forecasts, which we want to join to create the general expectation of the arbitrary timberland. On account of arrangement, we will utilize a greater part casting a ballot to settle on the anticipated class, and on account of relapse, we will take the mean worth of the forecasts of the multitude of assessors.

##### **Linear Discriminant Analysis LDA:**

In 1936, Ronald A.Fisher figured out Linear Discriminant for the first time and showed a few down-to-earth uses as a classifier. LDA is the most regularly involved dimensionality decrease strategy in administered learning. Fundamentally, it is a preprocessing venture for design order and AI applications (Ali, 2019).

It projects the dataset into moderate layered space with a certifiable class of distinguishable highlights that limit over fitting and computational expenses. With the means to characterize objects into one of at least two gatherings, given some arrangement of boundaries that portrays objects, LDA has thought of explicit capacities and applications; we will find out about that exhaustively in the approaching segments. Under LDA, we are fundamentally searching for

* Which set of boundaries can best depict the relationship of the gathering for an item?
* What is the best characterization preceptor model that isolates those gatherings?

It is generally utilized for demonstrating assortments in gatherings, for example, appropriating factors into at least two classes, assuming we have two classes, and we really want to proficiently arrange them (priyakun, 2021).

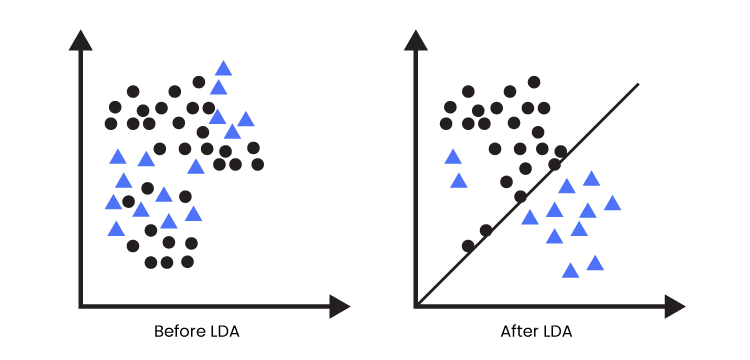


Figure LDA .

Classes can have numerous elements, and utilizing one single component to characterize may yield in a covering of factors of some sort or another, so there is a need to expand the number of highlights to try not to cover that would bring about appropriate order consequently.

#### What Is Unsupervised Machine Learning

Solo ML is best applied to information that doesn't have an organized or objective response. There is no pre-assurance of the right result for the given information. All things considered, the calculation should figure out the information and structure the suitable choice (Sanatan, 2019). The point is to inspect the data and recognize the structure inside it.

Unaided ML functions admirably on value-based data. For instance, the calculation can distinguish client portions that have comparative ascribes. Clients inside these portions can then be designated by comparative showcasing efforts. Famous strategies utilized in unaided learning incorporate closest neighbour planning, self-putting together guides, particular worth decay and k-implies grouping. The calculations are used to portion themes, distinguish anomalies and suggest things (Li, 2020).

It has been made feasible for us (people) to supply an enormous arrangement of information to a PC to learn designs so it might figure out how to settle on a choice when confronted with another occasion or occurrence.

#### What is Ensemble learning?

It involves the addition of label data, similar to supervised learning, but with the addition of many models to solve the problem. Classification is the most common application of this term.

### What is Active learning?

It functions as a teacher, assisting in the correction of errors and behaviour during natural changes. It's possible that it's a subcategory of assist learning.

#### What is Semi-super user learning?

When there is a dataset with a small amount of labelled data, it tries to mix both supervised and unsupervised approaches.

### 3.2.4 The Fraud Detection Problem in ML

In Machine Learning wording, issues, for example, the Fraud Detection issue might be outlined as a characterisation issue, of which the objective is to foresee the discrete name 0 or 1 were 0, for the most part, propose that an exchange is non-fake and 1 recommend that the exchange is by all accounts false.

Consequently, this issue expects experts to assemble models that are savvy to precisely recognise fake and non-false exchanges given different clients' exchange information — which is frequently unknown to safeguard clients' protection.

Since relying only upon rule-based frameworks isn't the best methodology, Machine learning has been the methodology numerous monetary establishments take to battle the issue.

What makes this issue (misrepresentation discovery) so testing is that when we model it in reality, most of the exchanges that happen are veritable exchanges, and just a tiny piece represents the fake way of behaving. This implies we manage the issue of imbalanced information.

However, our research will be on starting our machine learning framework to identify fraud in the insurance dataset.

The project's deliverables were based on this technique. It discusses each step's outcomes and Compares them to decide which method best addresses the fraud detection challenge.

The findings from each step of the project are described in the output for that phase. The following are the deliverables that were used in this final project:

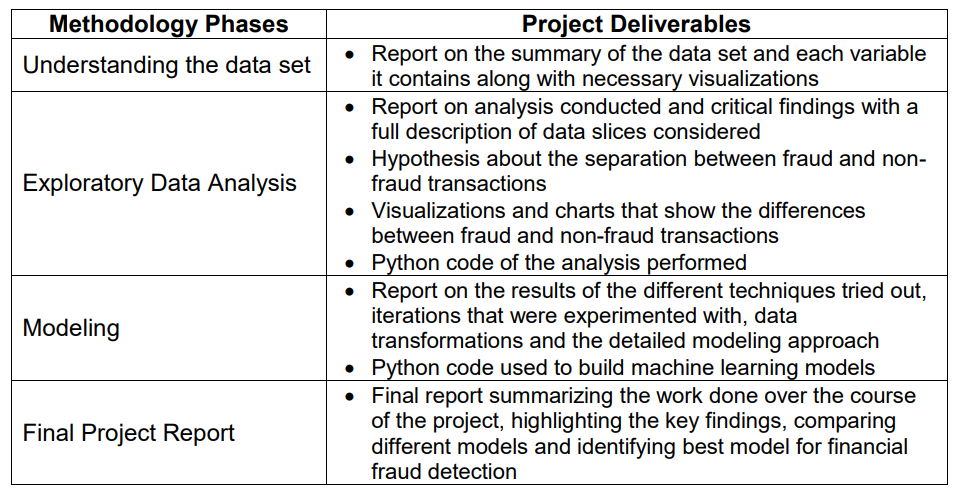


Figure Methodology phases

# 

# Chapter 4: Data Analysis and Implementation

This section goes through each phase of the research in-depth. All studies are recorded in Jupyter notebook format, and the code is included with the results.

## 4.1 Data Set:

In this research, I am using an open-source car insurance dataset available on Github.

The dataset contains 40 columns and 1001 rows of data. The key columns available are :-

* Months
* Policy blind date
* Age
* Policy state
* Policy csl
* Policy number and so on

The image below shows a snapshot of the first few lines of the data set.

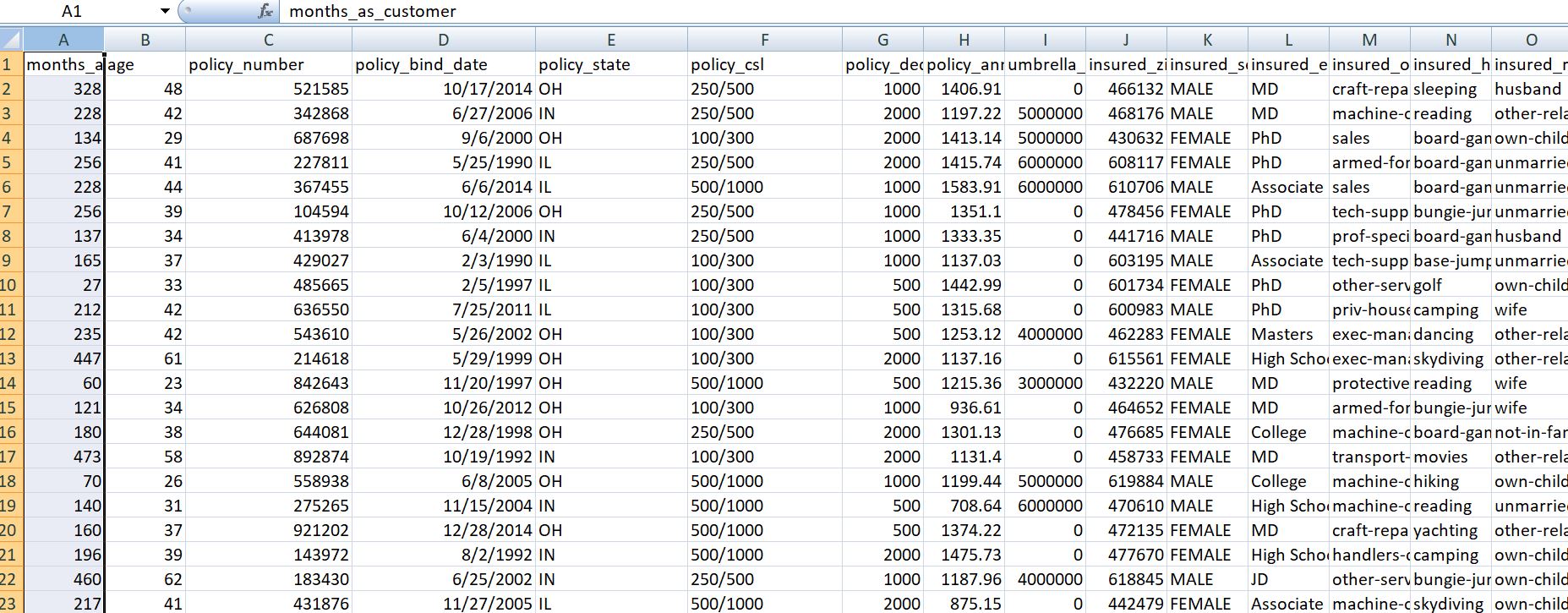


Figure screen shot of the dataset

## 4.2 Tools used:

This implementation was completely done utilizing Python, and the investigation was reported in a Jupyter notebook. Standard python libraries were utilized to lead various investigations. These libraries are portrayed beneath:

* Pandas
* Sklearn
* Numpy
* Matplotlib
* Seaborn

## 4.3 Data preprocessing

The ml technique requires various data pretreatment and parameter tweaking procedures for best performance (Alexandropoulos, 2019). I'll go through some of the typical methodologies and issues explored in subsequent sections of the thesis.

### 4.3.1 One hot encoding:

Numerous methods handle numerical input, so quantitative categories are seen as values on a scale that measures (Rodríguez, 2018). One-hot encoding (also known as dummies encoding in demographics) entails assigning a value of 1 if the attribute is true for the observations and a value of 0 if it is not. This approach often widens and shortens the dataset, resulting in more characteristics and fewer events (Rodríguez, 2018).

### 4.3.2 Missing data handling:

For example, missing information, not all perceptions having values for all elements, is a typical issue of importance. There is no perception for all highlights in each example. The standard techniques to deal with this issue are either eliminating the perceptions of missing information or ascribing values to them (Emmanuel, 2021). Eliminating perceptions with missing information prompts more adjusted datasets; however, it decreases how much preparation information is accessible and causes the data put away in the present qualities for elements of the perception to be lost. Attributing information with some gauge, including mean or middle (or utilizing prescient instruments to gauge the worth), is frequently the favored arrangement. For the straight out factors, one-hot encoding reduces missing information issues, and "information on include x is absent" can be made its own variable whenever considered important data.

### 4.3.3 Normalization:

A huge gap among various dimensions characteristic data within the dataset might cause issues such as sluggish model training and minimal accuracy increase; so, to tackle this issue, the MinMaxScaler was used to shift the dataset into the band of (0,1) as follows (Wu, 2018):

Additionally, standardization and scaling or normalization are utilized to forestall inclination because highlights have various scales. There are numerous standardization strategies, and the decision of technique can have a huge impact on the model. A straightforward model is a min-max technique where the base worth of an element is set to be the least of the new scale, for example, 0, and the same for the most extreme, for example, 1.

According to this new scale, the middle between is then changed from zero to one. Normalization additionally standardizes the circulation of information, so the various elements have a similar standard deviation.

### 4.3.4 Tuning

Boundary tuning implies adjusting the boundaries of a calculation as indicated by a particular rule, for example, further developed exactness. Dependable guidelines can be utilized, yet they require insight from applying the calculations to datasets with various properties or believing rules set by another person. A manual hunt is likewise conceivable. However, it also requires a beginning stage and could demonstrate slow. In turn, one would have to transform each boundary, train the model, and keep a rundown of results from various boundary blends. Framework search is a technique that permits testing for different boundaries and their qualities at once via preparing various models and featuring the best arrangement and its boundaries. In quintessence, it implies preparing models with various choices for one boundary while maintaining different boundaries steady. The prepared models are assessed to track down the boundary limiting a picked misfortune work (for instance, blunder rate). Boundaries found in lattice search are then set as constants, and the same routine is performed for the next boundary until all cared about boundaries are improved. Huge matrix looks are computationally ineffectual, and for the most part, going through every one of the options isn't achievable.

### 4.3.5 Confusion matrix:

The confusion matrix shapes the premise point for assessing the parallel grouping frameworks. An in-pairs lattice presents the quantity of accurately and erroneously arranged perceptions versus the genuine appropriation of classes in the test set. The issue of paired order can be introduced: does perception have a place with a class? This permits the portrayal of perceptions to up-sides (has a place with the class) and negatives (doesn't have a place with the class) (Luque, 2019).

Precision a, Preparing Time tt, the total time required to prepare a classifier, Specificity s, and Expectation Time pt, the total time required for computation to predict all of the information, were all examined inside the suggested system. The tp esteem refers to the suitable identifiable proof of an assault, the fp esteem to incorrectly differentiated assaults, the tn esteem to accurately identified ordinary associations, and the fp esteem to the number of assaults correctly recognized. Here is who we're talking to.

Accuracy = a

True ^ Positive = tp

False ^ Positive = fp

True ^ Negative = tn

False ^ negative = fn

## 4.4 Implementation

### 4.4.1 Import library

I am using Jupyter Notebook, & can create a local machine by directly downloading the TensorFlow backend.

! Pip install tensorflow

! pip keras pip install

First i import all libraries in jupyter note book.

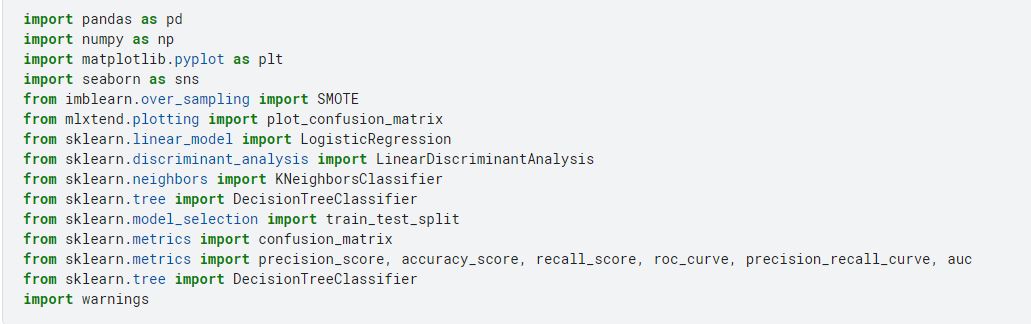


Figure import libraries

### 4.4.2 Import dataset

Now, I import dataset in jupyter notebook.

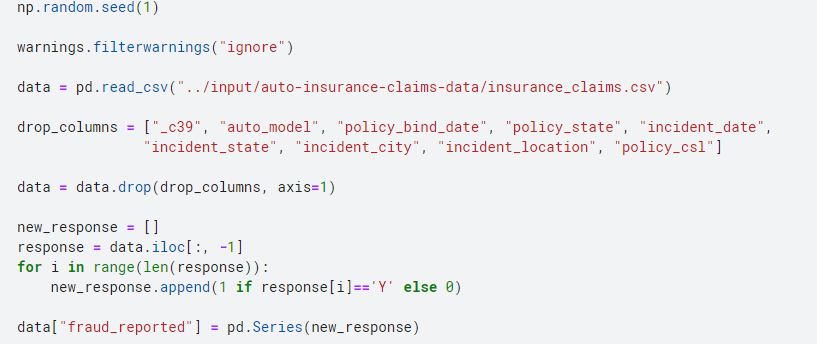


Figure import dataset

### 4.4.3 Dropping missing data

I will be dropping insignificant features like:

* policy\_bind\_date
* policy\_state
* incident\_date
* auto\_model
* \_c39
* policy\_csl

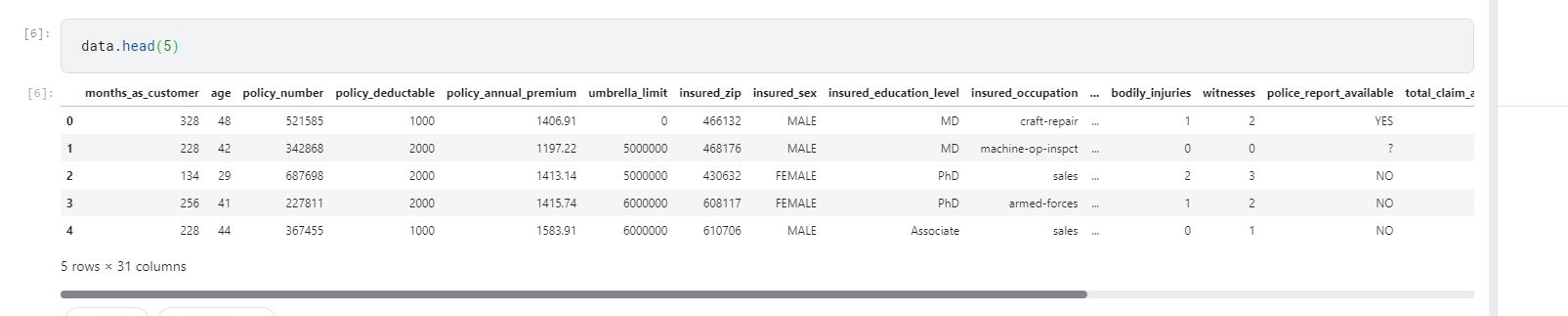


Figure dataset after missing data

### 4.4.4 Performing initial EDA:

Information researchers use EDA to deconstruct and analyze informative collections and summarise their main characteristics, typically using information representation approaches. It determines how to best regulate information sources to get the solutions you need, making it easier for information researchers to find designs, detect anomalies, test a hypothesis, or examine assumptions (nist, 2021).

Developed in the 1970s by American mathematician John Tukey, EDA techniques are now a widely used strategy in the information disclosure process. EDA is mostly utilized to determine what information can be uncovered beyond the practical presenting or hypothesis testing task. It provides a better understanding of information collecting variables and their relationships. Similarly, it might aid in determining whether the measured processes you are considering for information research are appropriate.



Figure EDA

I now analysis the fraud

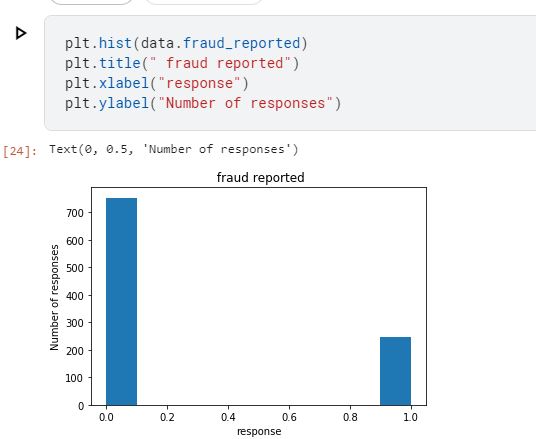


Figure fraud analysis

I can also see that there is a significant class imbalance, we will be using SMOTE, Synthetic Minority Oversampling Technique to add additional minority class data points.

#### SMOTE:

The Synthetic Minority Oversampling Technique is among the most often used oversampling strategies for dealing with irregularities. It plans to modify class distribution by randomly increasing minority class models and rebuilding them (swastik, 2020).

Destroyed intersperses fresh minority experiences with old minority situations. It develops virtual preparation records for the minority class by direct intervention. These engineering preparation records are generated by randomly selecting at least one of the known for each model in the minority class. Following the frame interpolation system, the material is reproduced, and some characterization models for the handled data may be applied.

#### Heatmap

A heatmap is a visual illustration of data in two dimensions. Each data value is represented by a matrix. It will demonstrate the connection between direct and indirect traits. To begin, create a pair plot of all independent and dependent variables (Liu, 2019). Use the independent feature to create a model if the connection between the independent & dependent characteristics is less than 0.2.

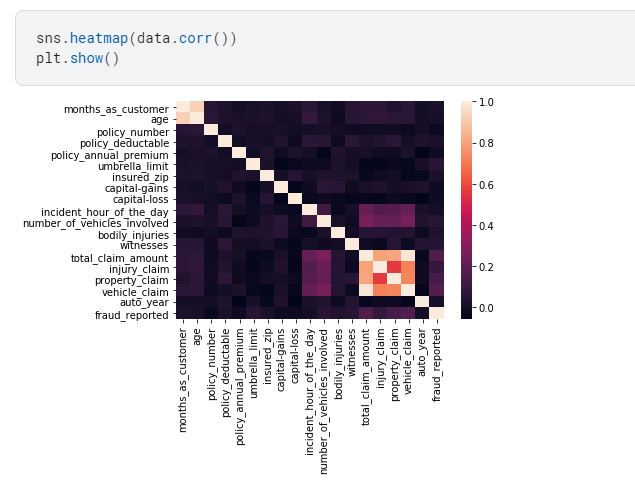


Figure heatmap

#### Categorical dataset:

All-out information is a kind of information utilized to bunch data with comparative qualities, while mathematical information is a sort of information that communicates data as numbers (Qu, 2018).

**For what reason does we really want categorical data encoding?**

Most ML calculations can't deal with unmitigated factors except if we convert them to mathematical qualities. Many calculation exhibitions even change given how the clear cut factors are encoded

**Downright factors can be partitioned into two classifications:**

* Normal: no specific request
* Ordinal: there is some request between values

I implement categorical data encoding in our research because since most of our data is categorical, I have two options, assign an integer value to each level of the categorical variable or one-hot encode these categorical variables. One major drawback of assigning an integer value to each level is adding additional characteristics to the data. For example, let's say I have a variable with levels such as BMW, Mazda, Mercedes, and Subaru, and I are assigning 0, 1, 2, and 3 integer values to them, respectively. When i apply any model, the model considers these values as continuous and assumes an unwanted hierarchy like BMQ < Mazda < Mercedes < Subaru, which might not be the case.

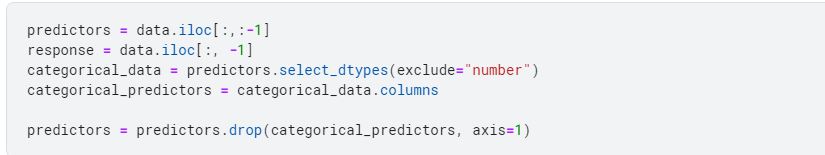


Figure categorical data encoding

Now, I perform one hot encoding:

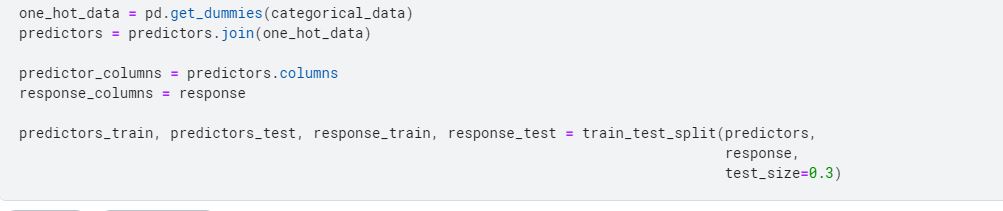


Figure one hot encoding

Since I have class imbalance in the data, I perform minority class oversampling using SMOTE, Synthetic Minority Oversampling Technique, which uses K nearest neighbors to come up with new samples in the minority class.

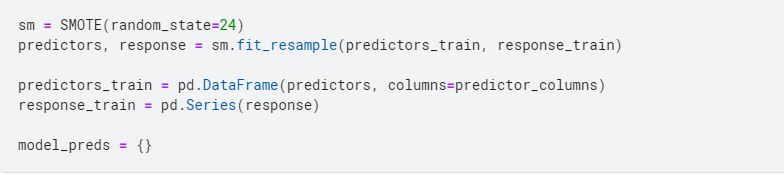


Figure oversampling using SMOTE

### 4.4.5 Model

First, i implement

#### 1. logistic regression

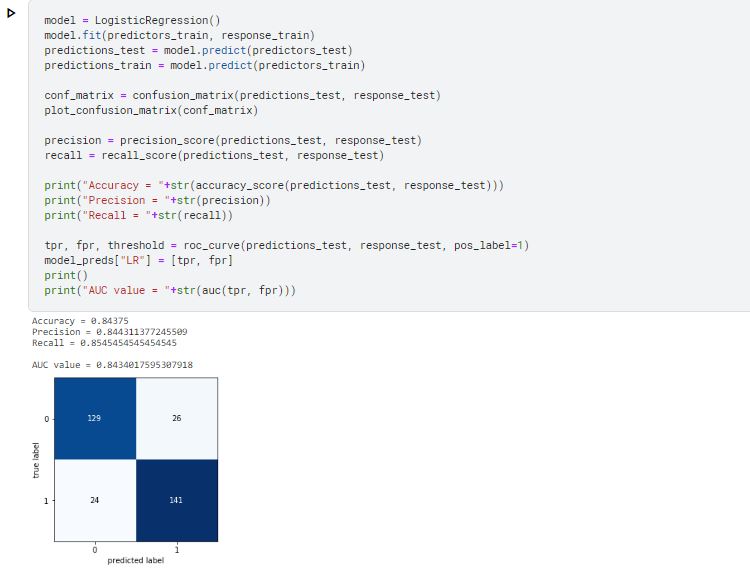


Figure LR model with output

#### 2. KNN

Now I implement KNN model



Figure KNN model with output

#### 3. DT (Decision Tree)

I implement decision tree model. Since it has a lot of categorical variables and the dataset is also not huge, I will use decision trees to get more accuracy.

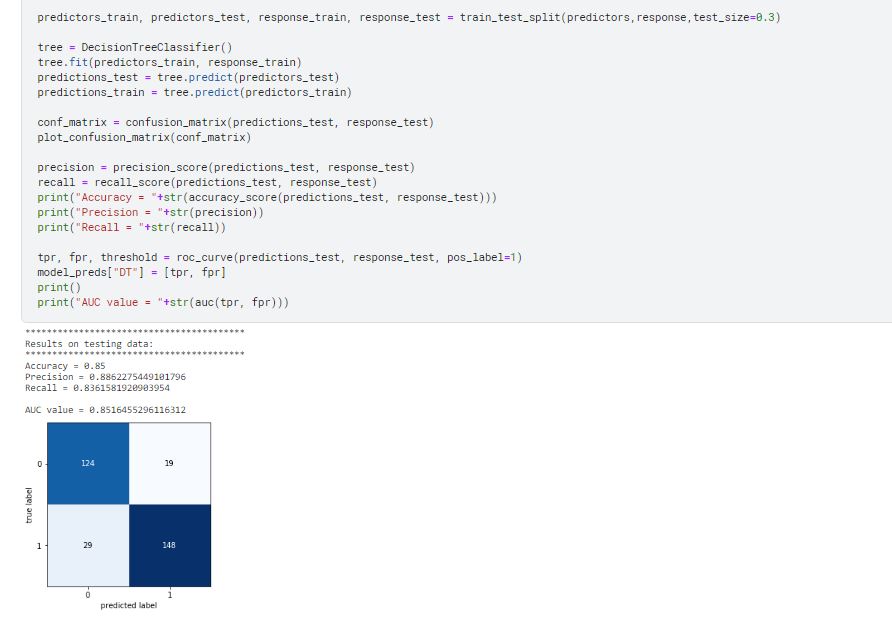


Figure 25 Decision tree implementation with output

#### 4. Random forest classifier

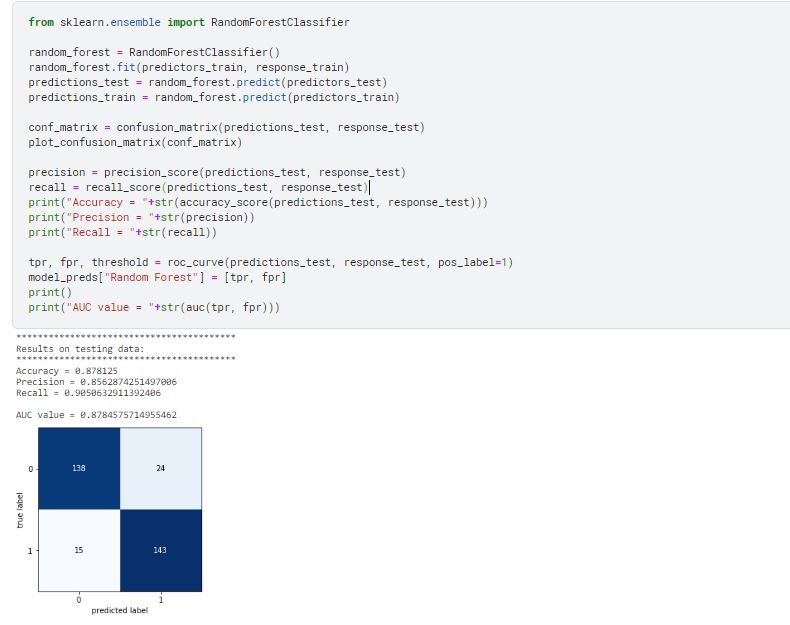


Figure 26 Random forest implementation with output

#### 5. Linear Discriminant Analysis

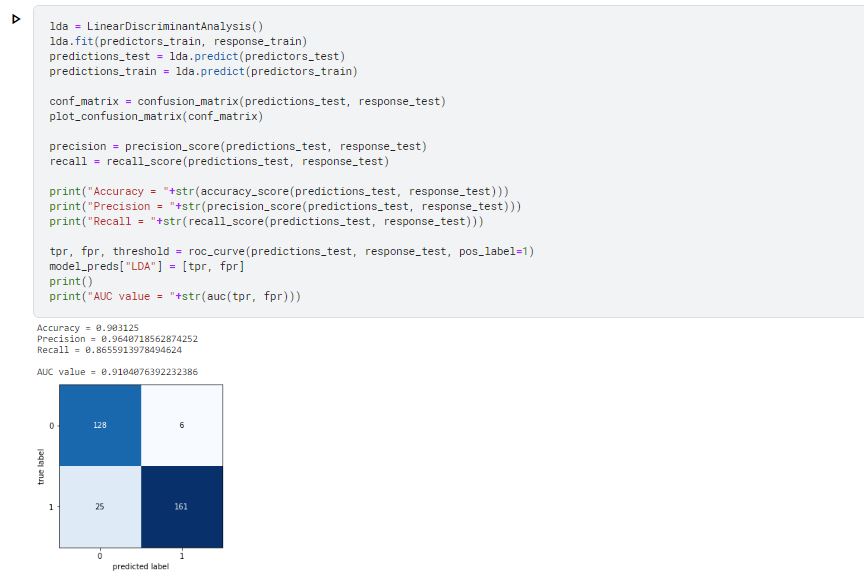


Figure 27 LDA implementation with output

## 4.6 Discussion

The parameter tweaking processes enhanced each of the basic models. The cross-validated AUC ratings for many models were shockingly low. I anticipated a greater range in performance amongst the models. Symptoms of All models had overfitting since the in-sample scores were much higher than the cross-validated results. If the overfitting assumption is valid, the evaluation Test-data scores should be lower than cross-validated scores. Overall, the model. The outcomes of the development phase are optimistic since the scores indicate that each algorithm performs well using data acquired with minimum feature selection, better than random guessing.

# Chapter 5: Result analysis

I begin by presenting and discussing the confusion matrices C, evaluating the Receiver Operating Characteristic curves, & summarising the key assessment measures. This section provides the findings of the final models LR, KNN, RF, DT, and LDA. The models' interpretability is detailed in a separate section after this chapter.

## 5.1 Results

As I can see from the above ROC curves and results, LDA performs well compared to all the classifiers. KNN is performing the worst out of all the classifiers. I was hoping to get better results with Random Forests, but with the data size, I am not surprised with this result.

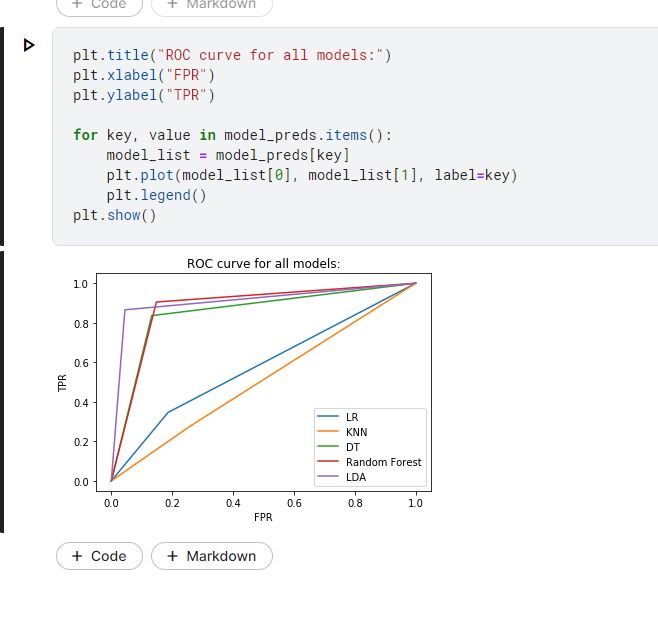


Figure 28 Result Analysis with ROC

# Chapter 6: Conclusion and Future work:

This section will examine the implications of the outcomes, their limits, and further exploration headings. Based on results achieved in this theory, ML techniques can be utilized to make protection guarantee risk scores from client risk information in the car insurance company. The outcomes achieved for the situation concentrated on the present in this proposal suggest that the LDA strategy would be generally reasonable for this undertaking. My outcomes line up with those who tracked down LDA to fit well to car protection guarantee information for misfortune cost forecast. In any case, the materialness of various techniques is a lot of ward on the information construction and meaning of the anticipated classes, in this manner the positioning of the calculations can't be summed up to other datasets. The outcomes infer that all of them tried appropriate strategies for the grouping task. Had comparative outcomes with car information utilizing calculated relapse, brain organization, and rare woods classifiers; as far as ROC, they generally accomplished comparative accuracy on a scale. The positioning of calculations changed with each unique element determination strategy utilized.

## 6.1 Conclusion:

In conclusion, we effectively enabled a structure for recognizing pretend exchanges in car insurance information. This design will assist with figuring out the subtleties of extortion location, for example, the formation of determining factors that might be useful to isolate the classes, addressing class imbalance & choosing the right ml algorithm.

We explored different ml algorithms - the LDA model gave better outcomes than Logistic Regression showing tree-based calculations function admirably for exchanging information with well-differentiated classes. We derived a few features that differentiated the classes better than the raw data through this exploratory analysis.

## 6.2 Future Work:

It would be useful to see close analyses of other Machine learning & deep learning techniques using this dataset; it would also be worth achieving this research with another insurance dataset to finalize whether LDA still has the best predictive output.

# References:

admin, 2019. [Online]   
Available at: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

admin, 2020. [Online]   
Available at: https://metamind.readme.io/docs/use-feedback-to-improve-a-model

admin, 2020. [Online]   
Available at: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

admin, 2020. [Online]   
Available at: https://hbr.org/1964/07/decision-trees-for-decision-making

Alexandropoulos, S.-A. N. S. B. K. a. M. N. V., 2019. Data preprocessing in predictive data mining.. *The Knowledge Engineering Review 34 .*

Al-Hashedi, K. a. M., 2021. 2021. Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019.. *Computer Science Review, 40, p.100402..*

Ali, L. C. Z. Z. Z. a. Y. L., 2019. Automated detection of Parkinson’s disease based on multiple types of sustained phonations using linear discriminant analysis and genetically optimized neural network.. *IEEE Journal of Translational Engineering in Health and Medicine 7 (.*

Anon., n.d. [Online]   
Available at: https://www.kiamotorssouth.com/services-parts?gclid=CjwKCAjwqauVBhBGEiwAXOepkVYvASDtbBlJBmFGoax\_bfXl9ZaX7tZMgOyOgZ2UDcbdtt0I-KhlShoCCVcQAvD\_BwE

Avainash, 2018. [Online]   
Available at: https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn

Benedek, B. C. C. a. B. Z. N., 2022. Automobile insurance fraud detection in the age of big data–a systematic and comprehensive literature review.. *Journal of Financial Regulation and Compliance.*

Bhowmik, R., 2011. Detecting auto insurance fraud by data mining techniques. *ournal of Emerging Trends in Computing and Information Sciences 2, no. 4 (2011):.*

Bi, Q. K. E. G. J. K. a. J. L., 2019. What is machine learning? A primer for the epidemiologist.. *American journal of epidemiology 188, no. 12 (2019): 2222-2239..*

David, 2017. [Online]   
Available at: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861

dmin, 2022. [Online]   
Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/naive-bayes-classifier

Emmanuel, T. T. M. D. M. T. S. B. M. a. O. T., 2021. A survey on missing data in machine learning.. *ournal of Big Data 8, no. 1.*

Gilda, S., 2017. Notice of Violation of IEEE Publication Principles: Evaluating machine learning algorithms for fake news detection.. *In 2017 IEEE 15th student conference on research and development (SCOReD).*

Gomes, C. Z. J. a. H. Y., 2021. Insurance fraud detection with unsupervised deep learning.. *ournal of Risk and Insurance 88.3.*

Gupta, A. a. M. C. L., 2022. Comparative Analysis of Numerous Approaches in Machine Learning to Predict Financial Fraud in Big Data Framework.. *Soft Computing: Theories and Applications. Springer, Singapore, 2022. 107-123..*

Hanafy, M. a. R. M., 2022. Classification of the Insureds Using Integrated Machine Learning Algorithms: A Comparative Study.. *Applied Artificial Intelligence .*

Haneesh, I. M. S. a. K. N. S., 2018. A Study on Insurance Amount Prediction from Classifying the Cars.. *IMPACT: International Journal of Research in Engineering & Technology 9, no. 2: 31-36..*

Han, N. J. W. X. F. J. W. S. Z. S. X. a. X. L., 2020. Transferable linear discriminant analysis.. *IEEE Transactions on Neural Networks and Learning Systems 31, no. 12 .*

Hansson, A. a. H. C., 2022. Insurance Fraud Detection using Unsupervised Sequential Anomaly Detection..

Harrison, C. J. a. C. J. S.-G., 2021. Machine learning in medicine: a practical introduction to natural language processing.. *BMC Medical Research Methodology 21, no. 1.*

Itri, B. M. Y. M. Q. &. O. B., 2019. Performance comparative study of machine learning algorithms for automobile insurance fraud detection.. *In 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS) (pp. 1-4). IEEE..*

Itri, B. Y. M. Q. M. a. B. O., 2019. Performance comparative study of machine learning algorithms for automobile insurance fraud detection. *In 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS.*

Itri, B. Y. M. Q. M. a. B. O., 2019. Performance comparative study of machine learning algorithms for automobile insurance fraud detection.. *In 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), pp. 1-4. IEEE.*

jason, 2016. [Online]   
Available at: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

Kathleen, 2021. [Online]   
Available at: https://www.techtarget.com/searchenterpriseai/feature/How-to-build-a-machine-learning-model-in-7-steps

Ko, N. Y. S. H. R. A. W. a. G. S. C., 2020. Association of insurance status and racial disparities with the detection of early-stage breast cancer.. *JAMA oncology 6, no. 3 (2020): 385-392..*

Li, N. M. S. a. Y. G., 2020. A systematic review of unsupervised learning techniques for software defect prediction.. *Information and Software Technology 122 (2020): 106287..*

Liu, D. Z. X. Y. Z. a. C. F., 2019. Heat map visualisation of fire incidents based on transformed sigmoid risk model.. *Fire Safety Journal 109.*

ltex, 2021. [Online]   
Available at: https://www.altexsoft.com/blog/datascience/preparing-your-dataset-for-machine-learning-8-basic-techniques-that-make-your-data-better/

Luque, A. A. C. A. M. a. A. d. L. H., 2019. The impact of class imbalance in classification performance metrics based on the binary confusion matrix.. *Pattern Recognition 91.*

Nian, K. H. Z. A. T. T. C. a. Y. L., 2016. Auto insurance fraud detection using unsupervised spectral ranking for anomaly.. *The Journal of Finance and Data Science 2, no. 1 .*

nist, 2021. Issue https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm.

onel, 2018. [Online]   
Available at: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Pranavi, P. S. H. D. S. S. S. K. S. K. a. B. H. S., 2020. Analysis of Vehicle Insurance Data to Detect Fraud using Machine Learning.. *International Journal for Research in Applied Science & Engineering Technology (IJRASET) 8, no. 7.*

Pranavi, P. S. S. H. D. K. S. S. K. S. &. S. B. H., 2020. Analysis of Vehicle Insurance Data to Detect Fraud using Machine Learning.. *International Journal for Research in Applied Science & Engineering Technology (IJRASET), 8(7).*

priyakun, 2021. [Online]   
Available at: https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning

Qu, Y. B. F. W. Z. R. T. M. N. H. G. Y. Y. a. X. H., 2018. Product-based neural networks for user response prediction over multi-field categorical data.. *ACM Transactions on Information Systems (TOIS) 37, no. 1.*

Rajkomar, A. J. D. a. I. K., 2019. Machine learning in medicine.. *New England Journal of Medicine 380.14 .*

Rodríguez, P. M. A. B. J. G. a. S. E., 2018. Beyond one-hot encoding: Lower dimensional target embedding.. *Image and Vision Computing 75.*

Rodríguez, P. M. A. B. J. G. a. S. E., 2018. Beyond one-hot encoding: Lower dimensional target embedding.". *mage and Vision Computing 75 (2018): 21-31..*

Roy, R. a. K. T. G., 2017. Detecting insurance claims fraud using machine learning techniques.. *In 2017 international conference on circuit, power and computing technologies (ICCPCT), .*

Sanatan, 2019. [Online]   
Available at: https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a

Schrider, D. R. a. A. D. K., 2019. Supervised machine learning for population genetics: a new paradigm. *Trends in Genetics 34, no. 4.*

Sheshasaayee, A. a. S. S. T., 2019. A purview of the impact of supervised learning methodologies on health insurance fraud detection.. *In Information Systems Design and Intelligent Applications.*

Shipe, M. E. S. A. D. F. F. a. E. L. G., 2019. Developing prediction models for clinical use using logistic regression: an overview.. *Journal of thoracic disease 11, no. Suppl 4 .*

sunil, 2017. [Online]   
Available at: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Sur, P. a. E. J. C., 2019. A modern maximum-likelihood theory for high-dimensional logistic regression.. *Sur, Pragya, and Emmanuel J. Candès. "A modern maximum-likelihood theory for high-dimensional logistic regression." Proceedings of the National Academy of Sciences 116, no. 29.*

swastik, 2020. Issue https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/.

tony, 2019. [Online]   
Available at: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Vyas, S. a. S. S., 2022. Fraud Detection in Insurance Claim System: A Review.. *In 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), pp. 922-927.*

Wu, Y. a. K. H., 2018. Group normalization. *In Proceedings of the European conference on computer vision (ECCV), pp. 3-19.*

# Appendix

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from imblearn.over\_sampling import SMOTE

from mlxtend.plotting import plot\_confusion\_matrix

from sklearn.linear\_model import LogisticRegression

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import precision\_score, accuracy\_score, recall\_score, roc\_curve, precision\_recall\_curve, auc

from sklearn.tree import DecisionTreeClassifier

import warnings

np.random.seed(1)

warnings.filterwarnings("ignore")

data = pd.read\_csv("../input/auto-insurance-claims-data/insurance\_claims.csv")

drop\_columns = ["\_c39", "auto\_model", "policy\_bind\_date", "policy\_state", "incident\_date",

"incident\_state", "incident\_city", "incident\_location", "policy\_csl"]

data = data.drop(drop\_columns, axis=1)

new\_response = []

response = data.iloc[:, -1]

for i in range(len(response)):

new\_response.append(1 if response[i]=='Y' else 0)

data["fraud\_reported"] = pd.Series(new\_response)

data.head(5)

plt.hist(data.age)

plt.title("Age of the car customers")

plt.xlabel("Age of the car customers")

plt.ylabel("Number of car customers")

plt.hist(data.fraud\_reported)

plt.title(" fraud reported")

plt.xlabel("response")

plt.ylabel("Number of responses")

sns.heatmap(data.corr())

plt.show()

predictors = data.iloc[:,:-1]

response = data.iloc[:, -1]

categorical\_data = predictors.select\_dtypes(exclude="number")

categorical\_predictors = categorical\_data.columns

predictors = predictors.drop(categorical\_predictors, axis=1)

one\_hot\_data = pd.get\_dummies(categorical\_data)

predictors = predictors.join(one\_hot\_data)

predictor\_columns = predictors.columns

response\_columns = response

predictors\_train, predictors\_test, response\_train, response\_test = train\_test\_split(predictors,

response,

test\_size=0.3)

sm **=** SMOTE(random\_state**=**24)

predictors, response **=** sm.fit\_resample(predictors\_train, response\_train)

​

predictors\_train **=** pd.DataFrame(predictors, columns**=**predictor\_columns)

response\_train **=** pd.Series(response)

​

model\_preds **=** {}

model **=** LogisticRegression()

model.fit(predictors\_train, response\_train)

predictions\_test **=** model.predict(predictors\_test)

predictions\_train **=** model.predict(predictors\_train)

​

conf\_matrix **=** confusion\_matrix(predictions\_test, response\_test)

plot\_confusion\_matrix(conf\_matrix)

​

precision **=** precision\_score(predictions\_test, response\_test)

recall **=** recall\_score(predictions\_test, response\_test)

​

print("Accuracy = "**+**str(accuracy\_score(predictions\_test, response\_test)))

print("Precision = "**+**str(precision))

print("Recall = "**+**str(recall))

​

tpr, fpr, threshold **=** roc\_curve(predictions\_test, response\_test, pos\_label**=**1)

model\_preds["LR"] **=** [tpr, fpr]

print()

print("AUC value = "**+**str(auc(tpr, fpr)))

knn **=** KNeighborsClassifier()

knn.fit(predictors\_train, response\_train)

​

predictions\_train **=** knn.predict(predictors\_train)

predictions\_test **=** knn.predict(predictors\_test)

​

conf\_matrix **=** confusion\_matrix(predictions\_test, response\_test)

plot\_confusion\_matrix(conf\_matrix)

​

precision **=** precision\_score(predictions\_test, response\_test)

recall **=** recall\_score(predictions\_test, response\_test)

​

print("Accuracy = "**+**str(accuracy\_score(predictions\_test, response\_test)))

print("Precision = "**+**str(precision))

print("Recall = "**+**str(recall))

​

tpr, fpr, threshold **=** roc\_curve(predictions\_test, response\_test, pos\_label**=**1)

model\_preds["KNN"] **=** [tpr, fpr]

print()

​

predictors\_train, predictors\_test, response\_train, response\_test **=** train\_test\_split(predictors,response,test\_size**=**0.3)

​

tree **=** DecisionTreeClassifier()

tree.fit(predictors\_train, response\_train)

predictions\_test **=** tree.predict(predictors\_test)

predictions\_train **=** tree.predict(predictors\_train)

​

conf\_matrix **=** confusion\_matrix(predictions\_test, response\_test)

plot\_confusion\_matrix(conf\_matrix)

​

precision **=** precision\_score(predictions\_test, response\_test)

recall **=** recall\_score(predictions\_test, response\_test)

print("Accuracy = "**+**str(accuracy\_score(predictions\_test, response\_test)))

print("Precision = "**+**str(precision))

print("Recall = "**+**str(recall))

​

tpr, fpr, threshold **=** roc\_curve(predictions\_test, response\_test, pos\_label**=**1)

model\_preds["DT"] **=** [tpr, fpr]

print()

print("AUC value = "**+**str(auc(tpr, fpr)))

**from** sklearn.ensemble **import** RandomForestClassifier

​

random\_forest **=** RandomForestClassifier()

random\_forest.fit(predictors\_train, response\_train)

predictions\_test **=** random\_forest.predict(predictors\_test)

predictions\_train **=** random\_forest.predict(predictors\_train)

​

conf\_matrix **=** confusion\_matrix(predictions\_test, response\_test)

plot\_confusion\_matrix(conf\_matrix)

​

precision **=** precision\_score(predictions\_test, response\_test)

recall **=** recall\_score(predictions\_test, response\_test)

print("Accuracy = "**+**str(accuracy\_score(predictions\_test, response\_test)))

print("Precision = "**+**str(precision))

print("Recall = "**+**str(recall))

​

tpr, fpr, threshold **=** roc\_curve(predictions\_test, response\_test, pos\_label**=**1)

model\_preds["Random Forest"] **=** [tpr, fpr]

print()

print("AUC value = "**+**str(auc(tpr, fpr)))

lda **=** LinearDiscriminantAnalysis()

lda.fit(predictors\_train, response\_train)

predictions\_test **=** lda.predict(predictors\_test)

predictions\_train **=** lda.predict(predictors\_train)

​

conf\_matrix **=** confusion\_matrix(predictions\_test, response\_test)

plot\_confusion\_matrix(conf\_matrix)

​

precision **=** precision\_score(predictions\_test, response\_test)

recall **=** recall\_score(predictions\_test, response\_test)

​

print("Accuracy = "**+**str(accuracy\_score(predictions\_test, response\_test)))

print("Precision = "**+**str(precision\_score(predictions\_test, response\_test)))

print("Recall = "**+**str(recall\_score(predictions\_test, response\_test)))

​

tpr, fpr, threshold **=** roc\_curve(predictions\_test, response\_test, pos\_label**=**1)

model\_preds["LDA"] **=** [tpr, fpr]

print()

print("AUC value = "**+**str(auc(tpr, fpr)))

plt.title("ROC curve for all models:")

plt.xlabel("FPR")

plt.ylabel("TPR")

​

**for** key, value **in** model\_preds.items():

model\_list **=** model\_preds[key]

plt.plot(model\_list[0], model\_list[1], label**=**key)

plt.legend()

plt.show()