d

by Taha Syed

Submission date: 20-May-2022 05:29PM (UTC-0400)
Submission ID: 1839997615
File name: report_insurance.docx (1.49M)
Word count: 9736
Character count: 56351

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

Payments-related fraud is a significant concern for cyber-crime organizations, and recent research has demonstrated that machine learning approaches may successfully detect fraudulent transactions in vast volumes of data. Such algorithms can detect fraudulent transactions that human auditors may not identify, and they can do so in real-time. Using publicly available simulated financial transaction data, we apply different supervised machine learning algorithms to the problem of fraud detection in this research. We want to show how supervised machine learning techniques may accurately categorize data with substantial class imbalance. In our research, we can see from the ROC curves and results that LDA is performing well compared to all the classifiers. KNN is performing the worst out of all the classifiers.

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

Lists of Figures

Lists of Abbreviations

Roc	Receiver Operating Characteristic curve
ML	Machine Learning
LR	Logistic regression
tp	True Positive
a	Accuracy
tn	True negative

NB	Navie Bayes
fn	False negative
fp	False positive
SVM	Support Vector Machines
KNN	k-Nearest Neighbors
CM	Confusion Matrix
LDA	Linear Discriminant Analysis
DL	Deep learning

Chapter 1: Introduction to the project

1.1 Introduction:

Detection of fraud affects a wide range of businesses, particularly financial sector, insurance, government organisations, and law enforcement. Fraudulent efforts have grown considerably in recent years, emphasising the importance of fraud detection. Despite the efforts of several institutions, fraud results in the loss of enormous sums of money each year. So the amount of fraudulent operations is so tiny, detecting these scams is challenging.

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

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.

1.2 Aim and Objective

The objective is to develop a model that can accurately 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.

A checklist with ratings for the different fraud indications might be created in other circumstances. The sum of these ratings and the claim's worth will decide if the case has to be investigated.

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

The project's three essential aims are as follows:

- To research the literature on financial fraud detection & better comprehend the many facets of the issue.
- Employing supervised ML procedures, tackle the concern of financial fraud detection using a publicly available selection dataset.
- To evaluate several categorization approaches to resolve which is most appropriate for this application.

The eventual aim is to develop a system and programmes 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 for them that are often irreversible. Because combating fraud cases, particularly in insurance firms, is a difficult process, the key objective is to avoid fraudulent acts at all costs. According to reports, between 22% and 34% of auto insurance claims are believed of being 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 develop a model that can predict insurance fraud. The difficulty with ML fraud detection would be that scams are significantly less prevalent than legitimate insurance payments.

Given the variety of fraud types and the low number of confirmed frauds in regular sampling, detecting financial fraud is difficult. When developing detection algorithms, the expense of false warnings must be weighed against the cost of loss avoidance. Machine learning approaches improve forecast accuracy, allowing loss controllers to cover more territory with fewer false positives.

Insurance fraud refers to various unethical behaviors that a person may engage in to obtain a favorable outcome from an insurance company. This might include arranging the event, exaggerating the circumstances, including the notable characters and the incident's cause, and lastly, exaggerating the magnitude 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 utilized as the prediction variable in machine learning models.

- We employed exploratory analysis to examine the data set and identify potential fraud predictors.
- 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.
- To compare the performance of the models, performance metrics such as the Confusion Matrix and Area Under Curve (AUC) were utilized.

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.

Chapter 2 Related works

Fraud detection is an issue that affects numerous businesses, including banking and finance, insurance, government organizations, and law enforcement. 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. Detection of these forgeries. It is challenging because the number of fraudulent actions is relatively low.

In this [1], 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 [2], 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 [3], 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 [4] , 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 [5] 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 [6], 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 [7], 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.

A large amount of data is used to calculate the likelihood of claims incidence. A claim prediction problem necessitates the use of large data models. As a result, there is a requirement for an efficient technique and a more accurate ML algorithm to estimate 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 analyse massive datasets comprising tens of thousands of customer details by the insurance provider.

Kaya is one of the largest auto and house insurance companies. According to Kaya, 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 Machine learning?

Many of the modeling approaches related with artificial intelligence are included in Machine Learning (ML). ML model is used to predict results from any digital information by finding significant features using statistics. Models may be trained on huge, complicated datasets, and they can also improve on their own, without the need for software upgrades or monitoring (Bi, 2019).

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 through utilizing information that envelops 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 info 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 make it workable for one to quickly and consequently foster

models that can rapidly and precisely dissect remarkably huge and complex informational collections.

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.

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 be illustrative of the information the AI application will ingest to tune model boundaries. Preparing information is once in a while marked, meaning it has been labelled 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 includes 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:

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 neighbour with unlabeled information. A few calculations like brain organizations can be designed to work with both bunching and forecast use cases.

3. Train the Algorithm to Build the Model

Preparing the calculation involves tuning model factors and boundaries to more precisely anticipate the proper outcomes. 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 client.

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. Where the new data will come from relies upon the idea of the issue to be tackled. For example, a ml model for self-driving vehicles will ingest genuine data on street conditions, articles and transit regulation.

3.2.3 Machine learning Methods

The learning algorithms are used in the building of various domain models. These learning algorithms, for example, are used to detect abnormalities inside a system, a network, computer vision, e-mail spam, and so on.

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. The calculation contrasts its own anticipated results and the

right results to work out model exactness and afterwards 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 such a large number of 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 will most presumably 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 information and show prepare testing information. For example, the inputs can be the climate figure, and the yields would be the visitors to the shoreline. The objective in 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 the supervision we offer presents inclination to the learning. The show can as it were copy precisely what it was appeared, so it is exceptionally vital to appear it solid, impartial illustrations. Moreover, directed learning ordinarily requires a part of information some time 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.

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.

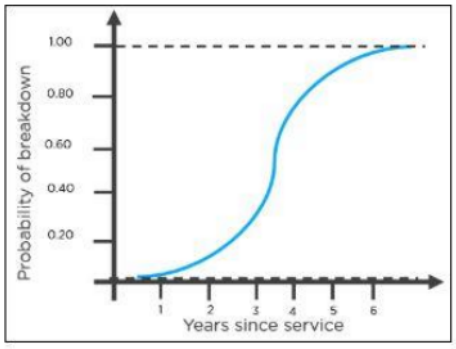


Figure 1 logistic regression curve example

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

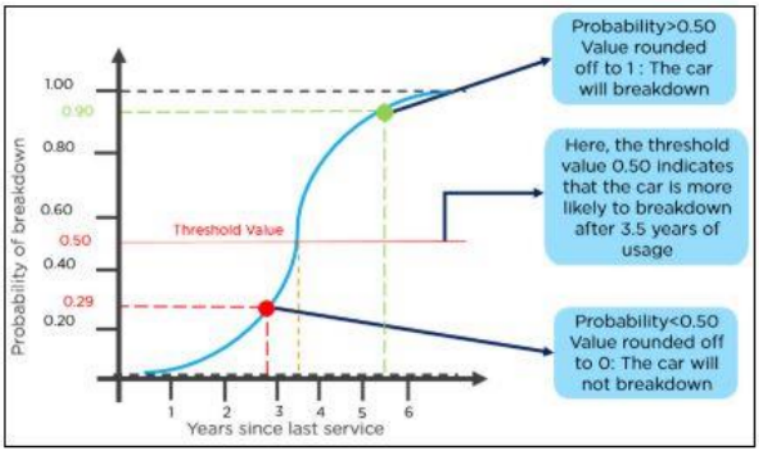


Figure 2 logistic regression threshold example

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.

Figure 3 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).

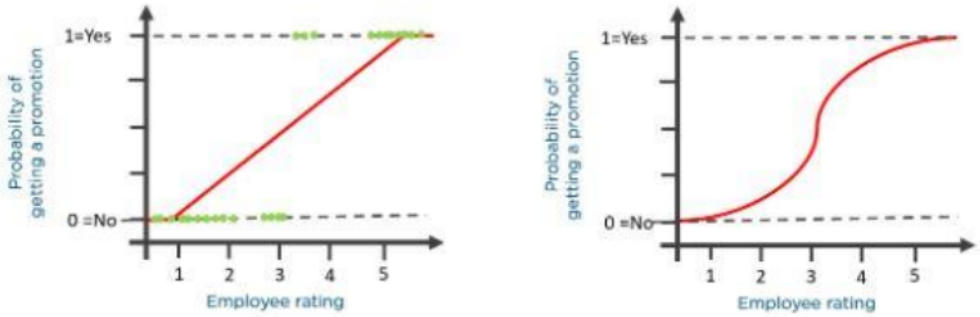


Figure 4 LR sbend

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

$\theta = p/1 - p$

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:

$y = \beta 0 + \beta 1 * x$

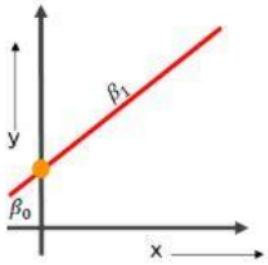


Figure 5 LR equation on graph

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

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

$$e^{ln}\left(\frac{p(x)}{1-p(x)}\right) = e^{\beta_0+\beta_1 x}$$

$$\left(\frac{p(x)}{1-p(x)}\right) = e^{\beta_0+\beta_1 x}$$

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

For sigmoid function the below equation is:

$$p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

Now sigmoid curved equation:

Figure 6 sigmoid curved equation

- Naïve bayes algorithm

It is an arrangement procedure in light of Bayes' hypothesis with suspicion of autonomy between indicators.

In basic terms, a Naive Bayes classifier expects that a specific component in a class is irrelevant to the presence of some other element. Indeed, it is truly Naïve!

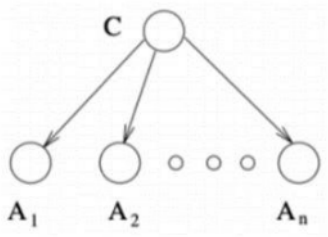


Figure 7 Naive bayes structure

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 Iris informational collection comprises the actual boundaries of three types of blossom: Versicolor, Setosa, and Virginia. The numeric boundaries that the dataset contains are Sepal width, Sepal length, Petal width, and petal length. With this information, we will foresee the classes of the blossoms given these boundaries. The information comprises ceaseless numeric qualities that portray the particular highlights' elements. Regardless of whether these elements rely upon one another or upon the presence of different highlights, a Naive Bayes classifier would think about these properties to autonomously add to the likelihood that the bloom has a place with specific animal categories.

An exploratory information examination was performed to enter the significant factors, and one such result is displayed underneath. Out of the numerous factors, this plot shows that petal length is the key differentiator, with the least cross-over.

- **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. 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 information arrangement technique for assessing the probability that an information point will turn into an individual from some gathering given what bunch the information focuses closest to it have a place with.

The k-closest neighbor calculation is a directed ML calculation used to cover characterization and relapse issues. Nonetheless, it's primarily utilized for order issues.

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.

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.

Not at all like arrangement utilizing fake brain organizations, KNN order is straightforward and easy to execute. It's ideal in circumstances where the information focuses are obvious or non-straight.

Basically, KNN plays out a democratic system to decide the class of a concealed perception. This implies that the class with the larger part vote will turn into the class of the data of interest being referred to.

On the off chance that the worth of K is equivalent to one, we'll utilize just the closest neighbor to decide the class of a data of interest. If the worth of K is equivalent to ten, we'll utilize the ten closest neighbors, etc.

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. If the worth of K is equivalent to one, we'll utilize only one closest neighbor to decide the gathering of the useful piece of information.

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. 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.

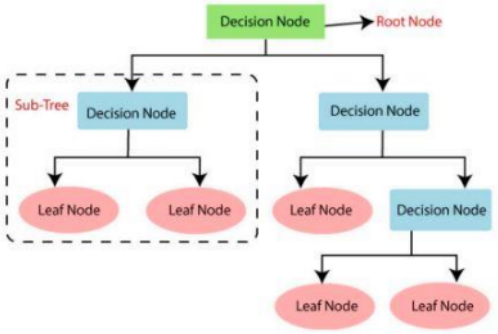


Figure 8 Decision tree framework

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.

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 9 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 an active ML calculation that can be utilized for various assignments, including relapse and order. It is a troupe strategy, implying that an arbitrary woodland model is comprised of an enormous number of little choice trees, considered assessors, which each produce their own expectations. The arbitrary timberland model joins the forecasts of the assessors to deliver a more precise expectation.

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. 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 the preparation set of N models, we over and over, example subsets of the preparation information of size n where n is not as much as N. Examining is done indiscriminately yet with substitution. This subsampling of a preparation set is called bootstrap totaling or sacking.

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. It was depicted for a 2-class issue and later summed up as 'Multi-class Linear Discriminant Analysis' or 'Numerous Discriminant Analysis' by C.R.Rao in 1948. LDA is the most regularly involved dimensionality decrease strategy in administered learning. Fundamentally, it is a preprocessing venture for design order and AI applications.

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.

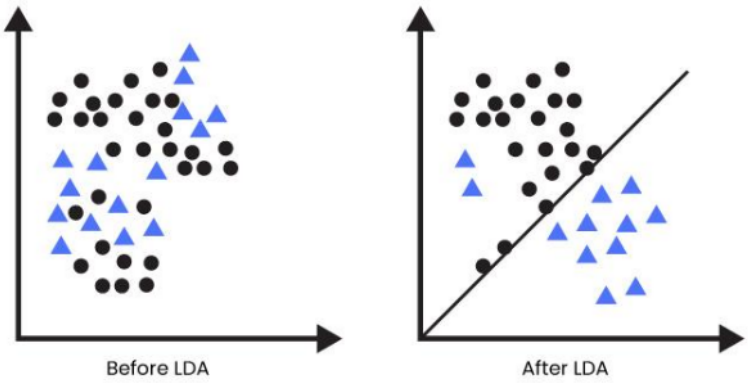


Figure 10 LDA

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. 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.

ML furnishes frameworks with the capacity to consequently gain and improve, as a matter of fact, without being expressly modified. 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 determine which strategy is the best for addressing 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:

Methodology Phases	Project Deliverables
Understanding the data set	• Report on the summary of the data set and each variable it contains along with necessary visualizations
Exploratory Data Analysis	• Report on analysis conducted and critical findings with a full description of data slices considered • Hypothesis about the separation between fraud and non-fraud transactions • Visualizations and charts that show the differences between fraud and non-fraud transactions • Python code of the analysis performed
Modeling	• Report on the results of the different techniques tried out, iterations that were experimented with, data transformations and the detailed modeling approach • Python code used to build machine learning models
Final Project Report	• Final report summarizing the work done over the course of the project, highlighting the key findings, comparing different models and identifying best model for financial fraud detection

Figure 11Methodology 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.

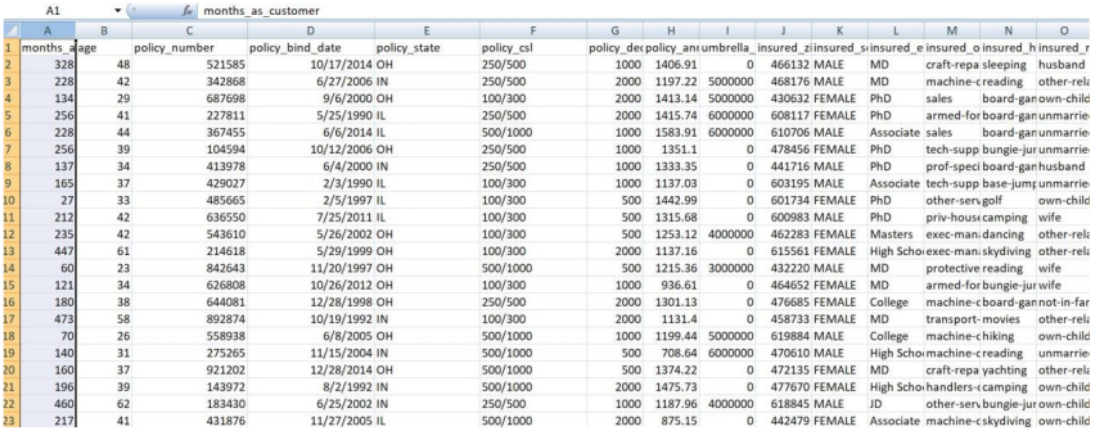
4.1 Data Set:

In this research, we are 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
- Age
- Policy number
- Policy blind date
- Policy state
- Policy csl and so on

A screenshot of the first few lines of the data set is shown in the picture below.



A1 months_as_customer

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	months_a	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_de	policy_an	umbrella_	insured_zi	insured_s	insured_e	insured_o	insured_h	insured_r
2	328	48	521585	10/17/2014	OH	250/500	1000	1406.91	0	466132	MALE	MD	craft-repa	sleeping	husband
3	228	42	342868	6/27/2006	IN	250/500	2000	1197.22	5000000	468176	MALE	MD	machine-c	reading	other-rela
4	134	29	687698	9/6/2000	OH	100/300	2000	1413.14	5000000	430632	FEMALE	PhD	sales	board-gan	own-child
5	256	41	227811	5/25/1990	IL	250/500	2000	1415.74	6000000	608117	FEMALE	PhD	armed-for	board-gan	unmarrie
6	228	44	367455	6/6/2014	IL	500/1000	1000	1583.91	6000000	610706	MALE	Associate	sales	board-gan	unmarrie
7	256	39	104594	10/12/2006	OH	250/500	1000	1351.1	0	478456	FEMALE	PhD	tech-supp	bungie-jur	unmarrie
8	137	34	413978	6/4/2000	IN	250/500	1000	1333.35	0	441716	MALE	PhD	prof-speci	board-gan	husband
9	165	37	429027	2/3/1990	IL	100/300	1000	1137.03	0	603195	MALE	Associate	tech-supp	base-jump	unmarrie
10	27	33	485665	2/5/1997	IL	100/300	500	1442.99	0	601734	FEMALE	PhD	other-serv	golf	own-child
11	212	42	636550	7/25/2011	IL	100/300	500	1315.68	0	600983	MALE	PhD	priv-house	camping	wife
12	235	42	543610	5/26/2002	OH	100/300	500	1253.12	4000000	462283	FEMALE	Masters	exec-mana	dancing	other-rela
13	447	61	214618	5/29/1999	OH	100/300	2000	1137.16	0	615561	FEMALE	High Scho	exec-mana	skydiving	other-rela
14	60	23	842643	11/20/1997	OH	500/1000	500	1215.36	3000000	432220	MALE	MD	protective	reading	wife
15	121	34	626808	10/26/2012	OH	100/300	1000	936.61	0	464652	FEMALE	MD	armed-for	bungie-jur	wife
16	180	38	644081	12/28/1998	OH	250/500	2000	1301.13	0	476685	FEMALE	College	machine-c	board-gan	not-in-far
17	473	58	892874	10/19/1992	IN	100/300	2000	1131.4	0	458733	FEMALE	MD	transport-	movies	other-rela
18	70	26	558938	6/8/2005	OH	500/1000	1000	1199.44	5000000	619884	MALE	College	machine-c	hiking	own-child
19	140	31	275265	11/15/2004	IN	500/1000	500	708.64	6000000	470610	MALE	High Scho	machine-c	reading	unmarrie
20	160	37	921202	12/28/2014	OH	500/1000	500	1374.22	0	472135	FEMALE	MD	craft-repa	yachting	other-rela
21	196	39	143972	8/2/1992	IN	500/1000	2000	1475.73	0	477670	FEMALE	High Scho	handlers-c	camping	own-child
22	460	62	183430	6/25/2002	IN	250/500	1000	1187.96	4000000	618845	MALE	JD	other-serv	bungie-jur	own-child
23	217	41	431876	11/27/2005	IL	500/1000	2000	875.15	0	442479	FEMALE	Associate	machine-c	skydiving	own-child

Figure 12 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. 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. 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.

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. 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:

$$x' = x - \text{xmin} / \text{xmax} - \text{xmin}$$

- xmax = maximum value
- xmin = minimum value

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).

Precision a, Preparing Time tt, which is the total time required to prepare a classifier, Specificity s, and Expectation Time pt, which is the total time required for a computation to predict all of the information, were all examined inside the suggested system. The tp esteem refers to the right 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 quantity of assaults correctly recognised. Here is who we're talking to.

Accuracy = a

32

True Positive = tp

False Positive = fp

True Negative = tn

False negative = fn

4.4 Implementation

4.4.1 Import library

We are using Jupyter Notebook, we can build up a local computer by downloading the TensorFlow backend directly

! Pip install tensorflow

! pip keras pip install

First i import all library in jupyter note book.

```
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
```

Figure 13 import libraries

4.4.2 Import dataset

Now, we import dataset in jupyter notebook.

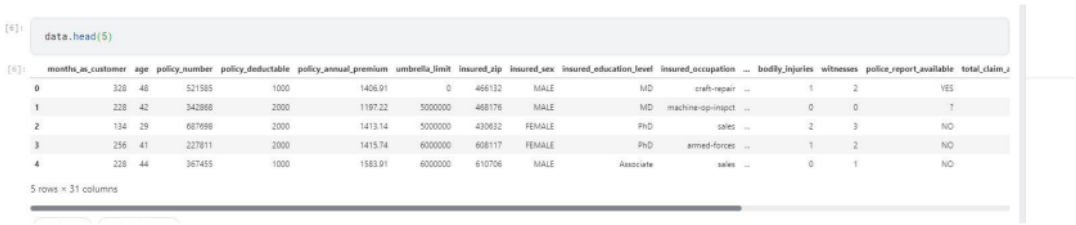
```
np.random.seed(1)

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)
```

Figure 14 import dataset

4.4.3 Dropping missing data

We will be dropping insignificant features like:

- policy_bind_date
- policy_state
- incident_date
- auto_model
- _c39
- policy_csl



```
[6]: data.head(5)
```

[6]:

	months_as_customer	age	policy_number	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	insured_sex	insured_education_level	insured_occupation	...	bodily_injuries	witnesses	police_report_available	total_claim_a
0	328	48	521585	1000	1406.91	0	466132	MALE	MD	craft-repair	...	1	2	YES	
1	228	42	342868	2000	1197.22	5000000	468176	MALE	MD	machine-op-inspct	...	0	0	?	
2	134	29	687698	2000	1413.14	5000000	430632	FEMALE	PhD	sales	...	2	3	NO	
3	256	41	227811	2000	1415.74	6000000	608117	FEMALE	PhD	armed-forces	...	1	2	NO	
4	228	44	367455	1000	1583.91	6000000	610706	MALE	Associate	sales	...	0	1	NO	

5 rows × 31 columns

Figure 15 dataset after missing data

4.4.4 Performing initial EDA:

EDA is utilized by information researchers to dissect and examine informational collections and sum up their primary qualities, frequently utilizing information representation techniques. It decides how best to control information sources to find the solutions you really want, making it simpler for information researchers to find designs, spot abnormalities, test a theory, or look at presumptions.

Initially created by American mathematician John Tukey during the 1970s, EDA procedures are a broadly involved strategy in the information revelation process today. EDA is essentially used to see what information can uncover past the proper displaying or speculation testing task and gives a superior comprehension of informational collection factors and the connections between them. Likewise, it can help decide whether the measurable procedures you are thinking about for information investigation are proper.

```
plt.hist(data.age)
plt.title("Age of the  car customers")
plt.xlabel("Age of the car  customers")
plt.ylabel("Number of  car customers")
```

```
[20]: Text(0, 0.5, 'Number of  car customers')
```

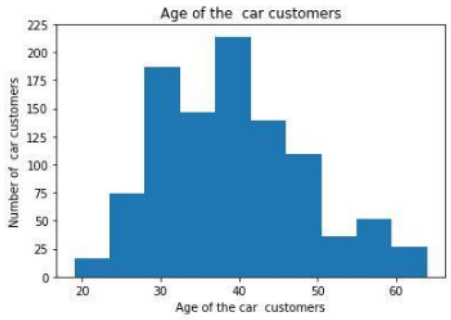


Figure 16 EDA

We now analysis the fraud

```
plt.hist(data.fraud_reported)
plt.title(" fraud reported")
plt.xlabel("response")
plt.ylabel("Number of responses")
```

```
[24]: Text(0, 0.5, 'Number of responses')
```

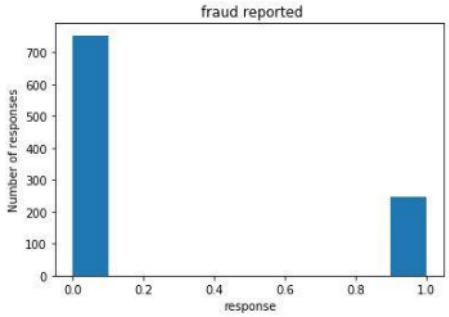


Figure 17 fraud analysis

We 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:

Synthetic Minority Oversampling Technique is one of the most usually utilized oversampling techniques to take care of the irregularity issue. It intends to adjust class dissemination by arbitrarily expanding minority class models by recreating them.

Destroyed combines new minority examples between existing minority cases. It creates the virtual preparation records by direct interjection for the minority class. These engineered preparing records are produced by arbitrarily choosing at least one of the k-closest neighbors for every model in the minority class. After the oversampling system, the information is recreated and a few characterization models can be applied for the handled information.

Heatmap

A heatmap is a two-dimensional graphical representation of data. A matrix represents each data value. To begin, make a pair plot of all independent & dependent characteristics. It will show the relationship between dependent and independent characteristics. If the relationship between the independent and dependent features is less than 0.2, then use that independent feature to develop a model.

```
sns.heatmap(data.corr())
plt.show()
```

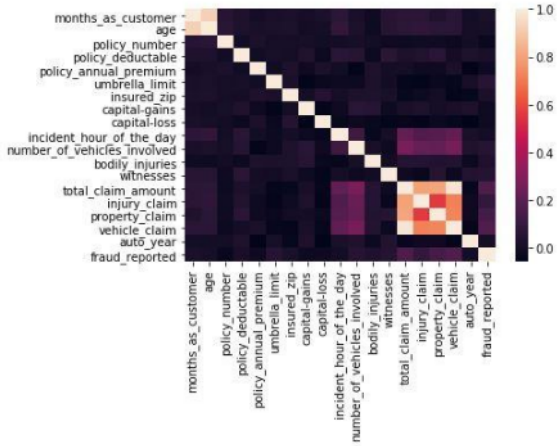


Figure 18 heatmap

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.

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

We implement categorical data encoding in our research because Since most of our data is categorical we have two options, assign a integer value to each level of the categorical variable or one-hot encode these categorical variables. One major drawback of assigning integer value to each level is that it adds additional charecteristics to the data. For example let's say we have a variable with levels as BMW, Mazda, Mercedes and Subaru and we are assigning 0, 1, 2 and 3 integer values to them respectively. When we 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 at all.

```
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)
```

Figure 19 categorical data encoding

Now, we perform one hot encoding:

```
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)
```

Figure 20 one hot encoding

Since we have class imbalance in the data, we 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.

```
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 = {}
```

Figure 21 oversampling using SMOTE

4.4.5 Model

First, i implement

1. logistic regression

```
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)))
```

```
Accuracy = 0.84375
Precision = 0.844311377245509
Recall = 0.8545454545454545

AUC value = 0.8434017595307918
```

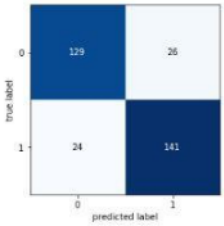


Figure 22 LR model with output

2. KNN

Now we implement KNN model

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

```
Accuracy = 0.7125
Precision = 0.8083832335329342
Recall = 0.6923076923076923

AUC value = 0.7181538461538461
```

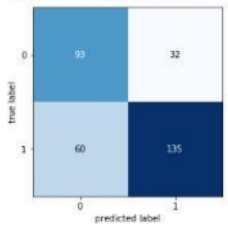


Figure 23 KNN model with output

3. DT (Decision Tree)

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

```
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)))
```

```
****************************************
Results on testing data:
****************************************
Accuracy = 0.85
Precision = 0.8862275449101796
Recall = 0.8361581920903954

AUC value = 0.8516455296116312
```

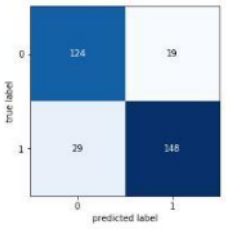


Figure 24 Decision tree implementation with output

4. Random forest classifier

```
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)))
```

```
****************************************
Results on testing data:
****************************************
Accuracy = 0.878125
Precision = 0.8562874251497006
Recall = 0.9050632911392406

AUC value = 0.8784575714955462
```

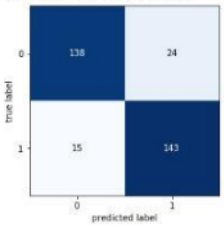


Figure 25 Random forest implementation with output

5. Linear Discriminant Analysis

```
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)))
```

```
Accuracy = 0.903125
Precision = 0.9640718562874252
Recall = 0.8655913978494624

AUC value = 0.9104076392232386
```

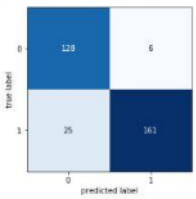


Figure 26 LDA implementation with output

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 matricescf, evaluating the Receiver Operating Characteristic curves, & summarising the key assessment measures. In this section, I provide 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 we can see from the above ROC curves and results LDA is performing well when 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 this size of the data I am not surprised with this result.

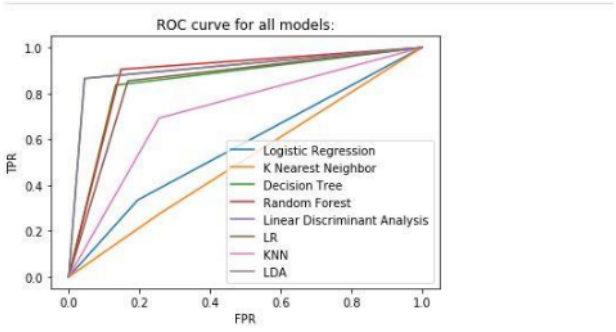


Figure 27 Result Analysis with ROC

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:

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)
```

55

```
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)
```

```
57


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)
29
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()
```

d

ORIGINALITY REPORT

19%	8%	2%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Liverpool John Moores University Student Paper	3%
2	Submitted to The University of Wolverhampton Student Paper	2%
3	scholarworks.rit.edu Internet Source	2%
4	Submitted to Westcliff University Student Paper	1%
5	Submitted to University of Bedfordshire Student Paper	1%
6	deepai.org Internet Source	1%
7	ijpsat.ijsht-journals.org Internet Source	1%
8	Submitted to Houston Community College Student Paper	1%
9	www.nehalemlabs.net	

	Internet Source	1 %
10	"Proceedings of International Conference on Trends in Computational and Cognitive Engineering", Springer Science and Business Media LLC, 2021 Publication	1 %
11	Submitted to University of Northampton Student Paper	<1 %
12	Submitted to University of Greenwich Student Paper	<1 %
13	Submitted to University of Southern Queensland Student Paper	<1 %
14	Submitted to University of North Texas Student Paper	<1 %
15	Submitted to Indian School of Business Student Paper	<1 %
16	www.coursehero.com Internet Source	<1 %
17	Submitted to UT, Dallas Student Paper	<1 %
18	blog.floydhub.com Internet Source	<1 %

19	Submitted to Asia Pacific Instutute of Information Technology Student Paper	<1 %
20	Submitted to King's College Student Paper	<1 %
21	www.bits.edu.in Internet Source	<1 %
22	Submitted to Ohio University Student Paper	<1 %
23	acikbilim.yok.gov.tr Internet Source	<1 %
24	Submitted to Botswana Accountancy College Student Paper	<1 %
25	Submitted to University of Teesside Student Paper	<1 %
26	dr.ntu.edu.sg Internet Source	<1 %
27	Submitted to Lal Bahadur Shastri Institute of Management Student Paper	<1 %
28	Nikhil Ketkar, Jojo Moolayil. "Deep Learning with Python", Springer Science and Business Media LLC, 2021 Publication	<1 %
	Submitted to University of Hertfordshire	

29	Student Paper	<1%
30	Wang, Chenli. "Cost Effective and Non-Intrusive Occupancy Detection in Residential Building Through Machine Learning Algorithm.", Santa Clara University, 2020 Publication	<1%
31	medium.com Internet Source	<1%
32	www.hindawi.com Internet Source	<1%
33	Poornachandra Sarang. "Artificial Neural Networks with TensorFlow 2", Springer Science and Business Media LLC, 2021 Publication	<1%
34	Laboni Akter, Nasrin Akhter. "Chapter 22 Detection of Ovarian Malignancy from Combination of CA125 in Blood and TVUS Using Machine Learning", Springer Science and Business Media LLC, 2021 Publication	<1%
35	copycoding.com Internet Source	<1%
36	orca.cardiff.ac.uk Internet Source	<1%
37	tudr.thapar.edu:8080	

	Internet Source	<1%
38	Martin Seehafer, Stefan Nörtemann, Jonas Offtermatt, Fabian Transchel, Axel Kiermaier, René Külheim, Wiltrud Weidner. "Actuarial Data Science", Walter de Gruyter GmbH, 2021 Publication	<1%
39	Ping Han. "An efficient enhancement technique of X-ray carry-on luggage images", 2008 9th International Conference on Signal Processing, 10/2008 Publication	<1%
40	machinelearningmastery.com Internet Source	<1%
41	link.springer.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

d

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51