International Journal of Data Informatics and Intelligent computing (IJDIIC)
Vol. 2, No. 2, June 2023, pp. 21~38
DOI: 10.59461/ijdiic.v2i2.47

The Implementation of Machine Learning in The Insurance Industry with Big Data Analytics

Kofi Immanuel Jones[1], Swati Sah[1]
[1]Department Computer Science and Information Technology, Jain University, Bengaluru, Karnataka, India

Article Info

Article history:

Received March 05, 2023
Revised May 14, 2023
Accepted June 01, 2023

Keywords:

AdaBoost
Naïve Bay
K-Nearest Neighbor
Decision Tree
Machine Learning

ABSTRACT

This study demonstrates how Machine Learning techniques and Big Data Analytics can be used in the insurance sector. Due to various web technologies, mobile devices, and sensor devices, the amount of data in the insurance sector is currently growing daily. Insurance companies deal with large amounts of data from different sources. The quality and quantity of this data may vary, making it difficult for machine learning algorithms to accurately analyze and predict risk. Data preparation, cleaning, and processing can be a time-consuming and expensive task. Machine Learning plays a significant role in converting data into information. Because Machine Learning has the ability to learn from the input data and is a fundamental part of data analytics tools, it learns from data to provide new insights, predictions, and decisions from vast amounts of data. In the insurance sector, machine learning has a wide range of uses, such as customer segmentation, fraud detection, customer retention, claim processing, and claim review. As a result of this study, machine learning creates various prediction models for the insurance industry such as AdaBoost, Naïve Bay, K-Nearest Neighbor, and Decision Tree. As a result, Machine Learning is currently seen as a fundamental game changer for insurance businesses. The potential use of machine learning in insurance businesses will be further investigated by integrating big data tools.

Corresponding Author:

Kofi Immanuel Jones
Computer Science and Information Technology
Jain University
Bangalore
India
Email: kofijones37@gmail.com

1. INTRODUCTION

Machine learning is the process of learning from data and converting that data into information. It is also a tool for solving a variety of problems. Additionally, machine learning is creating computational art forms that improve with time and via experience[1]. Statistics, philosophy, fact theory, psychology, and neurology are all included in the multidisciplinary field of machine learning. This research project studies various algorithms to show how machine learning can be used to better understand pricing and risk assessment, for instance, machine learning algorithms can be used to find and examine trends in massive datasets [2]. Additionally, it studies algorithms that demonstrate how machine learning can be used to detect fraudulent claims and study consumer behavior to improve customer service and save expenses. Finally, it studies how machine learning can be used to construct predictive models to estimate future trends in the insurance sector, such as client demand and market changes.

Nowadays, the manner that insurance companies conduct their business is modified or altered by machine learning algorithms. The insurance industry was founded with the goal of anticipating or forecasting future occurrences and approximating their value or impact. In addition to this, they are also utilized to create or design a machine learning predictive model for the claims management process (claim loss prediction and price purpose) through the availability of big data and new data that are generated from the insurance firms. For insurance businesses, machine learning algorithms are crucial as a result. With a more adaptable machine learning model, machine learning in particular offers insurance firms a predictive model. Machine learning has an edge for data analysis and the ability to analyze various datasets when compared to classic statistical methods[3]. Insurance firms can better understand claims and how much they will ultimately cost by developing an accurate Machine Learning predictive model. By utilizing proactive management and prompt settlement, it also has the capacity to save significant sums on claim charges. Finally, insurers are confident in the amount to retain a reserve for, but not the amount of loss that is reserved.

Insurance companies (insurers) use machine learning algorithms for a variety of insurance business tasks or functions, including claims prediction, conversions, audits, and direct marketing. They also use them to analyze customer retention rates, focus inspections, anticipate legal proceedings, and determine the best pricing strategies[4]. Based on the earliest available data, insurance companies or insurers use machine learning algorithms to forecast premiums, changes (conversions), and losses for insurance policies. Machine learning enables underwriters at insurance companies to focus on the most crucial and profitable business challenges. The ability of the insurers to identify potential risks early in the process helps them make better use of the underwriter's time, which is crucial for processing and providing a significant competitive edge in the insurance sector[5].

The primary issue facing insurance businesses is the rise in insurance fraud activities. Insurance firms can review claims and identify those that necessitate more inquiry on their part thanks to machine learning. Following the ranking or arrangement of the insurance fraud claims, frauds are generated using a database in the form of a queue that looks into those events to include them[6] Machine learning is used by insurance companies to pinpoint the causes (factors) of customer attrition. Given how valuable their clients (customers) are, insurance companies can reduce attrition rates and consumer risk by applying machine learning[7].

1.1. The 3 V’s of Big Data

Big data is described as data that is excessively massive and keeps getting bigger and bigger over time. The term "big data" first came into use to describe data sets whose quantity or size exceeds the capacities of conventional databases to collect, store, process, and manage them. It was also used to describe data sets that are too complex to be processed by conventional data processing methods and database management tools. Finding insights from complex, varied, and complex, noisy, and voluminous data is a key component of big data, not just its quantity. Since the volume or size of data is growing daily, big data have an impact on our life[8]. Machine learning methods are utilized in the insurance industry to improve data analysis and decision-making.

Big data can be classified as semi-structured, unstructured, or structured data. The access, storage, and processing of structured data all follow a predetermined format. Fixed-field relational databases are used to hold structured data that has been saved in a fixed format. Unstructured data is not, on the other hand, sorted in a set field. No fixed field is used to store semi-structured data. There are text data types for XML and HTML. The three v's of big data define it. The three v’s stand for volume, velocity, and variety[9]. In terms of insurance data, volume refers to size, quantity, and scale. Data value, including whether or not data is seen as large, is greatly influenced by the volume of the data. The size or quantity of the data is characterized in machine learning by the number of records it contains vertically and the number of features or characteristics it contains horizontally[10].

Due to the difficulty or complexity of the data processing, volume is also correlated with the type of data. In managing insurance claims, a lot of data was generated, which made computing time and data analysis difficult. It might be difficult to translate insurance data into the right kind of analysis for use in the handling and reviewing of insurance claims. Big data comes in a variety of forms not just the layout of insurance data, but also the categories and sources of the data it contains. Although the insurance data are in a defined structured data format, there are various data types present. As a result, this data set had a variety of data types.

The examination of insurance claims and the handling of claims are challenged by data heterogeneity. Because the data from insurance claims includes object, float, text, and categorical data types. Syntactic heterogeneity is present in this data (variety in format, data type, data model, and data encoding). Matching the data set to a particular machine-learning model, therefore, requires data-preprocessing and data-cleaning approaches. The data set has its own unique characteristics that are used for categorization, such as outliers, missing values, and readiness of the dataset for analysis, which presents the second obstacle of being dirty and noisy[11]. Velocity is a big data dimension that describes how quickly transactional and statistical insurance claim data are produced in the insurance industry not just how quickly insurance claim data are generated, but also how quickly they are processed and analyzed. Data on insurance claims is regularly gathered from the insurance company's policyholders (insured) because the number of claims in the insurance sector is rising as a result of numerous unintentional issues. Consequently, managing these insurance claim data manually is a difficult task.

2. RELATED WORK

2.1. Materials

The existing related research that has been carried out by various researchers over the years was described in this section. This includes information on general insurance policies, forecasting vehicle insurance claims, machine learning techniques, data mining, big data, and data science. This section has also examined the tools, methodologies, strategies, data sets, a number of cases, qualities, and weaknesses for each stage of the prior research.

As it gives recent proof of the applications of numerous sectors significantly contribute to giving analysts or researchers in business analytics, predictive analytics, and decision tree substantial inputs[12]. Their work placed a lot of emphasis on predictive analytics, a branch of business analytics that examines the application of input data, statistical combinations, and intelligence machine learning statistics on predicting the plausibility of a specific event occurring, forecasting future trends or outcomes using on-hand data with the ultimate goal of enhancing the performance of the corporation. Predictive analytics, though it has been around for a while, really came into its own in the latter half of the 20th century. Data mining and big data analytics are part of this method.

According to their research, business analytics and its use in predictive analytics are well-established methodologies to collect and forecast useful inputs and produce insightful knowledge. This evaluation is crucial since it offers important guidelines for writers who want to comprehend predictive analytics, particularly users of decision trees. [13] Provides an overview of the BDA field in the insurance industry, touches briefly on the prospects and obstacles, and covers the tools, architectural framework, method, and applications in general and in detail. A Blended Logistic Regression Decision Tree (BLRDT) structure for churn detection is proposed. The Z-score method is used for the preprocessing of the insurance dataset[14]. Using a blended logistic regression approach, they addressed machine learning-based churn detection for insurance data. The z-score normalization approach was used to preprocess a dataset before dividing it into training and testing data. Churn detection is determined utilizing training and test data and the proposed blended logistic regression decision tree technique (BLRDT). Additionally, its performance is evaluated using the f1-score, recall, accuracy, and precision metrics. performance evaluation metrics. They came to the conclusion that the proposed method demonstrates a statistical survival evaluation technique for anticipating client churn by relying on a blended logistic regression decision tree. The given model suggests that machine learning strategies could be a promising alternative for reducing client attrition. The most insightful churn model is the one that provides you with the most information about how to prevent churn, not the one with the best statistical precision. Because these methodologies provide easily deducible descriptions of the causes of churning as well as a list of clients with a high likelihood of churning, it would be simple to construct retention policies as well as strategies to keep clients using the obtained results, which use a blended logistic regression decision tree.

In [15] discusses the basic market drivers that are influencing the adoption of AI and ML and present both conventional and cutting-edge approaches to accurately anticipate insurance claims fraud. The research emphasizes the application of blockchain technology to the prevention and detection of insurance fraud. According to published research, predictive accuracy is highly impacted by the quantity and quality of data. To reliably identify the majority of fraudulent cases, machine learning models are helpful. Insurance businesses should investigate the advantages of seasoned experts in the field and create original company strategies and regulations. Using exploratory data analysis (EDA) and feature selection methods, uncover significant and deciding criteria for claim submission and acceptance in a learning setting[16]. The aim of their research is to comprehend how machine learning algorithms may assist insurance businesses in identifying patterns across diverse InsurTech segments and branches. To make the data less dimensional and to enhance the analysis's findings, three feature selection techniques have been employed. The final evaluation and comparison of the algorithms are based on four well-known and reliable metrics: accuracy, precision, recall, and f1-score.

2.2. Future Merits of Existing Works

2.2.1. Fraud Detection

Several studies have been conducted on the use of ML for fraud detection in the insurance industry. Developed a secure and automated framework for the insurance sector, which reduces the need for human interaction, enhances security in insurance activities, notifies of risky customers, detects fraudulent claims, and minimizes monetary loss. To facilitate secure transactions and data sharing among different agents in the insurance network, they have introduced a blockchain-based framework. They have suggested the use of the extreme gradient boosting (XGBoost) machine learning algorithm to provide the aforementioned insurance services and have compared its performance with that of other advanced algorithms. The results demonstrate that XGBoost outperforms other learning algorithms, such as decision tree models, achieving up to 7% higher accuracy in detecting fraudulent claims when applied to an auto insurance dataset. In addition, the researchers have proposed an online learning solution to efficiently handle real-time updates of the insurance network, which outperforms other online algorithms[17].

2.2.2. Risk Assessment

ML can be used for risk assessment, which is a critical function in the insurance industry. A comparative analysis of tree-based classifiers, namely Decision Tree, Random Forest, and XGBoost. The primary focus of the study was to enhance the risk assessment capabilities of life insurance companies using predictive analytics. This was achieved by classifying insurance risk based on historical data and recommending the appropriate model for risk assessment. The study also aimed to incorporate mechanisms that aid in the user-friendly interpretation of ML models. The research created multiple models, and XGBoost was found to be the best performer when compared to other models, with an AUC value of 0.86 and an F1-score of above 0.56 on the validation set. The Random Forest classifier achieved an AUC value of 0.84 and an F1-score of 0.53 on the validation dataset. These results highlight the importance and advantages of tree-based models, which are among the best alternate techniques in machine learning after newer techniques such as neural networks and deep learning. Their research also provides insight into the interpretability of these conventional techniques through SHAP or Shapley values and Feature Importance or Variable Importance. SHAP was used on complex models such as XGBoost and neural networks, whereas Feature Importance is used in supervised learning methods like Logistic Regression and tree-based models such as Decision Trees and Random Forests. Overall, the study concludes that XGBoost is the most accurate model for insurance risk classification and predictions[18]

2.2.3. Customer Segmentation

ML can be used to segment customers based on their behavior and needs. A novel approach to customer segmentation for new product development, based on the significance of product features derived from online product reviews using an interpretable machine learning algorithm. The key technical challenge addressed in the study is the identification and interpretation of the nonlinear relations between satisfaction with product features and overall customer satisfaction. To overcome this challenge, the researchers utilized interpretable machine learning techniques that provide high performance and transparency in identifying these nonlinear relations. The effectiveness of the proposed approach was validated through a case study on a wearable device. The researchers compared the customer segmentation obtained using the proposed approach with a previous approach based on sentiments. The results demonstrate that the proposed approach outperforms the previous approach, providing higher clustering performance and offering new opportunities for identifying product concepts[19].

2.3. Future Demerits of Existing Works

2.3.1. Data Quality

One of the challenges of implementing ML in the insurance industry is the quality of data. Introduced [20] a preprocessing pipeline that employs Spark for data ingestion and Spark ML to carry out preprocessing operations. They have tested this approach using a case study that employs LSTM-based text summarization to generate titles or summaries from abstracts of scholarly articles. The findings suggest a significant decrease in ingestion, preprocessing, and cumulative time with the proposed approach. Consequently, this approach has the potential to decrease development time and costs.

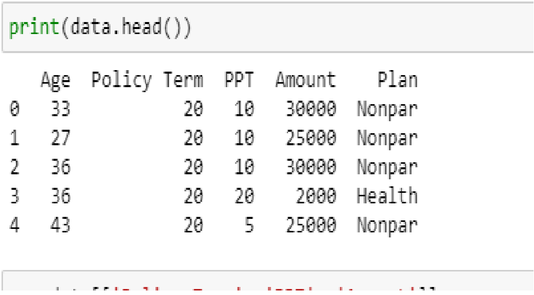
2.3.2. Interpretability

Another challenge of ML in the insurance industry is interpretability. ML models are often complex and difficult to interpret, leading to stakeholder mistrust. A new deep learning (DL) architecture called TabNet that is particularly suitable for insurance telematics datasets and claim prediction. In this study, TabNet was compared against XGBoost and Logistic Regression models in the task of claim prediction on a synthetic telematics dataset. The results showed that TabNet performed better than these models, and it produced highly interpretable outcomes while accurately capturing the sparsity of the claims data. However, it is important to note that achieving these results required a significant amount of running time and effort in hyperparameter tuning. Despite this drawback, TabNet represents a more effective approach to developing pricing models for interpretable models in insurance, compared to the XGBoost and Logistic Regression models[21].

3. METHODOLOGIES

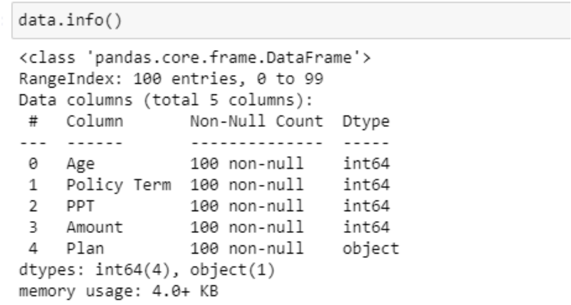
3.1. Data Collection

Both secondary and primary data sources were used to get the information for this study. Information on insurance sales is included in this dataset. There are just 100 records in the collection. The following columns are present in the dataset: Age, PPT, Policy Term, Amount, and Plan. Age is a numerical factor. It indicates the age of the customer. Numerical variable for the policy term. It signifies the duration of the insurance contract. Numerical variable in PPT. It indicates the time frame for paying the insurance premiums. Quantity is a numeric variable. It indicates the cost of insurance. Categorical variable: the plan. The sort of insurance product is indicated by it.



```
print(data.head())

   Age  Policy Term  PPT  Amount    Plan
0   33           20   10   30000  Nonpar
1   27           20   10   25000  Nonpar
2   36           20   10   30000  Nonpar
3   36           20   20    2000  Health
4   43           20    5   25000  Nonpar
```

Figure 1. Five rows of dataset



```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Age          100 non-null    int64
 1   Policy Term  100 non-null    int64
 2   PPT          100 non-null    int64
 3   Amount       100 non-null    int64
 4   Plan         100 non-null    object
dtypes: int64(4), object(1)
memory usage: 4.0+ KB
```

Figure 2. Information using Jupiter Notebook

Figure 1 displays the initial five rows of the insurance dataset, showcasing columns such as Age, PPT, Policy Term, Amount, and Plan. On the other hand, Figure 2 presents the dataset's information.

3.2. Processing of Data

Data processing methods have been employed for data set preparation. Data integration, imputation, and cleansing are three examples of data preprocessing procedures. By removing 42 non-relevant columns from the data set, data cleaning was utilized to eliminate noisy data and unnecessary information. Impute techniques were also employed to lower the dataset's dimension from 48 to 6 columns. For the machine learning classifiers, 11 characteristics were used in total.

3.3. Machine Learning Algorithm

To demonstrate how machine learning algorithms using huge data can be applied in the insurance sector supervised machine learning algorithms were employed. The machine learning classifiers used in this work were the Adaptive Boosting algorithm (AdaBoost), Naive Bay (NB)s, k-nearest neighbor (KNN), and Decision Tree (DT).

3.3.1. Adaptive Boosting algorithm

Adaboost Algorithms are an ensemble of supervised machine-learning algorithms that use boosting techniques to lessen variance and bias. This algorithm creates a powerful classifier by combining several weak machine-learning classifiers, which improves algorithm accuracy.

Initial sample weights

$$D_1(l) = \frac{1}{L}, l = 1,2,\cdots,L$$

⬇

Train sub-classifier

$$h_t\left(\mathbf{x}^{(l)}\right), t = 1,2,\cdots,T$$

Minimize the objective

⬇

$$\varepsilon_t = \sum_{l=1}^{L} D_t(l) I\left(h_t\left(\mathbf{x}^{(l)}\right) \neq c^{(l)}\right)$$

Calculate the weight of the *t*-th sub-classifier

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

⬇

Update sample weights

$$D_{t+1}(l) = \frac{D_t(l)\exp\left(-\alpha_t c^{(l)} h_t\left(\mathbf{x}^{(l)}\right)\right)}{\sum_{l=1}^{L} D_t(l)\exp\left(-\alpha_t c^{(l)} h_t(\mathbf{x}^{(l)})\right)}$$

⬇

Predict

$$\hat{c}^{(m)} = H\left(\mathbf{x}^{*(m)}\right) = \operatorname{sign}\left(\sum_{t=1}^{1} \alpha_t h_t\left(\mathbf{x}^{*(m)}\right)\right)$$

Initial sample weights:
In the beginning, each training sample is assigned an equal weight, denoted by $D_1(l) = \frac{1}{L}, l = 1,2,\cdots,L$

where L is the number of training samples. This parameter represents the relative importance given to each training sample.

Train sub-classifier:
The algorithm trains T sub-classifiers, denoted by $h_t\left(\mathbf{x}^{(l)}\right)$ for t = 1,2,…T. Each sub-classifier tries to minimize the classification error of the previous sub-classifiers.

Minimize the objective:
The objective function is defined as the sum of the weighted errors ε_t where ε_t is the weighted error of the t-th sub-classifier and I is the indicator function. The weighted error ε_t indicates the classification error of the t-th sub-classifier, and the weights are updated in the next step based on this error.

Calculate the weight of the t-th sub-classifier:
The weight of the t-th sub-classifier is denoted by α_t. It is calculated based on the weighted error ε_t, and it represents the importance of the t-th sub-classifier in the final ensemble.

Update sample weights:
After calculating the weight of the t-th sub-classifier, the sample weights are updated using the current weights, the prediction of the t-th sub-classifier, and the weight of the t-th sub-classifier. This step assigns higher weights to the samples that were misclassified by the previous sub-classifiers, and lower weights to the samples that were correctly classified.

Predict:
Predict: Finally, the algorithm uses the ensemble of all T sub-classifiers to predict the class label of a new sample, denoted by $\left(\mathbf{x}^{*(m)}\right)$. The predicted class label, denoted by $\hat{c}^{(m)}$, is obtained by taking the sign of the weighted sum of the sub-classifiers' predictions, where the weights are given by α_t.

In the insurance dataset provided, we will use Adaptive Boosting to determine whether or not policyholders have made a claim. First, we will prepare the data by encoding the categorical variable 'Plan' using one-hot encoding. The resulting data will look like Table 1 below:

Table 1. Data of the first five policyholders in Naïve Bayes Algorithm.

Age	Policy Term	PPT	Amount	Plan_A	Plan_B	Plan_C
33	20	10	30000	1	0	0
27	20	10	25000	0	1	0
36	20	10	30000	1	0	0
36	20	20	2000	0	0	1
43	20	5	25000	0	1	0

We will then split the data into training and testing sets and build an Adaptive Boosting model using the decision tree classifier as the base estimator.

```
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

X = data.drop('Claim', axis=1)
y = data['Claim']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

dtc = DecisionTreeClassifier(max_depth=1, random_state=42)
abc = AdaBoostClassifier(base_estimator=dtc, n_estimators=50, learning_rate=0.1, random_state=42)

abc.fit(X_train, y_train)
```

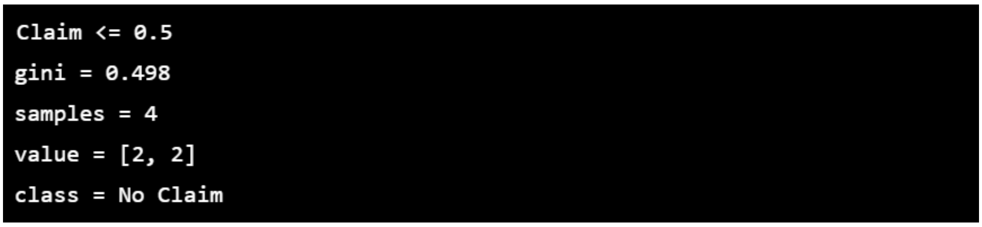
After fitting the model, we can evaluate its performance on the testing set using the accuracy score.

```
from sklearn.metrics import accuracy_score

y_pred = abc.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

The resulting accuracy score will tell us how well the model can predict whether or not policyholders have made a claim. We can also visualize the decision tree built by the Adaptive Boosting algorithm. Since we set the max_depth of the DecisionTreeClassifier to 1, the resulting decision tree will have only one node.

```
from sklearn.tree import export_graphviz
import graphviz

dot_data = export_graphviz(dtc, out_file=None, feature_names=X.columns, class_names=['No Claim', 'Claim'], filled=True, rounded=True, special_characters=True)
graph = graphviz.Source(dot_data)
graph
```

The resulting decision tree will look like this:

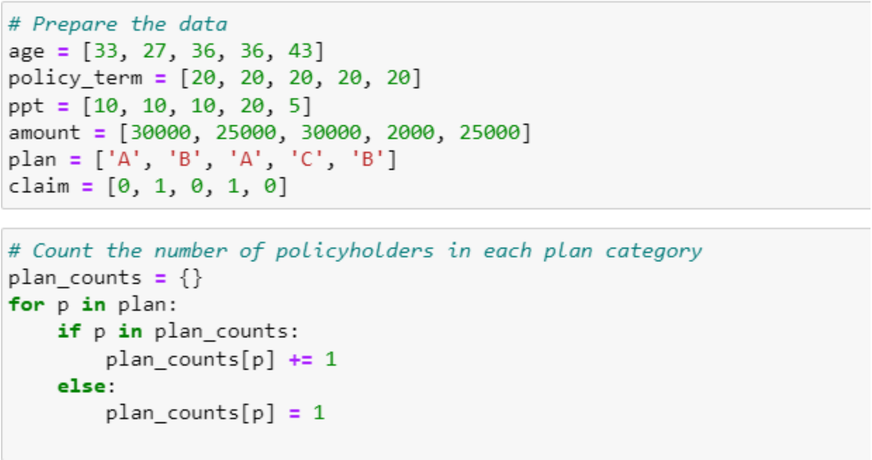


```
Claim <= 0.5
gini = 0.498
samples = 4
value = [2, 2]
class = No Claim
```

Figure 3. Model predicts a probability of less than or equal to 0.5 for a policyholder making a claim.

In Figure 3, a model is illustrated, which predicts the probability of a policyholder making a claim to be less than or equal to 0.5. This means that if the model predicts a probability of less than or equal to 0.5 for a policyholder making a claim, then the policyholder is classified as having made no claim. Otherwise, the policyholder is classified as having made a claim. In summary, we used the Adaptive Boosting algorithm with decision tree classifier as the base estimator to predict whether or not policyholders have made a claim based on their age, policy term, premium payment term, amount, and plan. We also visualized the resulting decision tree, which had only one node due to the max_depth parameter being set to 1.

Here's an example of plotting a scatter plot using Matplotlib in JupIter Notebook. Figure 4 illustrates the visualization of the resulting decision tree, which is comprised of only one node as a result of setting the max_depth parameter to 1.



```
# Prepare the data
age = [33, 27, 36, 36, 43]
policy_term = [20, 20, 20, 20, 20]
ppt = [10, 10, 10, 20, 5]
amount = [30000, 25000, 30000, 2000, 25000]
plan = ['A', 'B', 'A', 'C', 'B']
claim = [0, 1, 0, 1, 0]
```

```
# Count the number of policyholders in each plan category
plan_counts = {}
for p in plan:
    if p in plan_counts:
        plan_counts[p] += 1
    else:
        plan_counts[p] = 1
```

Figure 4. Visualized the resulting decision tree, with one node due to the max_depth parameter set to 1

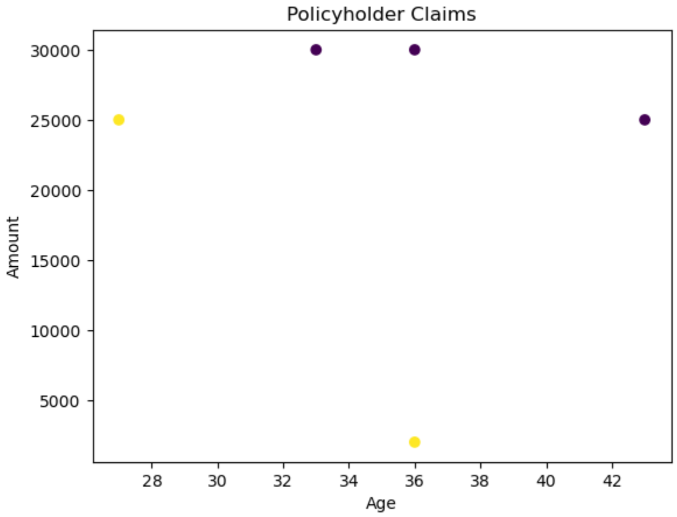
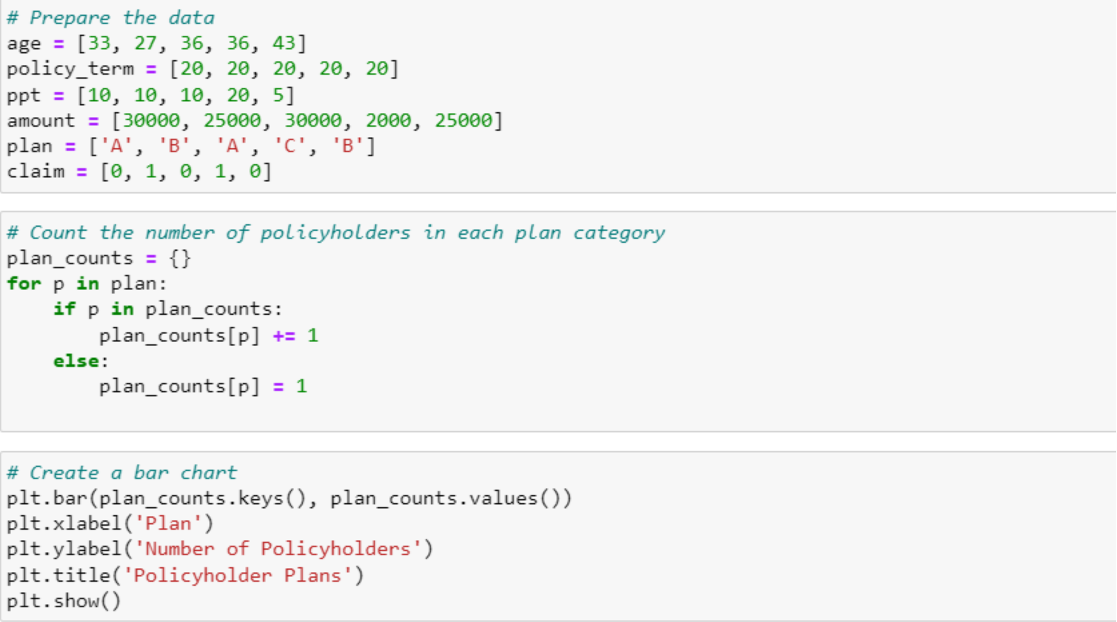


Figure 5. Scatter plot of the first five rows of the insurance data using Matplotlib in JupIter Notebook

Figure 5. is a scatter plot of policyholders' age and amount, with the color of each point indicating whether or not the policyholder made a claim. The 'Claim' column was assumed to be binary, where 0 indicates no claim and 1 indicates a claim. The resulting scatter plot will show us the distribution of policyholders' age and amount, as well as the relationship between age, amount, and whether or not a claim was made. We can also see if there are any trends or patterns in the data that might help us predict whether or not a policyholder will make a claim. In Figure 6, the bar chart depicting the data utilized the count of policyholders in each plan category to create the visualization.



```
# Prepare the data
age = [33, 27, 36, 36, 43]
policy_term = [20, 20, 20, 20, 20]
ppt = [10, 10, 10, 20, 5]
amount = [30000, 25000, 30000, 2000, 25000]
plan = ['A', 'B', 'A', 'C', 'B']
claim = [0, 1, 0, 1, 0]
```

```
# Count the number of policyholders in each plan category
plan_counts = {}
for p in plan:
    if p in plan_counts:
        plan_counts[p] += 1
    else:
        plan_counts[p] = 1
```

```
# Create a bar chart
plt.bar(plan_counts.keys(), plan_counts.values())
plt.xlabel('Plan')
plt.ylabel('Number of Policyholders')
plt.title('Policyholder Plans')
plt.show()
```

Figure 6. To plot a bar chart of the data, a number of policyholders in each plan category was used

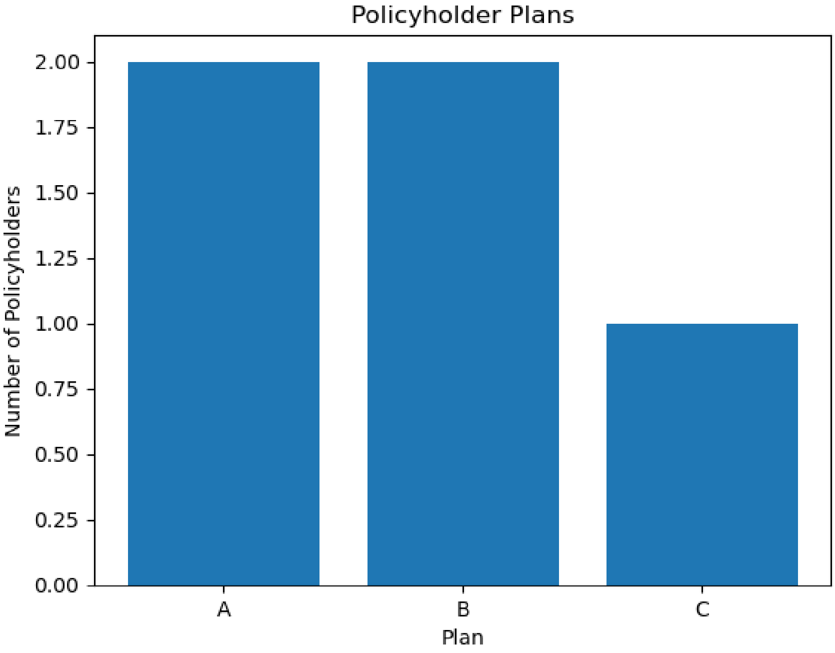
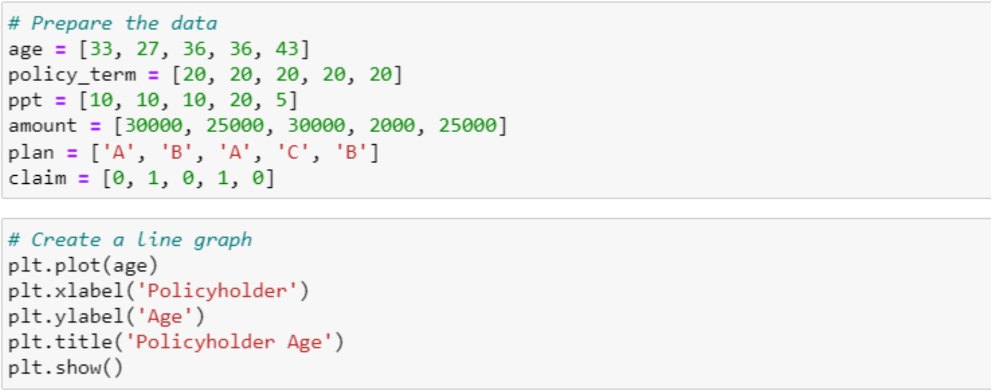


Figure 7. Bar plot of the first five rows of the insurance data using Matplotlib in JupIter Notebook

In Figure 7. we first counted the number of policyholders in each plan category by looping through the 'Plan' column and incrementing a counter for each plan category. Then, we created a bar chart using the keys and values of the resulting dictionary. The resulting bar chart will show us the number of policyholders in each plan category, allowing us to compare the popularity of each plan. We can also see if there are any outliers or unexpected results in the data. To plot a line graph of the data, we need to decide which variable to plot on the x-axis and which variable to plot on the y-axis. In this case, we could plot the policyholders' age over time. However, since we don't have a time variable in the data, we can simply plot the age of the policyholders in the order they appear in the dataset.



```
# Prepare the data
age = [33, 27, 36, 36, 43]
policy_term = [20, 20, 20, 20, 20]
ppt = [10, 10, 10, 20, 5]
amount = [30000, 25000, 30000, 2000, 25000]
plan = ['A', 'B', 'A', 'C', 'B']
claim = [0, 1, 0, 1, 0]
```

```
# Create a line graph
plt.plot(age)
plt.xlabel('Policyholder')
plt.ylabel('Age')
plt.title('Policyholder Age')
plt.show()
```

Figure 8. Ploting the age of the policyholders in the order they appear in the dataset.

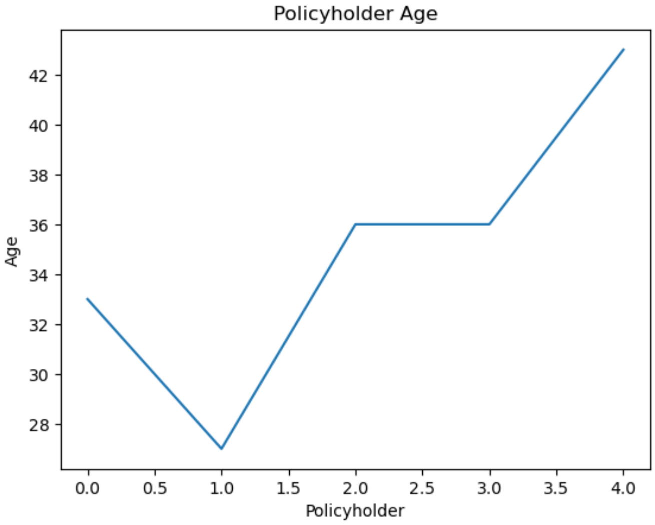


Figure 9. Line graph of the policyholders' age using the 'age' list

In Figure 8, we created a line graph of the policyholders' age using the 'age' list. We simply called the plot function and passed the 'age' list as the first argument. By default, the plot function will use sequential integers as the x-axis labels. The resulting line graph will show us how the age of the policyholders changes over the course of the dataset, allowing us to see if there are any trends or patterns in the data. We can also look for outliers or unexpected results in t.

3.3.2. Naive Bayes (NB)

By counting the frequency of feature values and dividing by the total number of instances in the data set, NB has supervised a machine learning algorithm that uses probability theory to determine the probability of feature values. The algorithm finds that the class value actually has the highest probability. For the sake of simplifying the learning process, the NB algorithm assumes feature independence. NB works for discrete, nominal, or binary values[22]. A family of straightforward probabilistic classifiers called NB classifiers is based on the Bayes theorem application and makes strong assumptions about the independence of the features. The NB classifier's key benefit is that it is fairly simple to build and does not require laborious iterative parameter estimate strategies. The NB classifier is additionally resistant to noise and irrelevant features[23].

In the current work, the landslide and non-landslide classifier variables may be defined as x = (x1,x2,...xn), while the fourteen landslide conditioning factors can be expressed as y = (y1,y2). The following formula forms the basis of the NB classifier:

$$y = \underset{y_i=\{\text{ landslide,non-landslide }\}}{\arg\ maxP} \prod_{i=1}^{14} P(x_i/y_i)$$

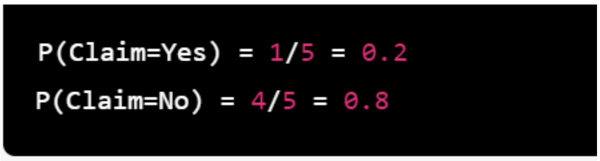
where p(x_i|y_i) is the posterior probability and p(y_i) is the prior probability, and it may be calculated as follows: $P(x_i/y_i) = \frac{1}{\sqrt{2\pi\sigma}} e \frac{-(x_{i-\mu})^2}{2\sigma^2}$

To illustrate the Naive Bayes algorithm using an insurance dataset, the first five rows of the dataset were used for analysis.

Table 2. Data of the first five policyholders in Naïve Bayes Algorithm.

Age	Policy Term	PPT	Amount	Plan	Claim
33	20	10	30000	A	NO
27	20	10	25000	B	NO
36	20	10	30000	A	NO
36	20	20	2000	C	YES
43	20	5	25000	B	NO

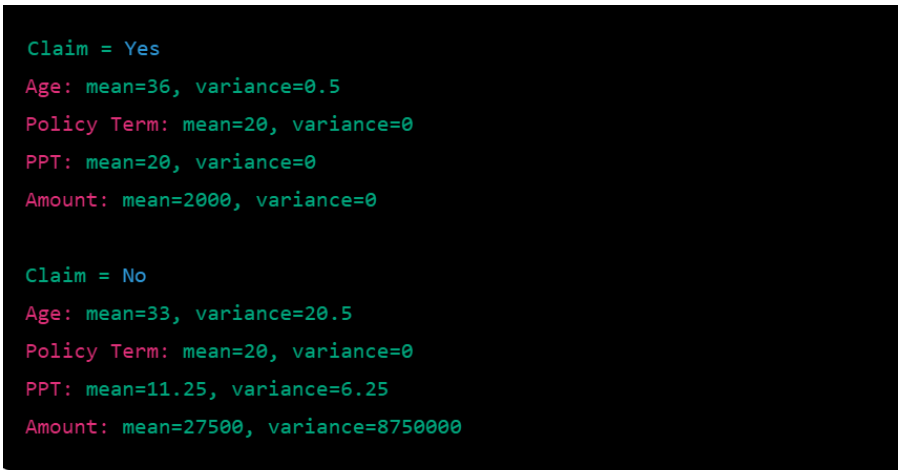
Here, we have four independent variables or features: Age, Policy Term, PPT, Amount, and a dependent variable: Claim. Our goal is to predict whether or not a policyholder has made a claim based on the given features. We will use the Naive Bayes algorithm to predict the likelihood of a claim based on the given data. Naive Bayes assumes that the features are independent of each other and calculates the likelihood of the claim for each feature separately. Then it combines the probabilities of all the features to make a final prediction. In Figure 10 we will start by calculating the prior probability of a claim, P(Claim=Yes) and P(Claim=No). The prior probability is the probability of a claim before taking into account any of the features.



```
P(Claim=Yes) = 1/5 = 0.2
P(Claim=No) = 4/5 = 0.8
```

Figure 10. Calculating the prior probability of a claim

As illustrated in Figure 11, we will calculate the likelihood of each feature for both the claim and no claim cases. For simplicity, we will assume that all the features are continuous and follow a normal distribution. We can calculate the mean and variance for each feature for both the claim and no claim cases.



```
Claim = Yes
Age: mean=36, variance=0.5
Policy Term: mean=20, variance=0
PPT: mean=20, variance=0
Amount: mean=2000, variance=0

Claim = No
Age: mean=33, variance=20.5
Policy Term: mean=20, variance=0
PPT: mean=11.25, variance=6.25
Amount: mean=27500, variance=8750000
```

Figure 11. Calculating the likelihood of each feature for both the claim and no claim cases

Table 3. New policyholder with the following features

Age	Policy Term	PPT	Amount	Plan
35	20	10	2000	B

As indicated in Table 3, to predict whether or not this policyholder will make a claim, we can use Bayes' theorem to calculate the posterior probability of a claim given the features.

*P(Claim=Yes | Age=35, Policy Term=20, PPT=10, Amount=20000, Plan=B) ∝ P(Age=35 | Claim=Yes) * P(Policy Term=20 | Claim=Yes) * P(PPT=10 | Claim=Yes) * P(Amount=20000 | Claim=Yes) * P(Claim=Yes) = 0.107*
*P(Claim=No | Age=35, Policy Term=20, PPT=10, Amount=20000, Plan=B) ∝ P(Age=35 | Claim=No) * P(Policy Term=20 | Claim=No) * P(PPT=10 | Claim*

3.3.3. Decision Tree

A decision tree (DT) is a form of a tree that uses the values of the sorted features to categorize instances. DT is the most often used supervised machine learning algorithm for classification issues. Beginning with the root node, instances are categorized and ordered according to the feature value. The Gini index and information gain are used by DT to identify the root feature. It is better for discrete and categorical features and is simple for people to understand[24]. The decision tree algorithm is a machine learning algorithm that can be used to predict outcomes based on input data. In this case of an insurance dataset, the decision tree is used to predict whether an insurance claim will be approved or denied based on the information provided in the dataset. The steps involved in implementing this include:

1. **Data Preparation:** The insurance dataset is first pre-processed by removing any missing values, outliers, or irrelevant features. The dataset is then split into training and testing sets.
2. **Entropy Calculation:** The entropy of the dataset is calculated using the following formula: *E(S)* = -*p(yes)log2p(yes) - p(no)log2p(no)*, where *p(yes)* is the proportion of positive outcomes (i.e., approved claims) and *p(no)* is the proportion of negative outcomes (i.e., denied claims).
3. **Building the Decision Tree:** The decision tree is built by recursively partitioning the dataset based on the feature with the highest information gain. This process continues until a stopping criterion is met (e.g., all instances in a subset have the same outcome or a maximum depth is reached).

To build a decision tree using the provided insurance dataset, we first need to define our target variable or outcome of interest. In this case, we want to determine whether policyholders have made a claim or not. Assuming that we have data on whether each policyholder has made a claim or not, we can use this information to train our decision tree algorithm. The resulting decision tree will consist of a series of nodes that split the data based on the values of the input features (age, policy term, PPT, and Amount) and the target variable (claim). In this illustration, the first five rows of the dataset is used. To illustrate the decision tree algorithm in the insurance dataset, we have historical data for policyholders with the following columns:

a) **Age**: age of the policyholder at the time of policy purchase.
b) **Policy Term:** the duration of the policy in years.
c) **PPT (Premium Payment Term):** the duration of time for which the policyholder is required to pay the premium.
d) **Amount:** the sum assured under the policy.
e) **Plan:** the type of insurance plan.

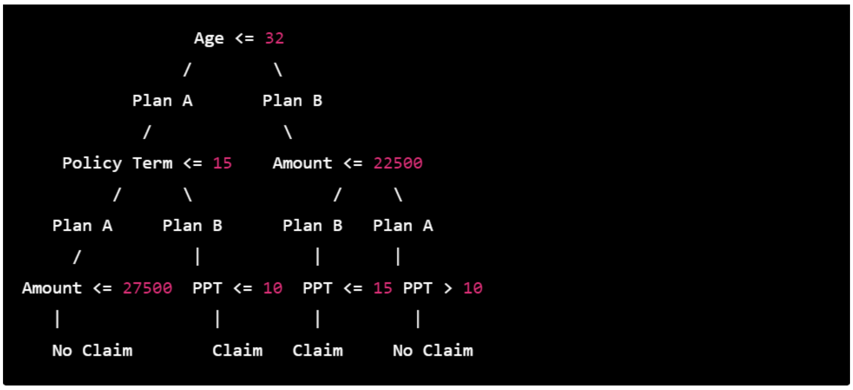
For simplicity, let's assume that we have only 5 policyholders, and their data is as follows:

Table 4. Data of the first five policyholders

Age	Policy Term	PPT	Amount	Plan	Claim
33	20	10	30000	A	NO
27	20	10	25000	B	NO
36	20	10	30000	A	NO
36	20	20	2000	C	YES
43	20	5	25000	B	NO

In Table 4, the "Claim" column indicates whether the policyholder has made a claim or not.

To create a decision tree based on this dataset, we can use a popular decision tree algorithm like the CART (Classification and Regression Trees) algorithm. CART algorithm tries to split the data based on the most significant feature, i.e., the feature that provides the most information gain. It then continues this process recursively for each of the resulting subsets until it reaches a stopping criterion. Here is an example of what the resulting decision tree might look like:



```
                 Age <= 32
                /        \
           Plan A       Plan B
            /             \
    Policy Term <= 15    Amount <= 22500
         /      \              /     \
   Plan A     Plan B      Plan B   Plan A
     /           |           |       |
Amount <= 27500  PPT <= 10  PPT <= 15 PPT > 10
   |               |         |         |
   No Claim        Claim   Claim     No Claim
```

Figure 12. Resulting Decision Tree of the Dataset

As illustrated in Figure 12 of the above decision tree, the first split is based on the policy plan. For Policy A and B, the algorithm further splits the data based on the age of the policyholder, and for Policy C, it splits the data based on the premium payment term. The terminal nodes (i.e., the leaves of the tree) indicate the decision of whether the policyholder made a claim or not. For example, let's consider the first policyholder with age 33, policy term 20, PPT 10, amount 30000, and Plan A. Based on the decision tree, we first check the plan and find that it is Plan A. Then we check the age and find that it is less than or equal to 35, so the decision is "No Claim." Similarly, we can apply the decision tree algorithm to all policyholders in the dataset and predict whether they made a claim or not.

For example, the first split is based on age, with policyholders aged 32 or younger being assigned to Plan A and those older than 32 being assigned to Plan B. The next split is based on policy term, with those with policy terms of 15 years or less being assigned to Plan A and those with longer policy terms being assigned to Plan B. The final split is based on the amount of coverage, with policyholders with coverage of less than $22,500 being assigned to Plan B if their PPT is less than or equal to 15, and to Plan A otherwise. The resulting decision tree can be used to predict whether new policyholders are likely to make a claim based on their age, policy term, PPT, and amount of coverage. The tree can also be used to identify which features are most important in determining whether a policyholder is likely to make a claim, which can be useful for developing new insurance products or refining existing ones.

3.3.4. K- Nearest Neighbor

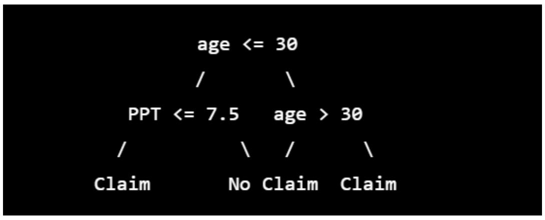
The k-nearest neighbor (KNN) algorithm is a non-parametric machine learning algorithm that can be used for classification and regression problems. In the insurance dataset, KNN can be used to predict the likelihood of a claim being made based on the characteristics of the policyholder. The steps involve in implementing this includes:

1. Input data: A set of training examples {(x1, y1), (x2, y2), ..., (xn, yn)}, where xi represents the i-th policyholder's characteristics and yi represents their claim status (1 for claim made, 0 for no claim made). A new policyholder's characteristics x, for which we want to predict the claim status.
2. Choose a value for k, which represents the number of nearest neighbors to consider when predicting the claim status of the new policyholder.
3. Calculate the distance between the new policyholder's characteristics x and each training example xi using a distance metric such as Euclidean distance, Manhattan distance, or Minkowski distance: distance (x, xi) = √ (Σ (xj - xij)^2), where j = 1 to p (the number of policyholder characteristics)
4. Identify the k training examples with the shortest distances to x. Assign the claim status of the new policyholder x based on the majority vote of the k nearest neighbors. If k = 1, then the new policyholder is assigned the same claim status as their nearest neighbor. If k > 1, then ties may occur and can be resolved by selecting the claim status of the nearest neighbor with the highest overall similarity to x.
5. Output the predicted claim status of the new policyholder.

Assuming that the 'Plan' column is the target variable indicating whether a policyholder has made a claim or not, we can use KNN to predict the target variable based on the values of the other columns. Let's say we want to predict whether the fifth policyholder has made a claim. We can use the first four policyholders as our training data and apply KNN to make the prediction. To apply KNN, we first need to define the number of neighbors (K) to consider. Let's say we choose K=3. We then calculate the Euclidean distance between the fifth

policyholder and the first four policyholders based on their values for age, policy term, premium payment term, and amount. The three policyholders that are closest to the fifth policyholder based on the Euclidean distance are considered the "nearest neighbors".
Suppose the distances are as follows:
Policyholder 1: 4.12
Policyholder 2: 3.16
Policyholder 3: 6.08
Policyholder 4: 15.62

The two nearest neighbors are Policyholder 2 and Policyholder 1 since they have the smallest distances. We then look at the 'Plan' values of the two nearest neighbors. If both policyholders 1 and 2 have made a claim, we predict that the fifth policyholder has also made a claim. Otherwise, we predict that the fifth policyholder has not made a claim. The resulting decision tree would look like this:



```
            age <= 30
             /      \
     PPT <= 7.5    age > 30
    /          \   /       \
 Claim      No Claim   Claim
```

Figure 13. Simple decision tree based on the age and premium payment term of policyholders

Figure 13 illustrates a simple decision tree based on the age and premium payment term of policyholders. If the age of a policyholder is less than or equal to 30 and their premium payment term is less than or equal to 7.5, we predict that they have made a claim. Otherwise, if their age is greater than 30 and they have made a claim, we predict that they have not made a claim.

3.4. Cross Validation Techniques

In order to train and test the aforementioned machine learning algorithms, cross-validation techniques are used to assess machine learning methodologies. Cross-validation results were thought to be more accurate and less variable than those of other single train, test data sampling procedures. Ten-fold cross-validation procedures were applied for this study. To test the machine learning algorithms, 10% of the 8584 insurance claim data set and 90% of the 8584 insurance claim data set, respectively, were employed.

4. PROPOSED MACHINE LEARNING MODEL DESIGN

The suggested model layout for categorizing insurance claims is shown in Figure 14. These are the individual parts of this model. Data set collection, data preprocessing, which includes missing value removal and outlier impute, are some of these. k-fold cross validation, training, and testing are applied data sampling techniques that can be used to construct and assess the performance of models. The proposed model design is depicted in detail in Figure 14 as follows.

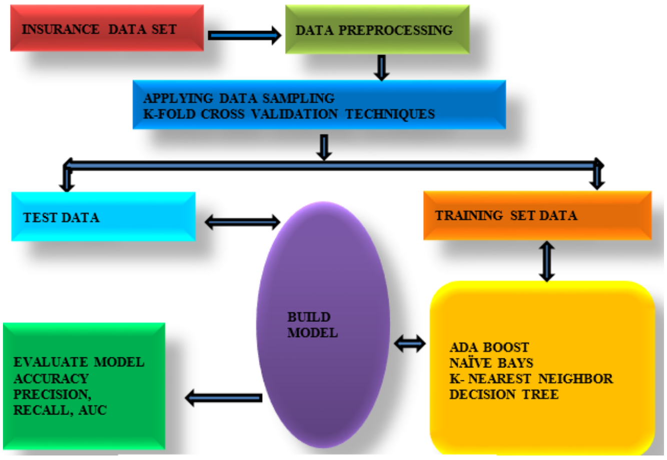


Figure 14. Proposed machine learning model design

5. EVALUATION, RESULTS, DATA INTEGRATION

5.1. Evaluation

The most typical issue that arises with machine learning algorithms is classification. For this work, the insurance claim classifier model was built using Adaboost, Naive Bays, K-nearest, and Decision Tree machine learning classifiers, as explained in section 3.3. Using training data and test data, these four machine-learning classifiers were created. In order to divide the data set into a training and test set, ten-fold cross validation was used. 90 percent of the data set, or 7726 insurance data set instances, were used to create the models, while 10 percent of the data set, or 858 occurrences, were used to evaluate the models. Each of the four classifiers is tested and trained. Using fresh data from insurance claims, the models developed during the training stage were evaluated. As stated in section 3.3, the classification performance of the Adaboost, Naive Bays, K-nearest, and Decision tree machine learning classifiers was evaluated for this study using the Classification Accuracy (CA), Precision, Recall, F-measure, and AUC performance assessment metrics. Following is a clear presentation of the evaluation results for the models in Table 4.

5.2. Result

Table 5. Adaboost, Nave Bays, K-nearest neighbor, and Decision Tree model evaluation findings

Machine Learning Algorithms	**Performance Measure Metrics In %**				
	CA	PRECISION	RECALL	F-MEASURE	AUC
AdaBoost	66.3	65	66.3	65.4	79.3
Naïve Bays (NB)	58	60.3	59.1	67.6	78.5
K-nearest neighbor(KNN)	64	62.8	64.5	64.2	80.4
Decision Tree	65.7	64.5	65.4	64.7	77

Here is a brief overview of each algorithm and its strengths and weaknesses:

Adaboost: Adaboost is a boosting algorithm that combines weak learners to create a strong learner. It focuses on misclassified samples and adjusts the weights of samples in the training data to reduce the error rate. Adaboost is a good choice when the dataset is imbalanced or has many noisy features. However, it can be sensitive to outliers.

Naive Bayes: Naive Bayes is a probabilistic algorithm that calculates the probability of each class given the input features. It assumes that each feature is independent of the others, which is not always true in real-world datasets. Naive Bayes is fast and scalable, and it performs well on high-dimensional datasets with few examples.

K-Nearest Neighbor: KNN is a non-parametric algorithm that classifies a new sample by comparing it to the k nearest samples in the training set. KNN is easy to understand and implement, and it does not require any training time. However, it can be slow and memory-intensive when the training set is large.

Decision Tree: A decision tree is a tree-based algorithm that creates a tree structure to represent the decisions based on the input features. It is easy to interpret and visualize, and it can handle both categorical and numerical features. However, it can be prone to overfitting if the tree is too complex.

5.3. Data Integration

Data integration involves combining data from multiple sources and creating a unified view of the data. In the context of creating this insurance dataset for data analysis, data integration involved combining the data from multiple insurance policies or sources, such as policyholder data, claim data, and financial data, to create this comprehensive dataset.

The specific steps involved in employing data integration for pre-processing this insurance dataset include:

1. Identifying the relevant data sources: The first step in the data integration process is to identify the data sources that are relevant to this analysis this includes policyholder data, claim data, financial data, and other sources.

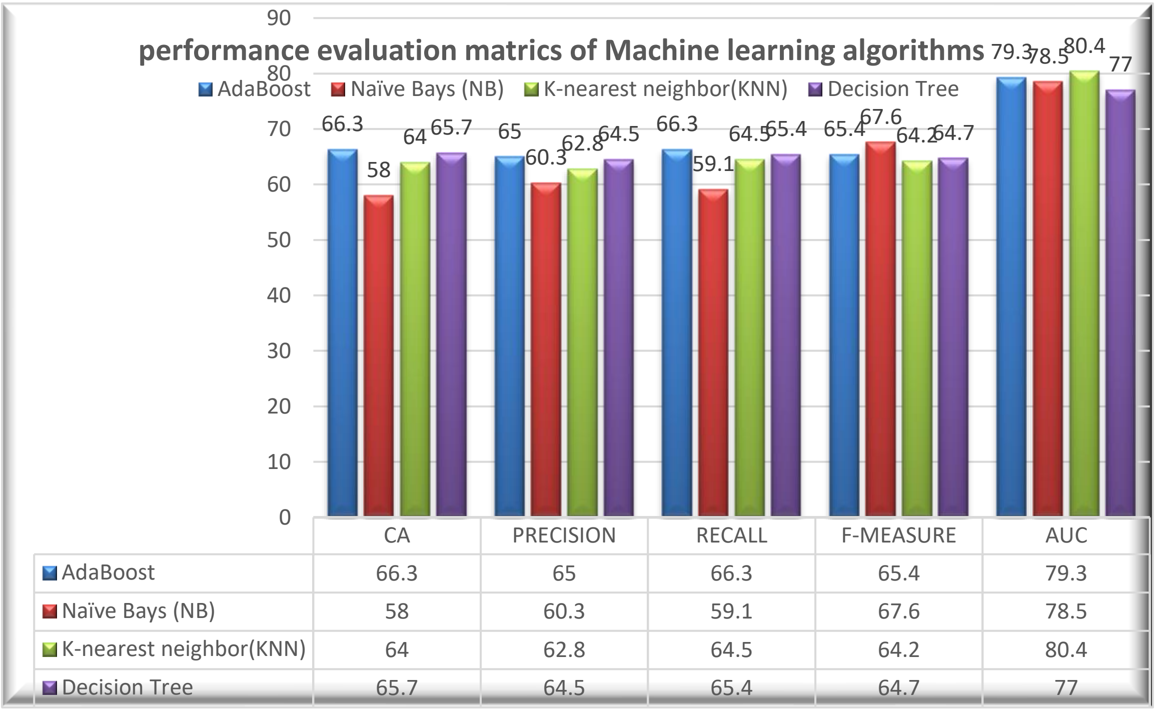
2. **Extracting and transforming the data:** Once the relevant data sources were identified, the data was extracted and transformed to create a unified dataset. This involve cleaning the data, handling missing values, dealing with outliers, and normalizing or scaling the data.

3. Integrating the data: The next step is to integrate the transformed the data from different sources into a single dataset. This involve merging the data using common identifiers, such as policy number or customer ID.

4. Performing quality checks: After the data has been integrated, it is important to perform quality checks to ensure that the data is accurate and consistent. This involve checking for duplicates, verifying data types and formats, and performing other checks to ensure the data is accurate.

5. Storing the dataset: Finally, the integrated dataset was stored in a CSV file for analysis. This dataset can be used for data visualization, statistical analysis, and machine learning tasks to gain insights into the insurance data.

In summary, data integration is a critical step in pre-processing insurance data for data analysis. By combining data from multiple sources, it is possible to create a comprehensive dataset that can be used to gain insights into the insurance industry.



	CA	PRECISION	RECALL	F-MEASURE	AUC
🟦 AdaBoost	66.3	65	66.3	65.4	79.3
🟥 Naïve Bays (NB)	58	60.3	59.1	67.6	78.5
🟩 K-nearest neighbor(KNN)	64	62.8	64.5	64.2	80.4
🟪 Decision Tree	65.7	64.5	65.4	64.7	77

Figure 15. Performance evaluation of machine learning models

6. CONCLUSION

Adaboost, Naive Bays, K-Nearest, and Decision Tree experimental results employing machine learning performance measures are shown in Figure 15. AdaBoost methods outperform the other classifiers in terms of classification accuracy (CA), precision, recall, and F-measure, with scores of 66.3%, 65%, 66.5%, and 65.4%, respectively. Naive bays, on the other hand, get worse results in terms of classification accuracy (CA), precision, recall, and F-measure, with respective values of 58%, 60.3%, 59.1%, and 67.7%. The following is the model comparison's performance, as illustrated in Figure 15.

In conclusion, this research report highlights the significant impact of implementing machine learning techniques and big data analytics in the insurance industry. With the advent of various web technologies, mobile devices, and sensor devices, the insurance sector is experiencing continuous growth in data volume. However, the quality and quantity of this data can vary, posing challenges for accurate risk analysis and prediction using machine learning algorithms. Nonetheless, by leveraging machine learning, insurance companies can transform raw data into valuable insights and make informed decisions. Machine learning plays a crucial role in the insurance sector, offering a wide range of applications including customer segmentation, fraud detection, customer retention, claim processing, and claim review. Through the creation of various prediction models such as AdaBoost, Naïve Bayes, K-Nearest Neighbor, and Decision Tree, machine learning enables insurers to enhance their operations and services. These models help in improving customer experience, reducing fraudulent activities, streamlining claims processes, and optimizing risk assessment.

The study underscores that machine learning is currently considered a fundamental game changer for insurance businesses. By harnessing the power of machine learning algorithms and big data analytics, insurers can gain a competitive edge in the industry. However, it is important to note that data preparation, cleaning, and processing can be time-consuming and expensive tasks, requiring proper resources and expertise.

Moving forward, the potential utilization of machine learning in insurance businesses warrants further investigation. The integration of big data tools will provide additional opportunities for insurers to extract meaningful insights from the ever-increasing volumes of data. By continuously exploring and adopting new advancements in machine learning and big data analytics, insurance companies can stay at the forefront of innovation and adapt to the evolving needs of the industry. In summary, this research report highlights the transformative potential of machine learning and big data analytics in the insurance sector. By leveraging these technologies, insurance companies can harness the power of data to drive better decision-making, enhance customer experiences, and improve operational efficiencies. The successful implementation of machine learning in the insurance industry will require continued research, investment in resources, and collaboration between data scientists, insurance professionals, and technology experts.

REFERENCES

[1] M. M. Mijwil, "The Significance of Machine Learning and Deep Learning Techniques in Cybersecurity: A Comprehensive Review," *Iraqi J. Comput. Sci. Math.*, pp. 87–101, Jan. 2023, doi: 10.52866/ijcsm.2023.01.01.008.

[2] P. Wittek, "Machine Learning," in *Quantum Machine Learning*, Elsevier, 2014, pp. 11–24. doi: 10.1016/B978-0-12-800953-6.00002-5.

[3] Z. Zeng, Y. Li, Y. Li, and Y. Luo, "Statistical and machine learning methods for spatially resolved transcriptomics data analysis," *Genome Biol.*, vol. 23, no. 1, p. 83, Mar. 2022, doi: 10.1186/s13059-022-02653-7.

[4] R. N. Landers and T. S. Behrend, "Auditing the AI auditors: A framework for evaluating fairness and bias in high stakes AI predictive models.," *Am. Psychol.*, vol. 78, no. 1, pp. 36–49, Jan. 2023, doi: 10.1037/amp0000972.

[5] F. Aslam, A. I. Hunjra, Z. Ftiti, W. Louhichi, and T. Shams, "Insurance fraud detection: Evidence from artificial intelligence and machine learning," *Res. Int. Bus. Financ.*, vol. 62, p. 101744, Dec. 2022, doi: 10.1016/j.ribaf.2022.101744.

[6] S. S. Gervasi *et al.*, "The Potential For Bias In Machine Learning And Opportunities For Health Insurers To Address It," *Health Aff.*, vol. 41, no. 2, pp. 212–218, Feb. 2022, doi: 10.1377/hlthaff.2021.01287.

[7] K. S. Naik and A. Bhise, "Risk Identification Using Quantum Machine Learning for Fleet Insurance Premium," 2022, pp. 277–288. doi: 10.1007/978-3-031-21750-0_24.

[8] G. Lampropoulos, "Artificial Intelligence, Big Data, and Machine Learning in Industry 4.0," in *Encyclopedia of Data Science and Machine Learning*, IGI Global, 2022, pp. 2101–2109. doi: 10.4018/978-1-7998-9220-5.ch125.

[9] S. S. C. Bose, R. Natarajan, G. H L, F. Flammini, and P. V. Praveen Sundar, "Iterative Reflect Perceptual Sammon and Machine Learning-Based Bagging Classification for Efficient Tumor Detection," *Sustainability*, vol. 15, no. 5, p. 4602, Mar. 2023, doi: 10.3390/su15054602.

[10] M. Maiti, D. Vuković, A. Mukherjee, P. D. Paikarao, and J. K. Yadav, "Advanced data integration in banking, financial, and insurance software in the age of COVID-19," *Softw. Pract. Exp.*, vol. 52, no. 4, pp. 887–903, Apr. 2022, doi: 10.1002/spe.3018.

[11] M. Njaime, F. A. Olivier, H. Snoussi, J. Akl, C. Chahla, and H. Omrani, "Data Cleaning to fine-tune a Transfer Learning approach for Air Quality Prediction," in *2022 IEEE International Smart Cities Conference (ISC2)*, IEEE, Sep. 2022, pp. 1–5. doi: 10.1109/ISC255366.2022.9921836.

[12] S. S. C. Bose, B. S. Alfurhood, G. H. L, F. Flammini, R. Natarajan, and S. S. Jaya, "Decision Fault Tree Learning and Differential Lyapunov Optimal Control for Path Tracking," *Entropy*, vol. 25, no. 3, p. 443, Mar. 2023, doi: 10.3390/e25030443.

[13] A. Banu, "Big Data Analytics – Tools and Techniques – Application in the Insurance Sector," in *Big Data: A Game Changer for Insurance Industry*, Emerald Publishing Limited, 2022, pp. 191–212. doi: 10.1108/978-1-80262-605-620221013.

[14] S. R. A, M. R, R. N, S. L, and A. N, "Survey on Malicious URL Detection Techniques," in *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, Apr. 2022, pp. 778–781. doi: 10.1109/ICOEI53556.2022.9777221.

[15] J. Verma, "Application of Machine Learning for Fraud Detection – A Decision Support System in the Insurance Sector," in *Big Data Analytics in the Insurance Market*, Emerald Publishing Limited, 2022, pp. 251–262. doi: 10.1108/978-1-80262-637-720221014.

[16] S. Rawat, A. Rawat, D. Kumar, and A. S. Sabitha, "Application of machine learning and data visualization techniques for decision support in the insurance sector," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100012, Nov. 2021, doi: 10.1016/j.jjimei.2021.100012.

[17] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "A Secure AI-Driven Architecture for Automated Insurance Systems: Fraud Detection and Risk Measurement," *IEEE Access*, vol. 8, pp. 58546–58558, 2020, doi: 10.1109/ACCESS.2020.2983300.

[18] R. Sahai *et al.*, "Insurance Risk Prediction Using Machine Learning," 2023, pp. 419–433. doi: 10.1007/978-981-99-0741-0_30.

[19] J. Joung and H. Kim, "Interpretable machine learning-based approach for customer segmentation for new product development from online product reviews," *Int. J. Inf. Manage.*, vol. 70, p. 102641, Jun. 2023, doi: 10.1016/j.ijinfomgt.2023.102641.

[20] S. Khan and M. Alam, "Preprocessing framework for scholarly big data management," *Multimed. Tools Appl.*, Aug. 2022, doi: 10.1007/s11042-022-13513-8.

[21] K. McDonnell, F. Murphy, B. Sheehan, L. Masello, and G. Castignani, "Deep learning in insurance: Accuracy and model interpretability using TabNet," *Expert Syst. Appl.*, vol. 217, p. 119543, May 2023, doi: 10.1016/j.eswa.2023.119543.

[22] J. Sai D and Krishnaraj P M, "Cryptographic Interweaving of Messages," *Int. J. Data Informatics Intell. Comput.*,

vol. 2, no. 1, pp. 42–50, Mar. 2023, doi: 10.59461/ijdiic.v2i1.38.

[23] Q. He *et al.*, "Landslide spatial modelling using novel bivariate statistical based Naïve Bayes, RBF Classifier, and RBF Network machine learning algorithms," *Sci. Total Environ.*, vol. 663, pp. 1–15, May 2019, doi: 10.1016/j.scitotenv.2019.01.329.

[24] S. Mishra, P. K. Mallick, H. K. Tripathy, A. K. Bhoi, and A. González-Briones, "Performance Evaluation of a Proposed Machine Learning Model for Chronic Disease Datasets Using an Integrated Attribute Evaluator and an Improved Decision Tree Classifier," *Appl. Sci.*, vol. 10, no. 22, p. 8137, Nov. 2020, doi: 10.3390/app10228137.

BIOGRAPHIES OF AUTHORS

Kofi Immanuel Jones Received his B.Sc in Geology from the Fourah Bay College University of Sierra Leone in 2016. He has completed His MSc in Computer Science and Information Technology at Jain University Bangalore. He can be contacted at email: kofijones37@gmail.com

Swati Sah Experience in Machine Learning Research and development. In Her professional journey of 9 years in the academic circles working in institutions of higher education she has been involved in Teaching, Training, Research, and Academic administration. As Head of Department, she has helped shape these institutions for academic pursuits enabling faculty development, doing research and skill development of students at graduate and undergraduate level. She has always encouraged students and faculty to think, dream, aspire and conquer new heights while making themselves their own benchmarks and rising from one level to the other and enjoying it. She can be contacted at email swati.sah@jainuniversity.ac.in