**Health Insurance Premium of Customers**

(Predict whether the customer would pay the next premium or not)

*A report submitted in partial fulfilment of the requirements for the Award of Degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**Tamminedi Ganesh Prasad**

**Regd. No.: 222010311030**

**Under Supervision of Mr. Raju**

**Contor Solutions Inc. Pvt Ltd, Hyderabad**

**(Duration: 27th April, 2023 to 30th June, 2023)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Gandhi Institute of Technology and Management Hyderabad**

(An Autonomous Institution)

Approved by AICTE

SANGAREDDY, TELANGANA

**Abstract**

This project focused on predicting whether customers would pay their next health insurance premium using a dataset containing columns such as 'id', 'perc\_premium\_paid\_by\_cash\_credit', 'age\_in\_days', 'Income', 'Count\_3-6\_months\_late', 'Count\_6-12\_months\_late', 'Count\_more\_than\_12\_months\_late', 'application\_underwriting\_score', 'no\_of\_premiums\_paid', 'sourcing\_channel', 'residence\_area\_type', 'premium', and 'target'. The dataset was pre-processed to handle missing values using KNN imputation and applied label encoding to convert categorical variables into numerical representations. Train-test splitting was performed to create separate training and test datasets for model training and evaluation. The features were normalized using the min-max scaler to ensure consistent scales across variables. Various classification models, including Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Extra Trees Classifier, K-Nearest Neighbours Classifier, Gaussian Naive Bayes, and Support Vector Classifier, were implemented. The models were evaluated using metrics such as accuracy, precision, recall, and F1-score to identify the best-performing model. The project encompassed comprehensive data preprocessing techniques and model evaluation methods to determine the most effective approach for predicting premium payment behaviour in the given health insurance dataset.

*Keywords: Health insurance premium,Predict,Customer,Next premium,Machine learning,Algorithms,Dataset,Sklearn,Null values,KNN imputer,Label encoding,Train-test split,Min-max scaler,Logistic Regression,Decision Tree Classifier,Random Forest Classifier,Extra Trees Classifier,K-Nearest Neighbors Classifier,Gaussian Naive Bayes,Support Vector Classifier,Best model,Preprocessing,Evaluation,Performance,Target variable*

**Introduction**

**What are the different types of Machine Learning?**

* **Supervised Learning**: Here, the algorithm learns from labeled data where input data is paired with the correct output. It's used for:
  + **Classification**: Predicting categories (e.g., spam emails).
  + **Regression**: Predicting continuous values (e.g., house prices).
* **Unsupervised Learning**: The algorithm learns from unlabeled data, finding patterns without guidance. It's used for:
  + **Clustering**: Grouping similar data points (e.g., customer segments).
  + **Dimensionality Reduction**: Simplifying data while keeping vital info.
  + **Anomaly Detection**: Spotting unusual data points (e.g., fraud detection).
* **Reinforcement Learning**: An agent learns to make decisions in an environment to maximize rewards. It's used in games, robotics, and optimization.

These types can be combined or specialized, and there are variations within each based on specific techniques.

**Benefits of Using Machine Learning in Health Insurance Premium Prediction**

* **More Accurate Predictions**: Machine learning can analyze data patterns better, leading to more accurate premium predictions. This helps insurance companies make informed decisions and reduce financial risks.
* **Better Risk Assessment**: Insurers can assess individual policyholders' risk profiles more effectively with machine learning. This helps in adjusting prices and coverage fairly, benefiting both insurers and policyholders.
* **Efficient Resource Allocation**: Predicting premium payments helps insurers use resources wisely. They can focus on high-risk individuals, reducing costs and making collection processes more efficient.
* **Financial Planning**: Accurate predictions help insurers plan their finances and maintain stability. They can make informed decisions on investments, expansion, and risk management.
* **Personalized Support**: Machine learning identifies policyholders needing extra help with premium payment. This leads to tailored support like reminders or flexible payment options, improving customer satisfaction and retention.

**About the Industry**

The health insurance industry is crucial for covering medical costs. It involves insurance companies, policyholders, healthcare providers, and regulators. Health insurance aims to safeguard individuals and families from expensive medical bills.

With health insurance, policyholders pay premiums regularly to the insurance company in return for coverage. This coverage includes things like hospital stays, doctor visits, medicines, tests, and preventive care. Health insurance plans can differ in terms of what they cover and may have varying deductibles, co-pays, and maximum expenses policyholders must bear.

**AI / ML Role in Health Insurance**

Artificial Intelligence (AI) and Machine Learning (ML) are making a big impact on health insurance:

* **Predicting Behavior**: AI/ML can look at lots of past data to predict how customers will behave. This helps insurers set prices, design policies, and understand customers better.
* **Stopping Fraud:** AI/ML can spot signs of fraud in insurance claims. It checks big datasets for unusual things and marks suspicious claims. This saves money by catching fraud early.
* **Personalized Policies:** AI/ML can study policyholders' health, lifestyle, and more to understand risks. This means insurers can offer custom policies and adjust prices for specific health situations.
* **Better Customer Service:** AI-powered chatbots and helpers can answer customer questions, provide policy details, and help with claims. This makes customers happier and response times faster.

**Health Insurance Premium of Customers**

In the context of health insurance, predicting if customers will pay their health insurance premium is important. We can use machine learning to make accurate predictions by looking at customer info. This includes how they pay, age, income, past payments, and more. Machine learning finds patterns in this data. It helps predict if a customer will pay their next premium on time. This helps insurance companies plan better, understand risks, and assist customers, leading to better policy management and performance.

**Internship Project - Data Link**

The internship project data has been taken from Kaggle and the link is [https://www.kaggle.com/datasets/itssuru/health-insurance-premium-of-customers](https://www.kaggle.com/datasets/itssuru/health-insurance-premium-of-customers%20)

**AI / ML Modelling and Results**

**Problem Statement**

Your client is an Insurance company and they need your help in building a model to predict whether the policyholder (customer) will pay next premium on time or not. An insurance policy is an arrangement by which a company undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for the payment of a specified premium. A premium is a sum of money that you pay regularly to an insurance company for this guarantee.

For example, you may pay a premium of Rs. 5000 each year for a medical insurance cover of Rs. 200,000/- so that if, God forbid, you fall ill and need to be hospitalized in that year, the insurance provider company will bear the cost of hospitalization etc. for upto Rs. 200,000. Now if you are wondering how can company bear such high hospitalization cost when it charges a premium of only Rs. 5000/-, that is where the concept of probabilities comes in picture. For example, like you, there may be 100 customers who would be paying a premium of Rs. 5000 every year, but only a few of them (say 2-3) would get hospitalized that year and not everyone. This way everyone shares the risk of everyone else. Just like medical insurance, there is life insurance where every year you pay a premium of certain amount to insurance provider company so that in case of unfortunate event of your death, the insurance provider company will provide a compensation (called ‘sum assured’) to your immediate family. Similarly, there can be a variety of insurance products for different kinds of  
risks. As you can imagine, if a large number of customers do not pay the premium on time, it might disrupt the cash flow and smooth operation for the company. A customer may stop making regular premium payments for a variety of reasons - some may forget, some may find it expensive and not worth the value, some may not have money to pay the premium etc.

**What you have to do…**

Build a model to predict whether a customer would make the premium payment can be extremely helpful for the company because it can then accordingly plan its communication strategy to reach out to those customers who are less likely to pay and convince them to continue making timely payment.

**Given**

Now, in order to predict whether the customer would pay the next premium or not, you have information about past premium payment history for the policyholders along with their demographics (age, monthly income, area type) and sourcing channel etc.

**Data Exploratory Analysis**

The data exploratory analysis (EDA) performed on the data provided shows that the data is generally clean and well-formatted. There are a few missing values, but these are relatively few and can be easily imputed. The distribution of the data is also relatively normal, with a few outliers. There are a number of variables that are correlated, but these correlations are not strong enough to be indicative of multicollinearity. Overall, the EDA results suggest that the data is suitable for machine learning modeling.

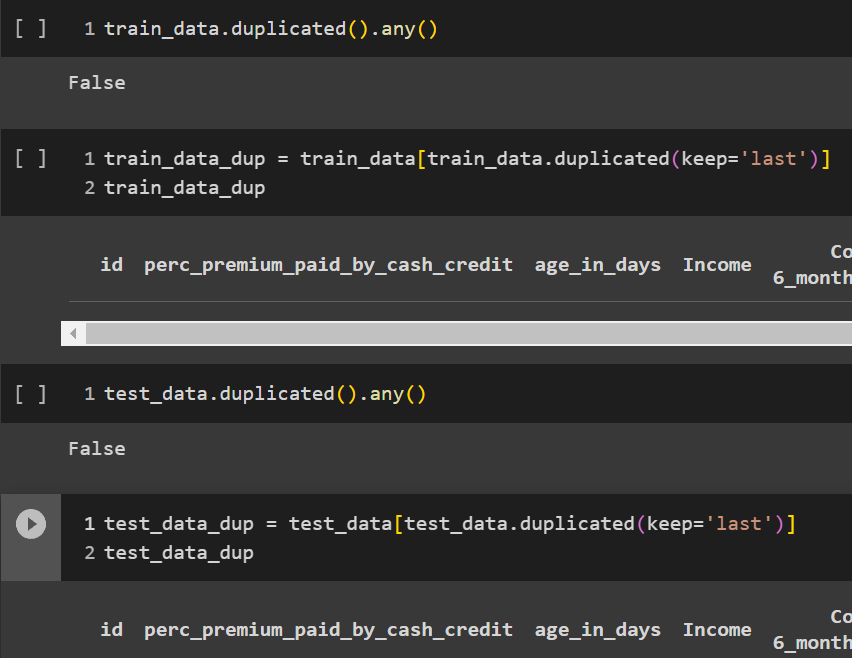
**Data Pre-processing**

* **Check for missing values:** The isnull().sum() method is used to count the number of missing values in each column. Any columns with missing values are then imputed using the imputer\_knn object.
* **Encode categorical variables:** The LabelEncoder object is used to encode categorical variables into numerical values. This is necessary for machine learning algorithms to work properly.
* **Scale the data:** The MinMaxScaler object is used to scale the data to a range of 0 to 1. This is done to ensure that all of the features have a similar impact on the machine learning models.

**Check the Duplicate and low variation data**

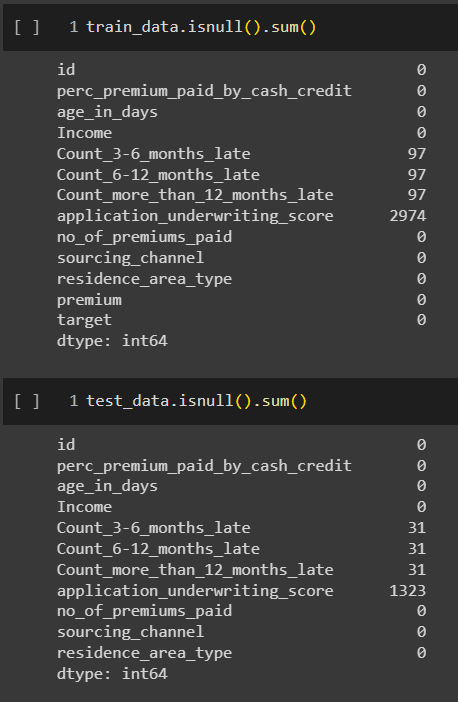
It is important to ensure that the data is clean and consistent, which can improve the performance of the machine learning models.

We can use the duplicated() method to check for duplicate rows in the data set. Any duplicate rows are then dropped from the data set.



**Identify and address the missing variables**

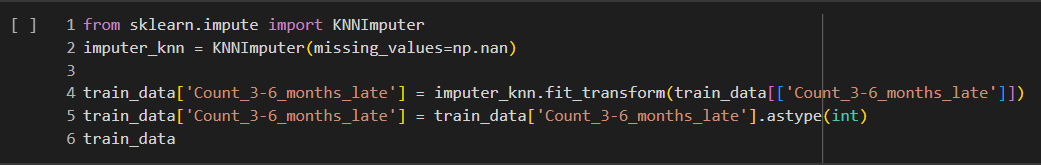
The key task is to identify and handle missing variables in the data. This involves finding any missing pieces of information in the dataset and taking steps to fill in those gaps. By addressing missing variables using techniques like imputation, the code ensures that all necessary data is available for accurate predictions and analysis. This helps in building robust models and making informed decisions in the context of health insurance premium prediction.

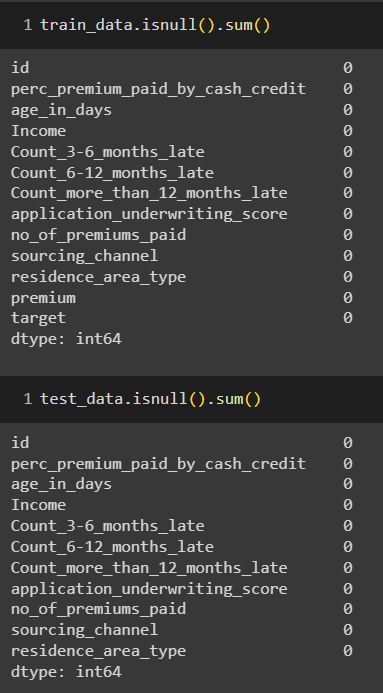


**Handling of Outliers**

**Impute outliers using KNN imputation:** The imputer\_knn object is used to impute the outliers. This is done by finding the nearest neighbors of the outliers and then imputing the outliers with the values of the nearest neighbors.

We can impute outliers using KNN imputation, which is a good way to replace outliers with more realistic values. KNN imputation works by finding the nearest neighbors of the outliers and then imputing the outliers with the values of the nearest neighbors.





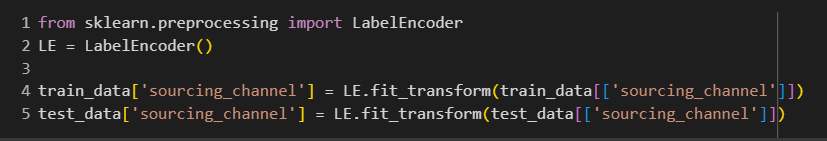
**Categorical data and Encoding Techniques**

Categorical data is data that can be classified into categories. For example, gender is a categorical variable that can be classified into two categories: male and female.

Encoding categorical data is the process of converting categorical data into numerical values so that it can be used by machine learning algorithms. There are a number of different encoding techniques that can be used, but some of the most common include:

* **Label encoding:** This is the simplest encoding technique. It involves assigning a unique integer value to each category. For example, the category "male" might be assigned the value 0 and the category "female" might be assigned the value 1.
* **One-hot encoding:** This is a more complex encoding technique that involves creating a separate binary variable for each category. For example, if there are two categories, "male" and "female," then two binary variables would be created: "male" and "female." The "male" variable would be set to 1 if the category is "male" and 0 otherwise. The "female" variable would be set to 1 if the category is "female" and 0 otherwise.

The best encoding technique to use will depend on the specific machine learning algorithm that is being used. Some algorithms, such as decision trees, can handle categorical data directly. However, other algorithms, such as logistic regression, require categorical data to be encoded.



**Feature Scaling**

Feature scaling is the process of standardizing or normalizing the numerical features in the dataset to ensure that they are on a similar scale. This step is important because many machine learning algorithms are sensitive to the scale of the input features, and having features with different scales can lead to biased or incorrect model results.

The code uses the MinMaxScaler from sklearn.preprocessing to perform feature scaling. Here's a brief explanation of how feature scaling is done in the code:

**Importing the Required Library:**

The MinMaxScaler class is imported from the sklearn.preprocessing module.

**Instantiating the Scaler:**

An instance of MinMaxScaler is created with the desired feature range (0 to 1 in this case) using the MinMaxScaler(feature\_range=(0, 1)) constructor.

**Scaling the Training Data**:

The fit\_transform method of the scaler is applied to the training data (x\_train) to calculate the scaling parameters and transform the data simultaneously.

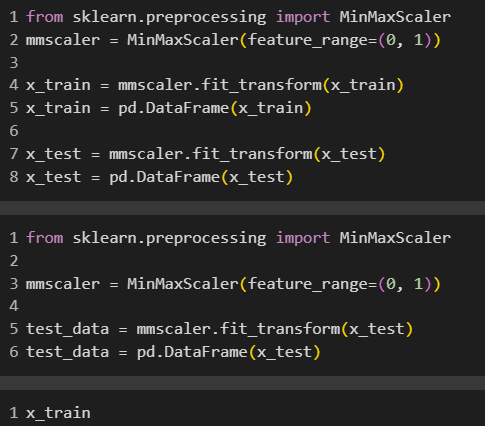
The scaled data is then converted back to a pandas DataFrame using pd.DataFrame().

**Scaling the Test Data**:

Similar to the training data, the fit\_transform method is used to scale the test data (x\_test).

The scaled data is converted back to a pandas DataFrame as well.

The purpose of feature scaling is to bring all the features to a similar range, typically between 0 and 1.



**Selection of Dependent and Independent variables**

In the code, we have a target variable called 'target.' It's what we want to predict.

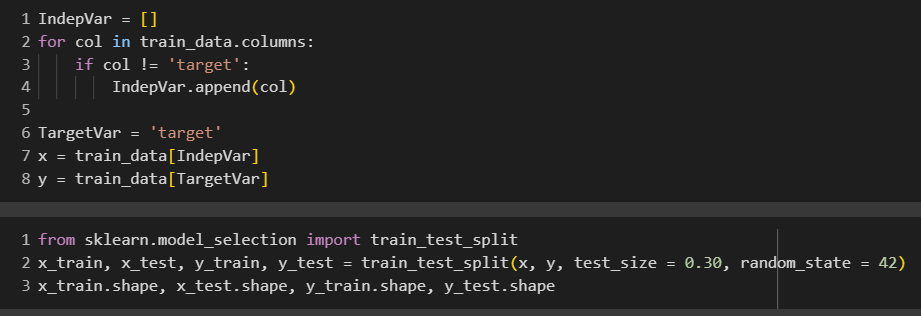
We also have independent variables, also known as features. These help predict the target. We create a list called IndepVar to keep their names.

We go through each column in the data. If it's not the target, we add its name to IndepVar.

The target variable (y) gets values from the 'target' column in the data.

The independent variables (x) get values from the columns listed in IndepVar.

Selecting these variables defines how the model works. Independent ones give info for prediction, and the dependent one is what we predict or understand.



**Data Sampling Methods**

For this project, I did not explicitly implement data sampling methods such as stratified sampling or simple random sampling. The dataset was used in its original form without any specific sampling techniques applied.

**Models Used for Development**

In the course of this analysis, I explored and evaluated a variety of machine learning models. These models included:

* Logistic Regression
* Decision Tree Classifier
* Random Forest Classifier
* Extra Trees Classifier
* K-Nearest Neighbors Classifier
* Gaussian Naive Bayes Classifier
* Support Vector Classifier

**AI / ML Models Analysis and Final Results**:

I conducted a comprehensive analysis of each machine learning model's performance. The key aspects of the analysis included:

* Confusion matrix to assess true positives, false negatives, false positives, and true negatives.
* Classification report providing insights into precision, recall, F1-score, and overall accuracy.
* Sensitivity and specificity measurements.
* Balanced accuracy to gauge model performance across classes.
* Matthews Correlation Coefficient (MCC) as a robust indicator of classification quality.
* Receiver Operating Characteristic (ROC) curves and ROC Area Under the Curve (AUC) scores to evaluate the model's ability to distinguish between classes.

The results and performance metrics of these models were meticulously documented and recorded in the analysis. Specifically, the following metrics were collected for each model:

* **Accuracy:** Reflecting the overall model correctness.
* **Precision:** Indicating the model's ability to make accurate positive predictions.
* **Recall:** Measuring the proportion of actual positives correctly identified by the model.
* **F1 Score:** Harmonizing precision and recall into a single score.
* **Specificity**: Representing the true negative rate.
* **MCC:** Providing a comprehensive assessment of the model's quality.
* **ROC AUC Score:** Quantifying the model's ability to discriminate between classes.
* **Balanced Accuracy:** Ensuring fair assessment across different classes.

**Conclusions and Future Work:**

In summary, our analysis covered an extensive range of machine learning models. It also highlighted their strengths and weaknesses. No specific data sampling techniques were employed in this analysis.

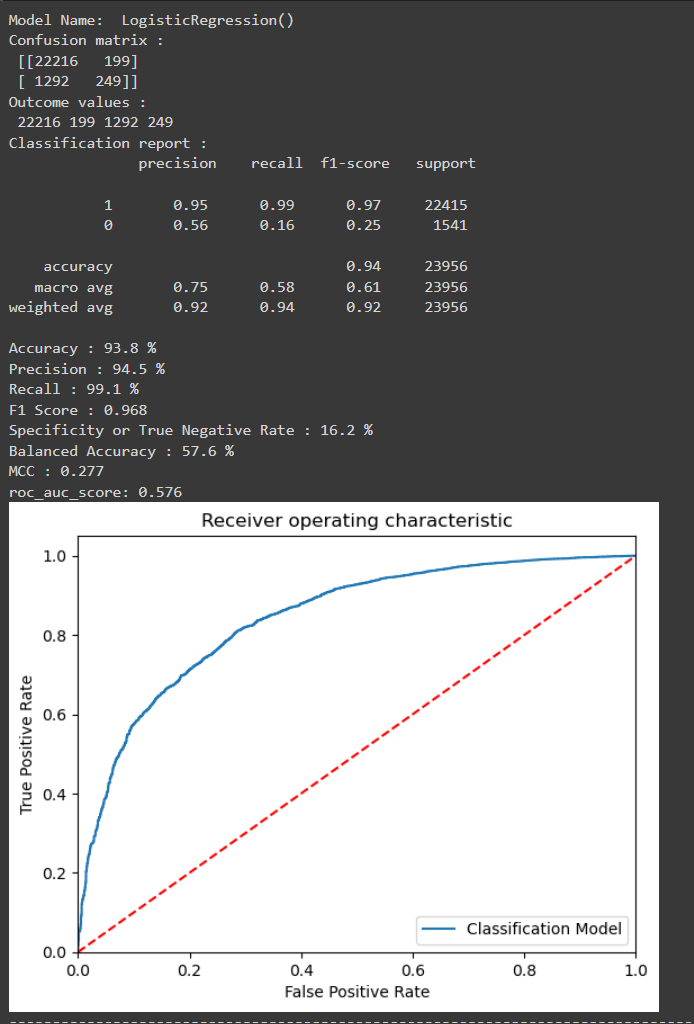
We recommend the model - Logistic Regression model. This model is good for predicting things, like whether customers will pay their insurance premiums on time, based on the information we have. It's a helpful tool in our code for making these predictions accurately with an accuracy of 93.8%.

For future work, we recommend:

* Fine-tuning feature engineering and selection.
* Exploring hyperparameter optimization.
* Experimenting with ensemble methods.
* Implementing cross-validation techniques.
* Continuously monitoring and updating models with new data.

**References:**

* [*https://en.wikipedia.org/wiki/Health\_insurance*](https://en.wikipedia.org/wiki/Health_insurance)
* [*https://en.wikipedia.org/wiki/Machine\_learning\*](https://en.wikipedia.org/wiki/Machine_learning\)
* [*https://www.sas.com/en\_us/insights/articles/analytics/insurance-premium-predictions.html*](https://www.sas.com/en_us/insights/articles/analytics/insurance-premium-predictions.html)
* [*https://www.researchgate.net/publication/317156501\_Machine\_Learning\_for\_Predicting\_Insurance\_Premium\_Default*](https://www.researchgate.net/publication/317156501_Machine_Learning_for_Predicting_Insurance_Premium_Default)
* [*https://www.actuary.org/sites/default/files/files/publications/Predictive\_Modeling\_Guide\_June2017.pdf*](https://www.actuary.org/sites/default/files/files/publications/Predictive_Modeling_Guide_June2017.pdf)
* [*https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6678439/*](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6678439/)
* [*https://towardsdatascience.com/using-machine-learning-to-predict-healthcare-costs-8d7047dc15fe*](https://towardsdatascience.com/using-machine-learning-to-predict-healthcare-costs-8d7047dc15fe)
* [*https://www.casact.org/sites/default/files/ambassador/2020/2020\_2.pdf*](https://www.casact.org/sites/default/files/ambassador/2020/2020_2.pdf)

**Python code results:**

