PROJECT PHASE 2

Bhavani Kukunoor - 50560517
Surya Venkata Rohit Moganti - 50560088

Overview of Phase 1:

In project phase one we have done a lot of data cleaning to our original dataset in order to train the data with various models. We have removed all the null values using various imputation methods. The outliers in the policy premium column are removed using IQR and all other data inconsistencies like datatypes and categorical values were handled perfectly. We have taken all the numerical columns into consideration in order to train our model.

Phase 2 Introduction:

The primary objective of our model is to use predictive analysis to improve lead prediction for health insurance companies. Many health insurance companies currently have trouble finding new clients effectively, which results in low conversion rates and revenue loss. Through user behavior analysis and lead conversion prediction from website visits, the goal is to predict the number of people who are most likely to choose health insurance. In the cutthroat health insurance market, obtaining leads effectively is essential to business success. In order to raise conversion rates and income, the model seeks to enhance lead prediction performance.

Six major and relevant algorithms were applied to the cleaned lead insurance dataset in our project. These algorithms cover a wide range of statistical, regression, and machine learning models, each having a special method for evaluating the data and generating predictions. The objective of this section is to apply these techniques to examine the information from a modeling approach and provide insightful conclusions. By using a variety of models, we can evaluate how well various modeling approaches forecast lead consumers and develop a thorough grasp of the dataset. This section employs a number of algorithms, including decision tree, random forest, K-Nearest Neighbors, logistic regression, linear regression, and neural network models.

By evaluating the performance of each model, which has unique benefits and features, we can determine which method is best for lead prediction in health insurance firms. We not only put the algorithms into practice but also made illustrations to efficiently convey and understand the outcomes. In order to comprehend the complex relationships, patterns, and trends included in the data, visualizations are essential. Our algorithms' outputs can be visually represented, which helps us present our results more clearly and make a thorough examination of the dataset easier. We hope to demonstrate in this section our expertise with various algorithms, our understanding of their advantages and disadvantages, and our capacity to derive valuable insights from the health insurance lead prediction dataset. With the aid of visualizations and computational analysis, we are able to offer an in-depth evaluation of the dataset and provide valuable information that can guide decision-making in the insurance markets.

Feature selection and Data Preprocessing : To make sure the data is appropriate for applying all the algorithms, we converted the text data to numeric data. It is a crucial step in preparing data for machine learning algorithms. All these numeric columns are taken into consideration. In our dataset we have two different target classes, (1 for lead , 0 otherwise) the classes are biased towards zero i.e we have more number of rows that classify zero. We performed sklearn

1. Linear Regression:

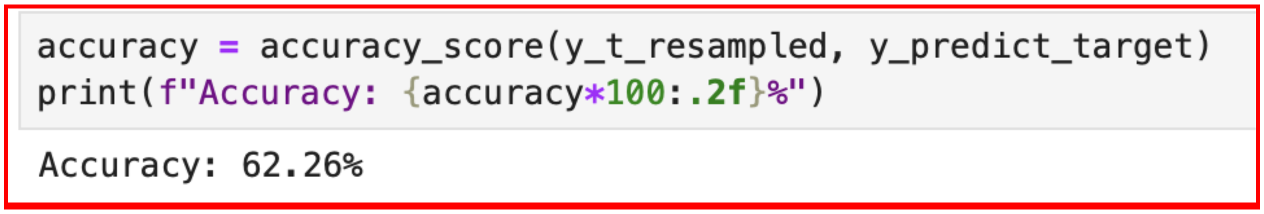
Linear Regression provides easily interpretable results. The coefficients of the linear model represent the relationship between the independent variables (features) and the dependent variable (lead conversions), allowing for straightforward interpretation of the impact of each feature on lead conversions.
Linear Regression is relatively simple to implement and computationally efficient. It does not require complex parameter tuning or training procedures, making it suitable for quick prototyping and deployment in production environments.

To Train the Model:

Model Training and Evaluation: We first split the dataset into training and testing sets. The Linear Regression model was trained on the training set . We have trained the model in two different ways, one with a scaled training set and the other with a non scaled training set and we were able to achieve high accuracy using the non scaled training data.
After the model training we check our model on the testing data and we obtain the results in the form of probability. Since our problem is a classification problem we then add a threshold to our model inorder to predict the final target variable. In our code we have set the threshold to 0.5 and all the values predicted by our model which are greater than 0.5 are labeled as 1 and all the others as 0 . In this way we can use the potential of linear regression for classification problems.

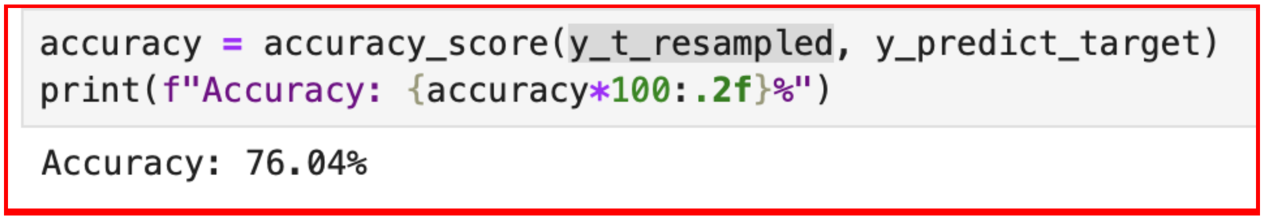
Effectiveness of the Algorithm:

By analyzing the coefficients of the linear model, intelligence can be gained regarding the relationship between each feature and lead conversions. Features with larger coefficients have a greater impact on lead conversions, while features with smaller coefficients have a smaller impact.



```
accuracy = accuracy_score(y_t_resampled, y_predict_target)
print(f"Accuracy: {accuracy*100:.2f}%")

Accuracy: 62.26%
```

An accuracy of 62.26% is achieved for scaled data



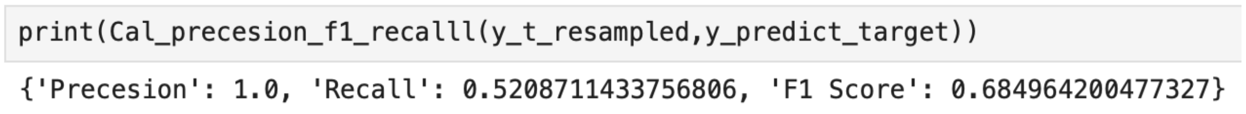
```
accuracy = accuracy_score(y_t_resampled, y_predict_target)
print(f"Accuracy: {accuracy*100:.2f}%")

Accuracy: 76.04%
```

For non-scaled data, the accuracy is raised to 76.04%.

Presion: The ratio of accurately predicted positive observations to all expected positives is known as precision. It measures how accurate positive forecasts are.

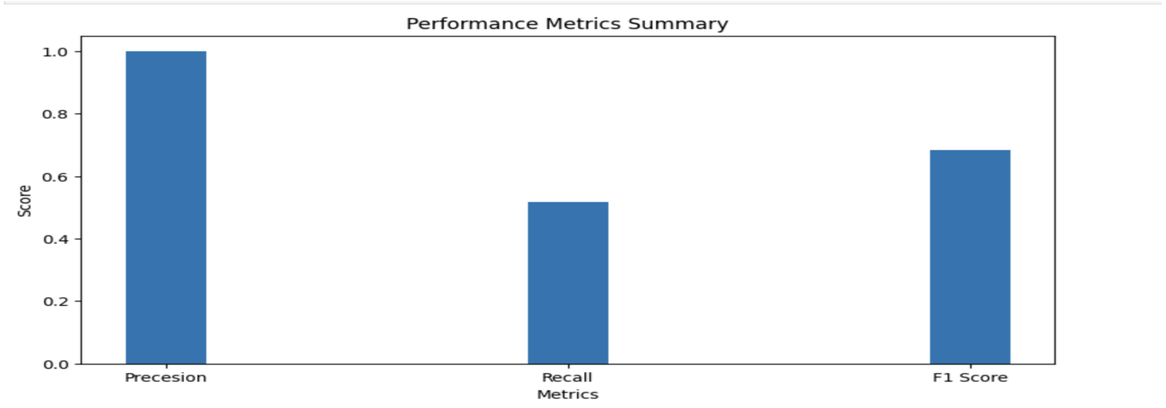
Recall: The ratio of accurately predicted positive observations to all observations made during the actual class is known as recall. It evaluates the classifier's capacity to identify every positive sample.

F1 Score: The precision and recall mean is the F1 score. It offers a balance between recall and precision.
F1 score reaches its best value at 1 and worst at 0.

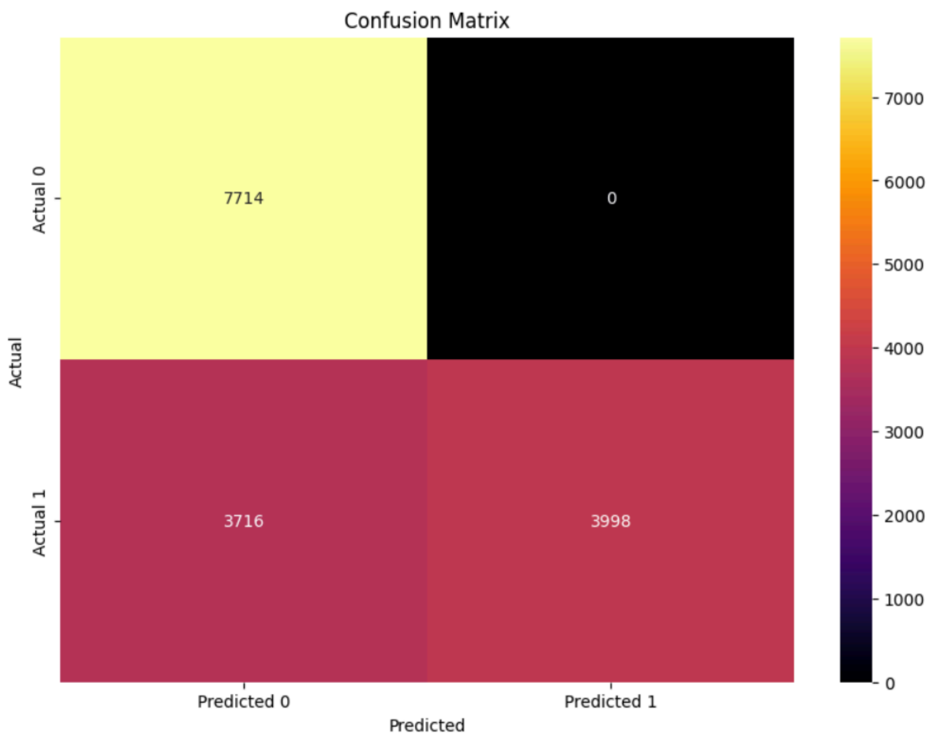


```
print(Cal_precesion_f1_recalll(y_t_resampled,y_predict_target))

{'Precesion': 1.0, 'Recall': 0.5208711433756806, 'F1 Score': 0.684964200477327}
```

Visualization :



Confusion matrix: Is a tabular representation of the counts of true positive, true negative, false positive, and false negative predictions that helps illustrate the effectiveness of the Linear regression model.



In the confusion matrix it can be seen that, when the actual value is 0, the model correctly predicted the value 0, 7714 times. when the actual value is 1, the model correctly predicted the value 1, 3998 times. When the value is 0, the model predicted 1, 0 times. When the value is 1, the model predicted 0, 3716 times.

2. Logistic Regression:

One of the approaches for binary classification tasks that can be used to forecast whether or not a consumer would choose health insurance is logistic regression. Because it offers probabilities of class membership, it is simple to use and straightforward to interpret.

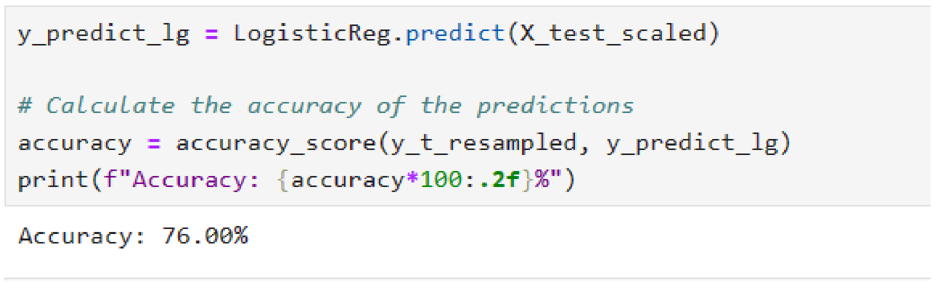
Each feature's effect on a person's likelihood of choosing health insurance is indicated by the logistic regression model's coefficients. Understanding the variables impacting consumer decisions is made easier with the help of this interpretability.

To Train the Model:

Model Training and Evaluation: We have first resampled and scaled the training data before feeding it to the model. Since Logistic regression is more effective on classification problems than linear regression we have fitted our data to the logistic regression model, In this we have given the regularization value as 0.01 which inturn results in higher accuracy and also avoids overfitting of Data. We have also utilized the capability of newton-cg optimization algorithm by specifying it in the parameters while model initialization. After training our model with the training dataset we have predicted the target values using the test feature set and have obtained an accuracy of 76%. This clearly states that the model is effective in identifying the separable data points. There is 24% error in the model which is because of the complex feature relationships on the target where the Logistic regression model struggles.

Effectiveness of the Algorithm:

An accuracy of 76 % is obtained while performing the Logistic Regression algorithm.



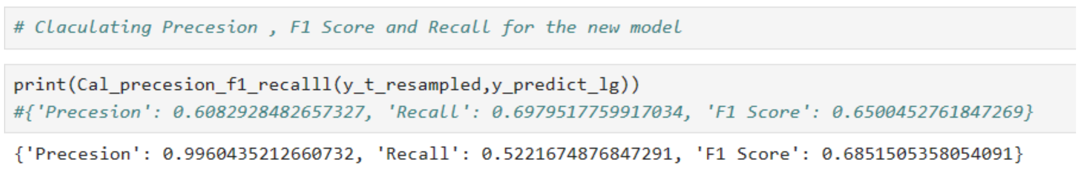
```
y_predict_lg = LogisticReg.predict(X_test_scaled)

# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_t_resampled, y_predict_lg)
print(f"Accuracy: {accuracy*100:.2f}%")
```

```
Accuracy: 76.00%
```

Presion: The ratio of accurately predicted positive observations to all expected positives is known as precision. It measures how accurate positive forecasts are.

Recall: The ratio of accurately predicted positive observations to all observations made during the actual class is known as recall. It evaluates the classifier's capacity to identify every positive sample.

F1 Score: The precision and recall mean is the F1 score. It offers a balance between recall and precision.
F1 score reaches its best value at 1 and worst at 0.

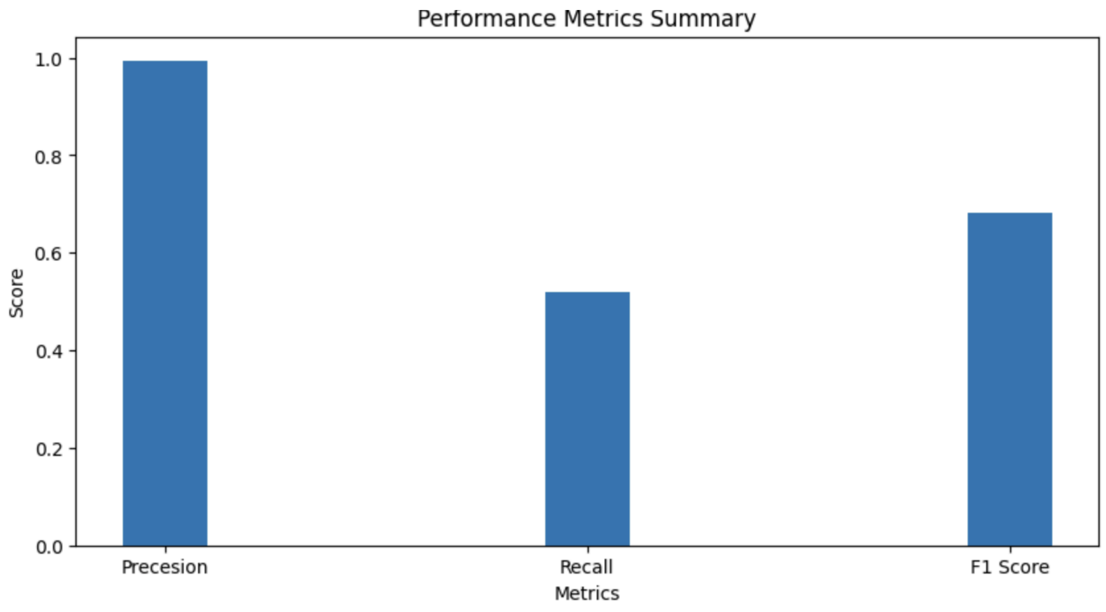


```
# Claculating Precesion , F1 Score and Recall for the new model
```

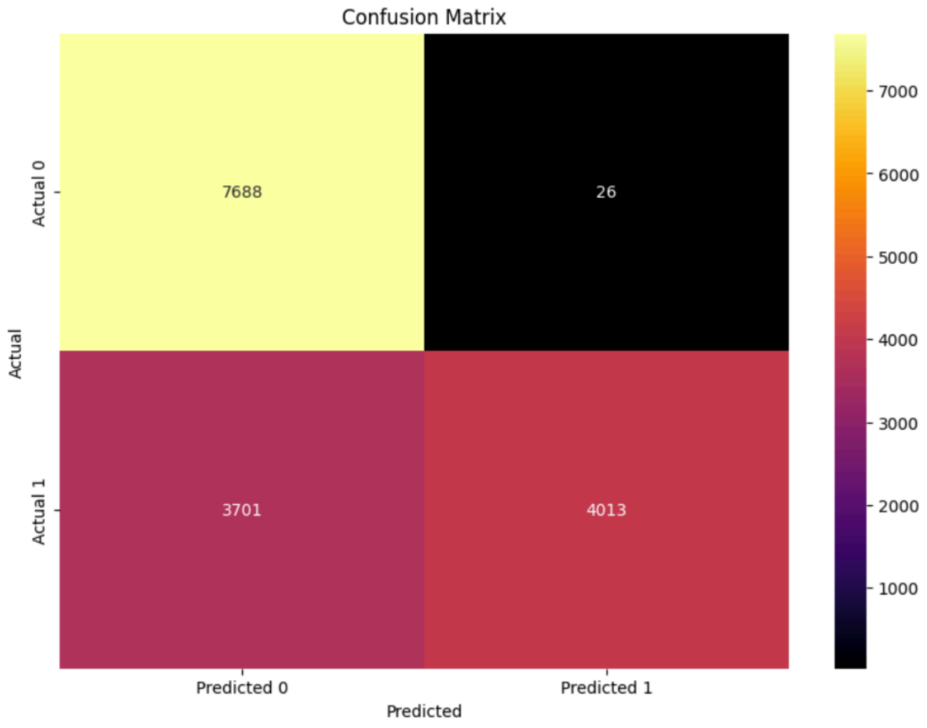
```
print(Cal_precesion_f1_recalll(y_t_resampled,y_predict_lg))
#{'Precesion': 0.6082928482657327, 'Recall': 0.6979517759917034, 'F1 Score': 0.6500452761847269}
```

```
{'Precesion': 0.9960435212660732, 'Recall': 0.5221674876847291, 'F1 Score': 0.6851505358054091}
```

Visualization :



Confusion matrix: Is a tabular representation of the counts of true positive, true negative, false positive, and false negative predictions that helps illustrate the effectiveness of the Logistic Regression model.



In the confusion matrix it can be seen that, when the actual value is 0, the model correctly predicted the value 0, 7688 times. when the actual value is 1, the model correctly predicted the value 1, 4013 times. When the value is 0, the model predicted 1, 26 times. When the value is 1, the model predicted 0, 3701 times.

3. Naive Bayes:

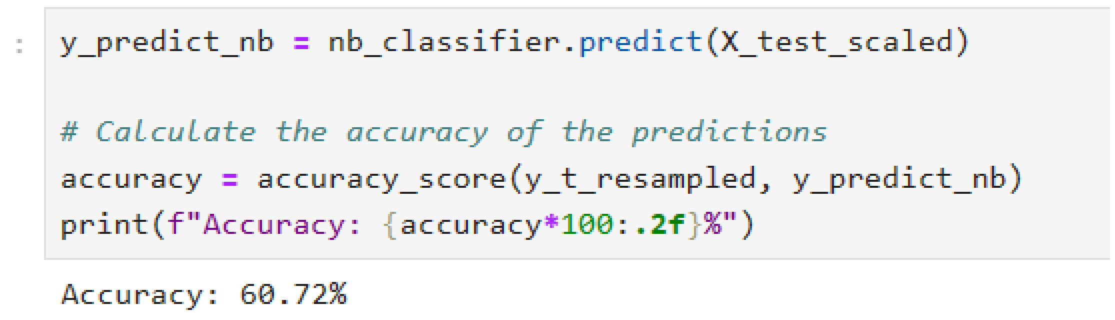
The probabilistic classifier Naive Bayes is straightforward and efficient when applied to categorical features. It makes the logical assumption that each predictor is independent in many real-world situations.

To Train the Model:

Model Training and Evaluation: For training the previously scaled and resampled data to a naive bayes classifier we have taken advantage of sklearn’s Naive bayes classifier. Specifically the GaussianNB. Unlike MultinomialNB which is excellent for text classification, GaussianNB is good at classifying Numerical Data.We have added a var_smoothing parameter which is a smoothing parameter added to variance for stability .After fitting our model with the training data we have predicted the target values using a testing feature set and have achieved an accuracy of 60% which is considerably low. This low accuracy is because the Naive Bayes model by default assumes that the given features are independent to each other and hence it is unable to make a good relationship of the features to the target.

Effectiveness of the Algorithm:

An accuracy of 60.72% is obtained by performing Naive Bayes algorithm.



```
y_predict_nb = nb_classifier.predict(X_test_scaled)

# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_t_resampled, y_predict_nb)
print(f"Accuracy: {accuracy*100:.2f}%")
```

```
Accuracy: 60.72%
```

Presion: The ratio of accurately predicted positive observations to all expected positives is known as precision. It measures how accurate positive forecasts are.

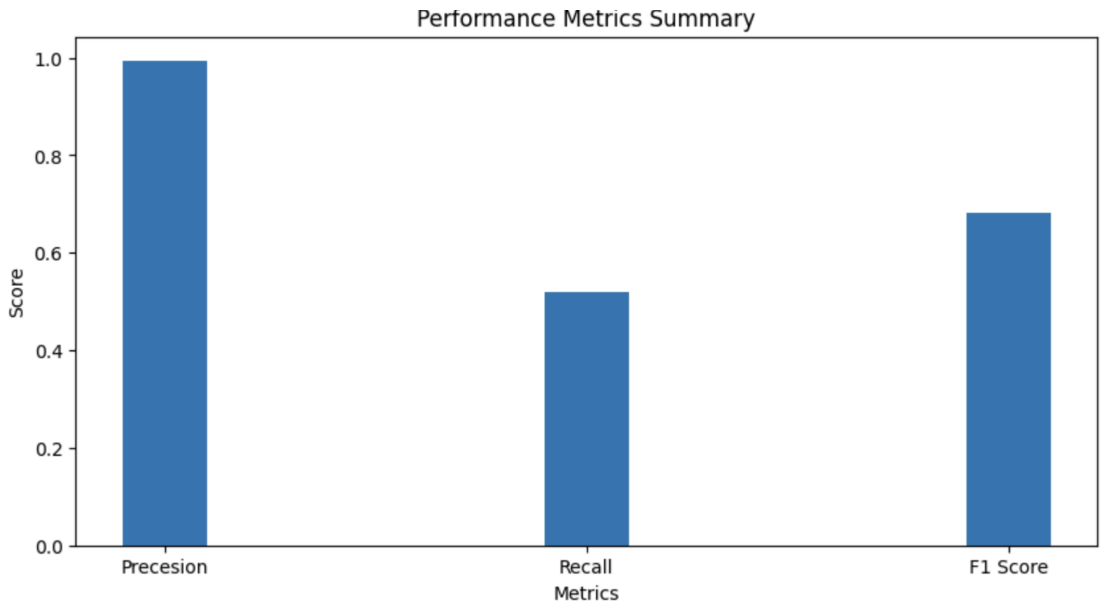
Recall: The ratio of accurately predicted positive observations to all observations made during the actual class is known as recall. It evaluates the classifier's capacity to identify every positive sample.

F1 Score: The precision and recall mean is the F1 score. It offers a balance between recall and precision.
F1 score reaches its best value at 1 and worst at 0.

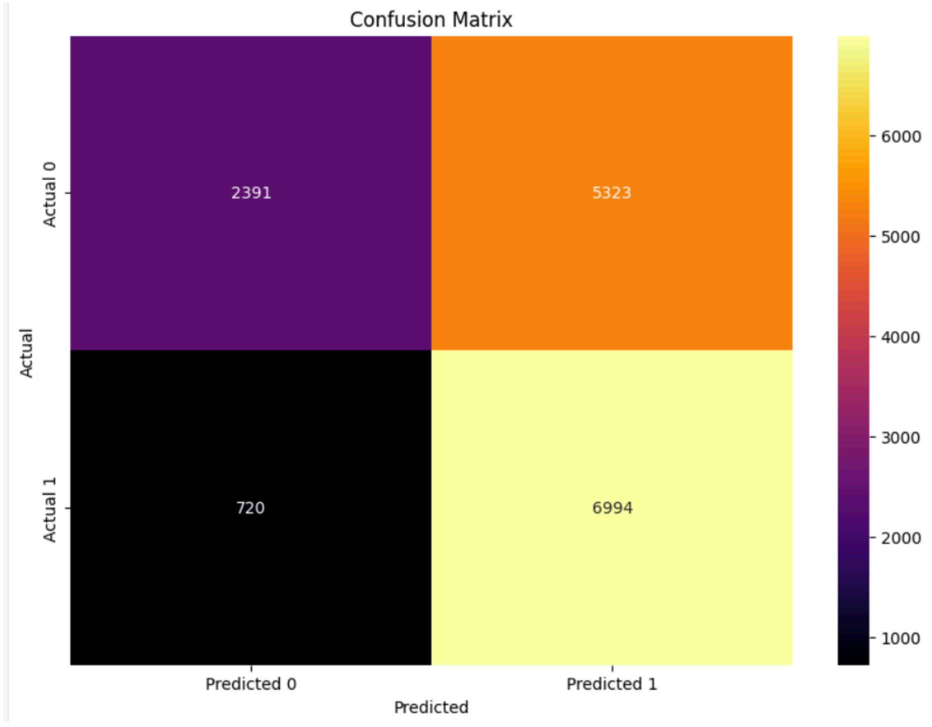
```
print(Cal_precesion_f1_recalll(y_t_resampled,y_predict_nb))
```

```
{'Precesion': 0.5667366042608134, 'Recall': 0.9104226082447499, 'F1 Score': 0.6985974336019098}
```

Visualization :



Confusion matrix: Is a tabular representation of the counts of true positive, true negative, false positive, and false negative predictions that helps illustrate the effectiveness of the Naive Bayes model.



Metrics such as accuracy, precision, recall, F1-score, and were used to evaluate the effectiveness of the Naive Bayes model in predicting lead conversions for health insurance.

In the confusion matrix it can be seen that, when the actual value is 0, the model correctly predicted the value 0, 2391 times. when the actual value is 1, the model correctly predicted the value 1, 6994 times. When the value is 0, the model predicted 1, 5323 times. When the value is 1, the model predicted 0, 720 times.

4. K-Nearest Neighbors (KNN):

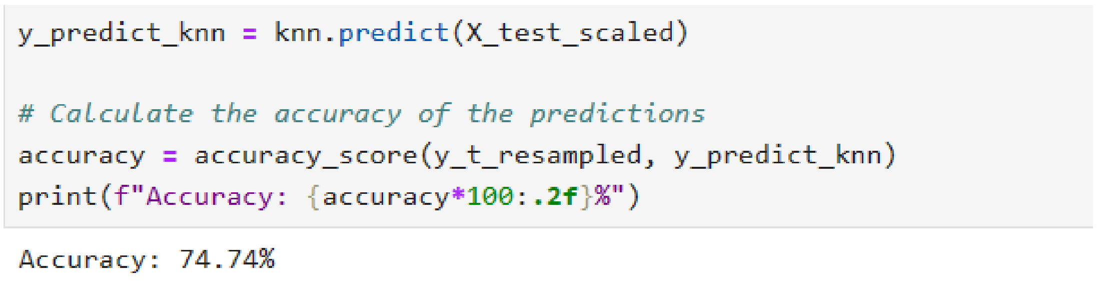
KNN is an instance-based learning algorithm, where predictions are made based on the similarity of new instances to existing data points in the feature space. In the context of health insurance lead prediction, this approach can be beneficial for identifying patterns in customer behavior.

To Train the Model:

Model Training and Evaluation: We are fitting the already scaled and resampled data for the KNN model imported from sklearn . For initializing the KNN classifier we need to specify the number of neighbors for each new data point for it to be classified as a lead or not. After continuous testing and accuracy analysis we have decided that the neighbor value of 10 gives us more accuracy compared to other neighbor values. We have also given hyper parameters to while model initialization and they are weights = distance , this ensures that the closer values to the data points are given more importance and distance = euclidean, this ensures that the distance measured between points is euclidean distance.The main advantage of this algorithm is that it does not make any presumptions on the dataset as this algorithm is a non parametric algorithm. After training the dataset we have predicted the values using the testing data and we were able to achieve an accuracy of 74%.

Effectiveness of the Algorithm:

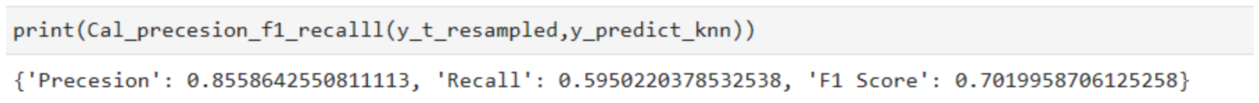
An accuracy of 74.7% is obtained by performing KNN.



```
y_predict_knn = knn.predict(X_test_scaled)

# Calculate the accuracy of the predictions
accuracy = accuracy_score(y_t_resampled, y_predict_knn)
print(f"Accuracy: {accuracy*100:.2f}%")
```

```
Accuracy: 74.74%
```

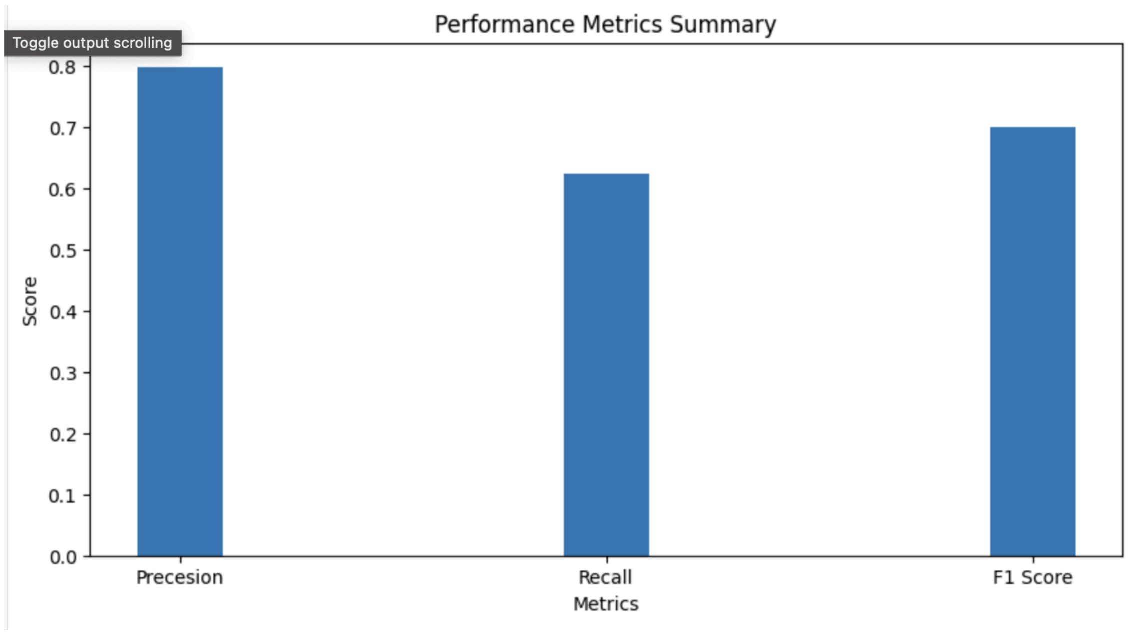
Presion: The ratio of accurately predicted positive observations to all expected positives is known as precision. It measures how accurate positive forecasts are.
Recall: The ratio of accurately predicted positive observations to all observations made during the actual class is known as recall. It evaluates the classifier's capacity to identify every positive sample.
F1 Score: The precision and recall mean is the F1 score. It offers a balance between recall and precision. F1 score reaches its best value at 1 and worst at 0.



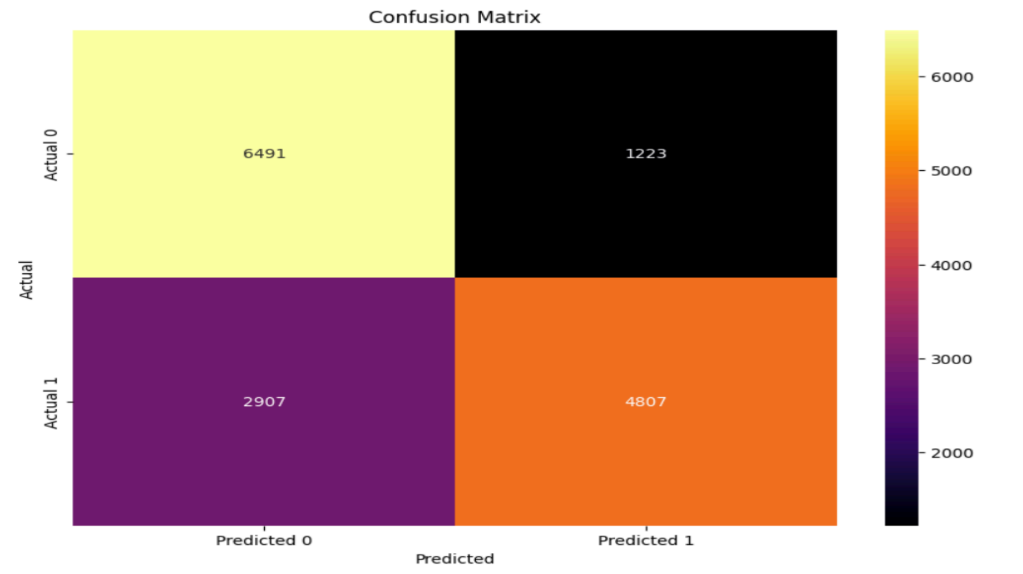
```
print(Cal_precesion_f1_recalll(y_t_resampled,y_predict_knn))
```

```
{'Precesion': 0.8558642550811113, 'Recall': 0.5950220378532538, 'F1 Score': 0.7019958706125258}
```

Visualization :



Confusion matrix: Is a tabular representation of the counts of true positive, true negative, false positive, and false negative predictions that helps illustrate the effectiveness of the KNN model.



Metrics such as accuracy, precision, recall, F1-score were used to evaluate the effectiveness of the KNN model in predicting lead conversions for health insurance.

In the confusion matrix it can be seen that, when the actual value is 0, the model correctly predicted the value 0, 6491 times. when the actual value is 1, the model correctly predicted the value 1, 4807 times. When the value is 0, the model predicted 1, 1223 times. When the value is 1, the model predicted 0, 2907 times.

5. K-Means:

K-Means is an unsupervised learning algorithm used for clustering. It groups similar data points together based on their features without needing labeled data. In the context of health insurance lead prediction, clustering can help identify distinct groups or segments of customers based on their characteristics.

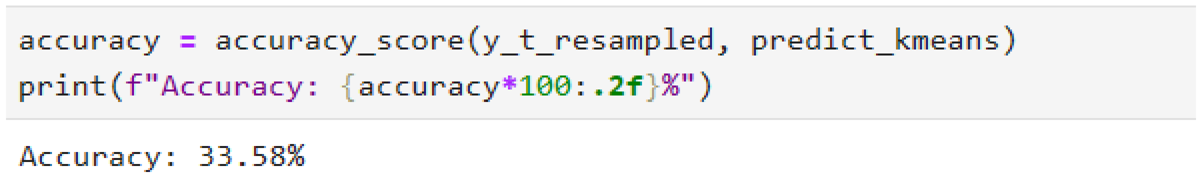
K-Means is computationally efficient and scales well with large datasets. It can handle a large number of data points and features, making it suitable for analyzing customer data in the health insurance sector, which involve a large volume of data.

To Train the Model:

Model Training and Evaluation : We are training this model using the resampled and scaled data First we have to decide the number of clusters that we are using in this particular algorithm . This is a crucial part and the model accuracy completely depends on choosing the number of clusters. Since this is an unsupervised learning algorithm the model training does not require any target values. We have feeded the model with the training feature set and have tested the model using the testing feature set. At first we have set the value for number of clusters as 3 and have achieved an accuracy of 30% and by increasing the value of number of clusters we have observed a major drop in the accuracy of our model. After continuous testing we have concluded the value of the number of clusters as 2 and we have achieved an accuracy of 51%. Since our dataset has various feature types and the K-means model not capable of interpreting the relationships between the feature attributes makes this model not suitable for predicting specific classes of this data.

Effectiveness of the Algorithm:

An accuracy of 33.58% is obtained by performing the K-means algorithm on our model

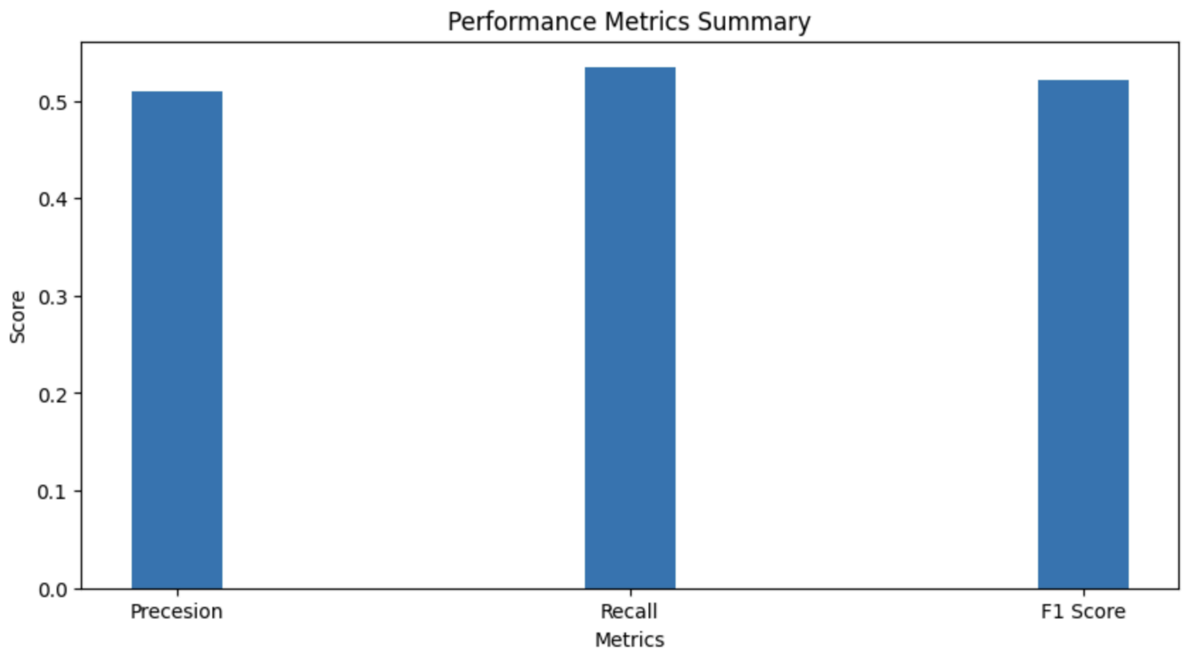


```
accuracy = accuracy_score(y_t_resampled, predict_kmeans)
print(f"Accuracy: {accuracy*100:.2f}%")
```

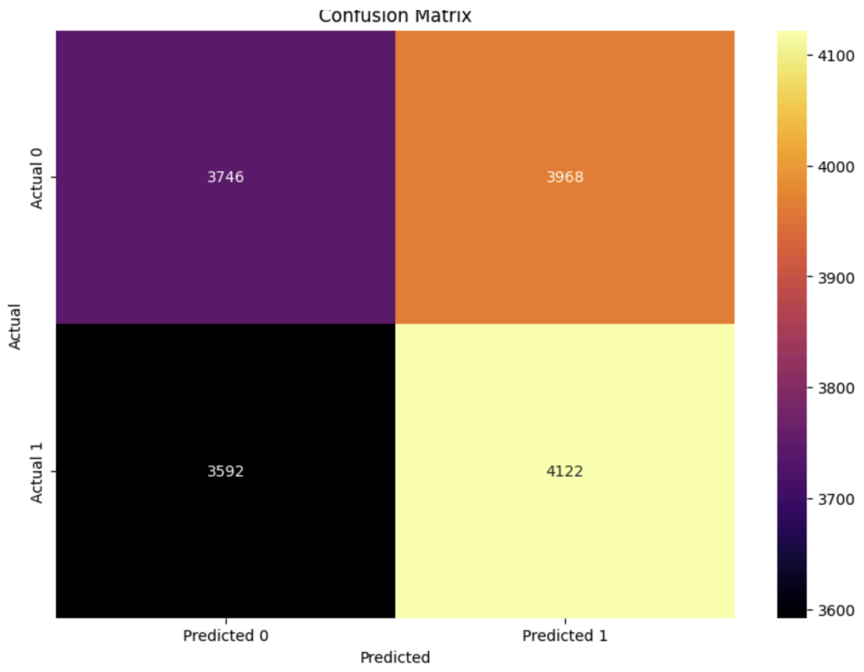
```
Accuracy: 33.58%
```

Model Performance Metrics: While K-Means does not have traditional performance metrics like accuracy or F1-score since it's an unsupervised learning algorithm, it is not very effective.

Visualization :



Confusion matrix: Is a tabular representation of the counts of true positive, true negative, false positive, and false negative predictions that helps illustrate the effectiveness of the K-means model.



In the confusion matrix it can be seen that, when the actual value is 0, the model correctly predicted the value 0, 6491 times. when the actual value is 1, the model correctly predicted the value 1, 4807 times. When the value is 0, the model predicted 1, 1223 times. When the value is 1, the model predicted 0, 2907 times.

6. Decision Tree:

Decision Trees provide easily interpretable results that can be understood even by non-technical stakeholders. This is valuable in the context of health insurance lead prediction, where decision-making processes need to be transparent and understandable.

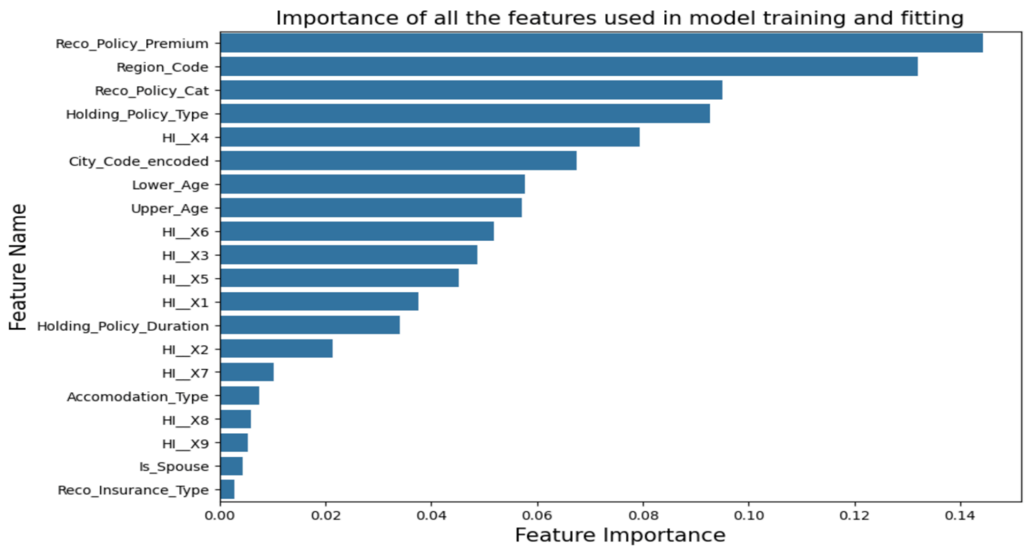
Decision Trees can capture non-linear relationships and interaction effects between features, making them suitable for modeling complex decision-making processes in the health insurance sector.

To Train the Model:

Model Training and Evaluation: We have trained the Decision tree model imported from sklearn with the previously scaled and resampled data.While initializing the Decision tree classifier we have give the parameters for criterion as entropy which depicts the quality of the split which happens while training the model and splitter as best which ensures the best split happens while splitting the decision tree.After Model training we have tested the model with the resampled testing data.

Effectiveness of the Algorithm:

This graph depicts the importance of each individual feature that is involved in model training.



An accuracy of 73.58% is achieved by performing a decision tree algorithm on the model.

```
y_predict_dtree = dt_classifier.predict(X_test_scaled)
```

```
accuracy = accuracy_score(y_t_resampled, y_predict_dtree)
print(f"Accuracy: {accuracy*100:.2f}%")
```

```
Accuracy: 73.58%
```

Presion: The ratio of accurately predicted positive observations to all expected positives is known as precision. It measures how accurate positive forecasts are.

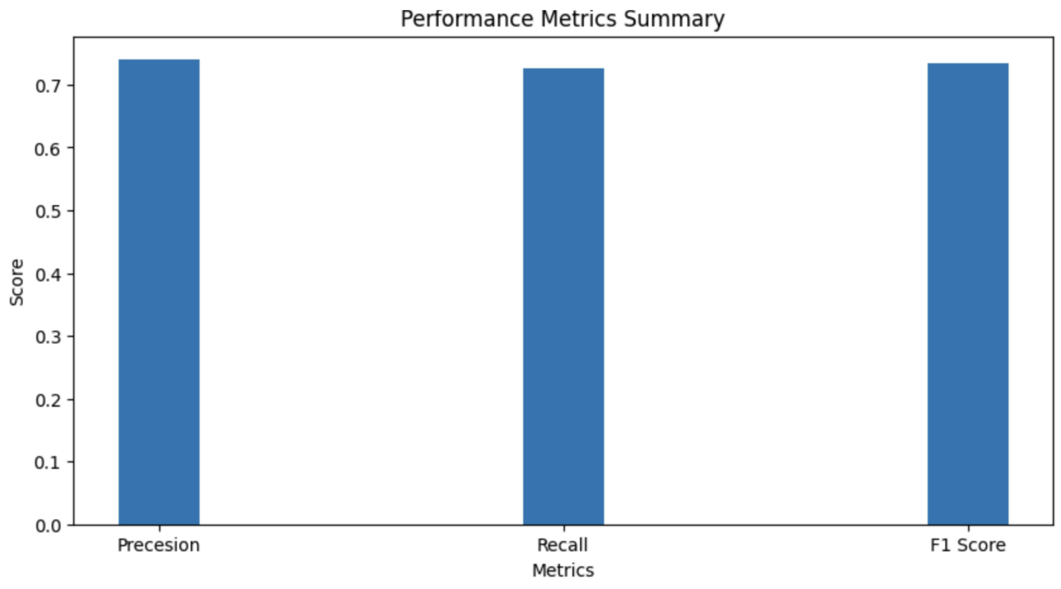
Recall: The ratio of accurately predicted positive observations to all observations made during the actual class is known as recall. It evaluates the classifier's capacity to identify every positive sample.

F1 Score: The precision and recall mean is the F1 score. It offers a balance between recall and precision.
F1 score reaches its best value at 1 and worst at 0.

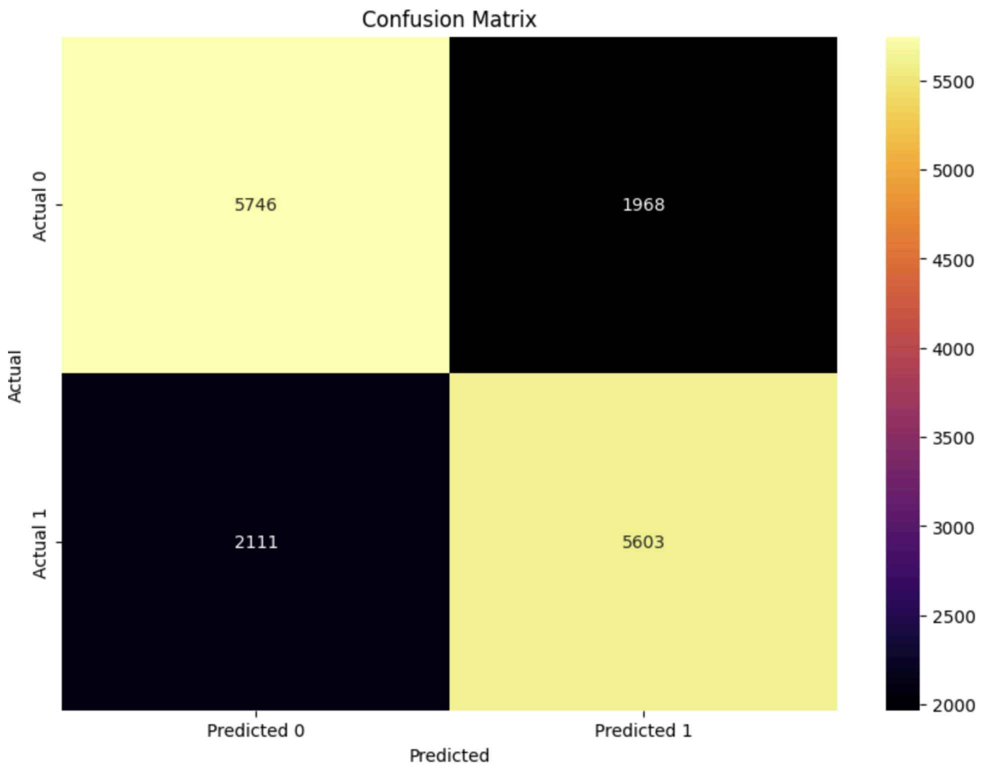
```
print(Cal_precesion_f1_recalll(y_t_resampled,y_predict_dtree))
```

```
{'Precesion': 0.737653514502221, 'Recall': 0.7319159968887736, 'F1 Score': 0.7347735554398751}
```

Visualization :



Confusion matrix: Is a tabular representation of the counts of true positive, true negative, false positive, and false negative predictions that helps illustrate the effectiveness of the decision tree model.



In the confusion matrix it can be seen that, when the actual value is 0, the model correctly predicted the value 0, 6491 times. when the actual value is 1, the model correctly predicted the value 1, 4807 times. When the value is 0, the model predicted 1, 1223 times. When the value is 1, the model predicted 0, 2907 times.

Decision Trees provide a set of rules that can be easily interpreted to understand the factors influencing lead conversions. By analyzing the decision paths, intelligence can be gained regarding the decision-making process and key features driving customer behavior.

Decision Trees are prone to overfitting, especially with deep trees. Regularization techniques such as pruning or using ensemble methods like Random Forests.

7. Neural network:

Neural networks are a good model for the complicated decision-making processes involved in health insurance lead prediction because they can capture non-linear correlations and complex interactions between features.

By automatically extracting features from the data, neural networks might possibly reveal links and patterns that would not be visible through manual feature engineering.

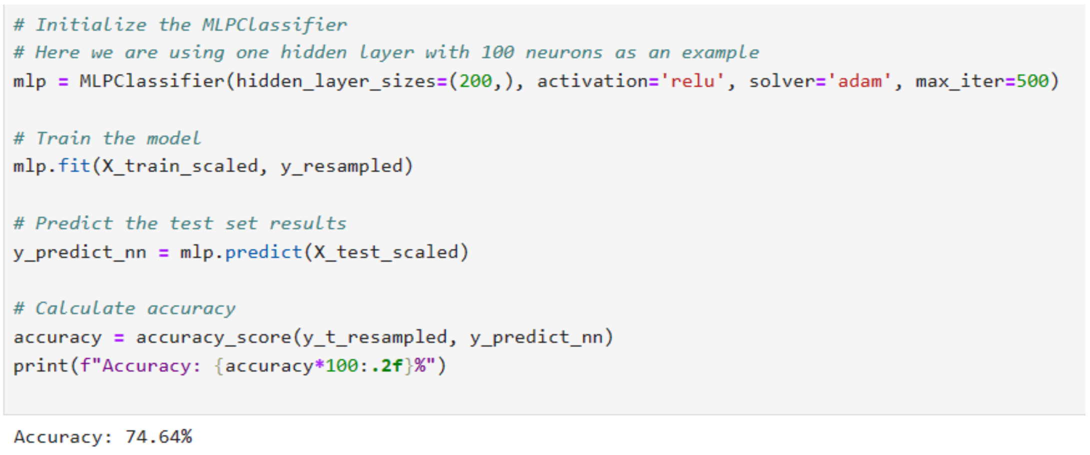
Large datasets and high-dimensional feature spaces, which are typical in health insurance datasets with lots of variables and observations, are good options for neural networks' scalability.

To Train the model:

Model Training and Evaluation : Initially we feed our scaled and resampled data to this algorithm . While training the model, first the weight and biases of all the nodes in the hidden layers are initialized to a random value. After one full run of the neural network the results are obtained and based on the results the weights and biases across all the nodes are updated through Backpropagation. In our algorithm we use the activation function “relu”. We also used Adam optimization technique which adjusts the learning rate of the algorithm for each iteration thereby increasing the efficiency of the algorithm. We have also utilized the feature of early stopping while defining the Neural network which ensures that the neural net is stopped when there is no change in the loss.This is effective while training large data to reduce load on the processing.

Effectiveness of the algorithm:

An accuracy of 74.64% is obtained while performing a neural network algorithm.



```
# Initialize the MLPClassifier
# Here we are using one hidden layer with 100 neurons as an example
mlp = MLPClassifier(hidden_layer_sizes=(200,), activation='relu', solver='adam', max_iter=500)

# Train the model
mlp.fit(X_train_scaled, y_resampled)

# Predict the test set results
y_predict_nn = mlp.predict(X_test_scaled)

# Calculate accuracy
accuracy = accuracy_score(y_t_resampled, y_predict_nn)
print(f"Accuracy: {accuracy*100:.2f}%")
```

```
Accuracy: 74.64%
```

Presion: The ratio of accurately predicted positive observations to all expected positives is known as precision. It measures how accurate positive forecasts are.

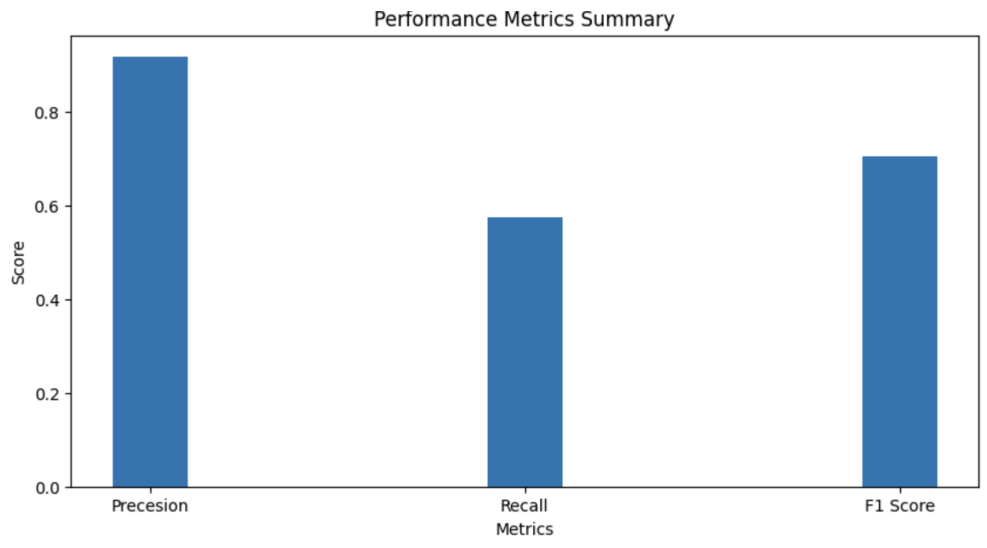
Recall: The ratio of accurately predicted positive observations to all observations made during the actual class is known as recall. It evaluates the classifier's capacity to identify every positive sample.

F1 Score: The precision and recall mean is the F1 score. It offers a balance between recall and precision.
F1 score reaches its best value at 1 and worst at 0.

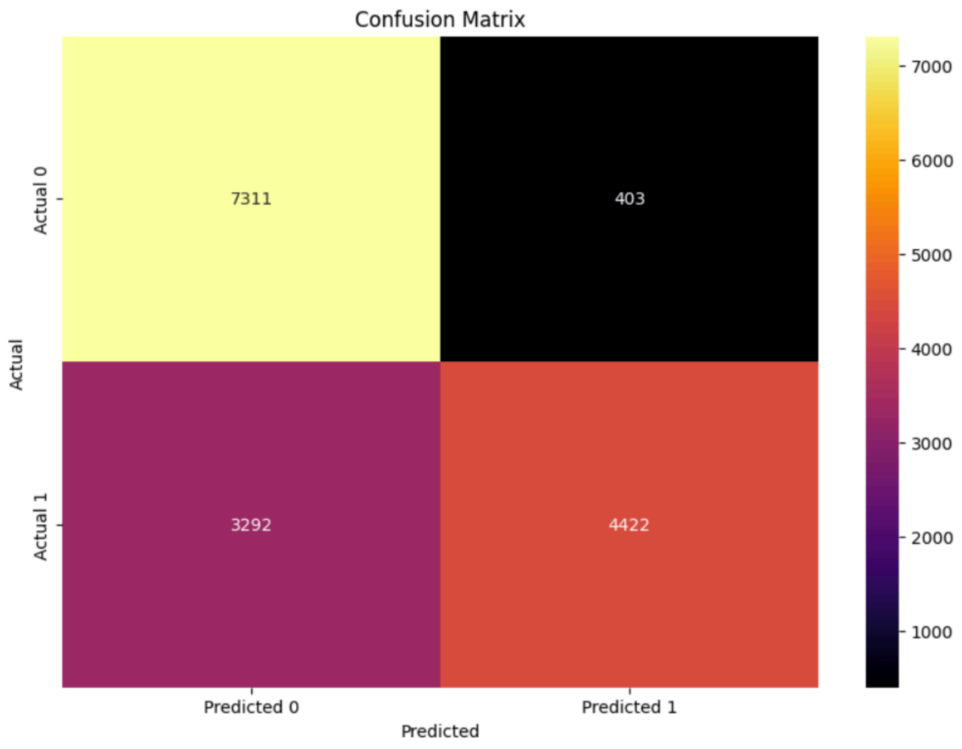
```
print(Cal_precesion_f1_recalll(y_t_resampled,y_predict_nn))
```

```
{'Precesion': 0.8152264057057556, 'Recall': 0.6371532278973295, 'F1 Score': 0.715273230008004}
```

Visualization :



Confusion matrix: Is a tabular representation of the counts of true positive, true negative, false positive, and false negative predictions that helps illustrate the effectiveness of the neural network model.



In the confusion matrix it can be seen that, when the actual value is 0, the model correctly predicted the value 0, 7311 times. when the actual value is 1, the model correctly predicted the value 1, 4422 times. When the value is 0, the model predicted 1, 403 times. When the value is 1, the model predicted 0, 3292 times.

8. Random Forest:

Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive performance and generalization. In the context of health insurance lead prediction, ensemble methods can help mitigate overfitting and increase model robustness.

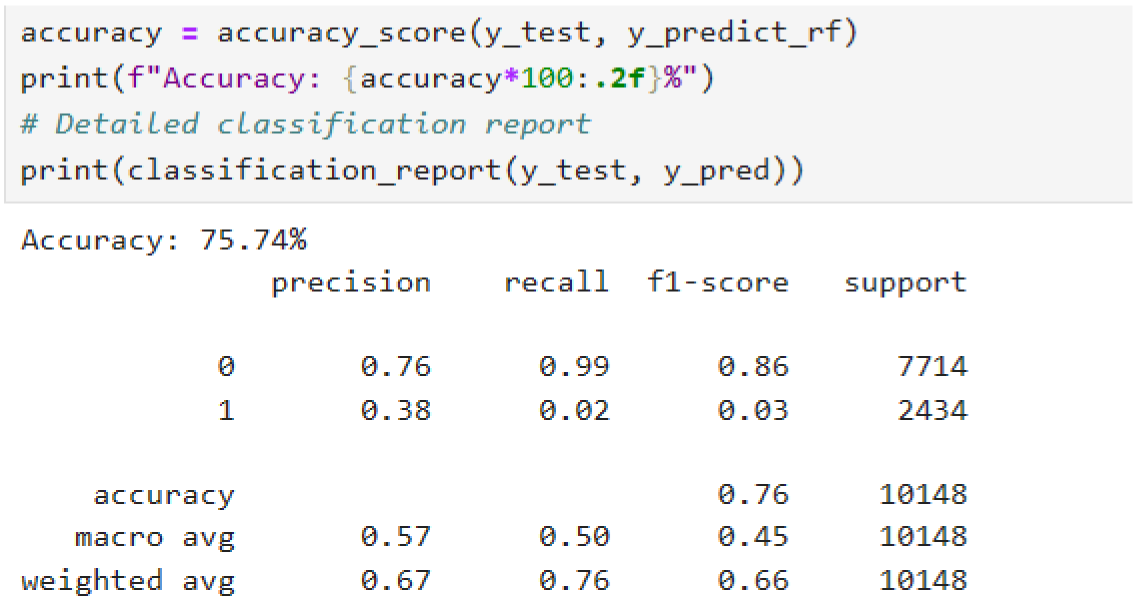
Random Forest can capture non-linear relationships and interaction effects between features, making it suitable for modeling complex decision-making processes in the health insurance sector.

By aggregating predictions from multiple decision trees, Random Forest reduces variance and improves predictive accuracy compared to individual decision trees.

To Train the Model:

Model Training and Evaluation: The main advantage of using the random forest classifier is that we do not need to resample the data as the random forest classifier has an inbuilt resampler which resamples the data to get the maximum performance. While initializing the random forest we have given several parameters to it . n_estimators parameter shows how many trees should be created while training, we have set this parameter to 500 , increasing this further can significantly increase the performance of the model , but it also increases load on the model.
Next we have set the criterion parameter to entropy which ensures that we are using the information gain metric in order to split each node of the decision tree while training.
Next we have set the max_depth parameter to 30 , this is a limit that we set on the depth of the tree.This feature prevents to model to overfit and give accurate results

Effectiveness of the Algorithm:
An accuracy of 74.74% is obtained by performing a decision tree algorithm.



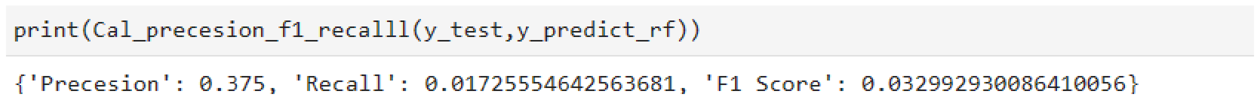
```
accuracy = accuracy_score(y_test, y_predict_rf)
print(f"Accuracy: {accuracy*100:.2f}%")
# Detailed classification report
print(classification_report(y_test, y_pred))
```

```
Accuracy: 75.74%
              precision    recall  f1-score   support

           0       0.76      0.99      0.86      7714
           1       0.38      0.02      0.03      2434

    accuracy                           0.76     10148
   macro avg       0.57      0.50      0.45     10148
weighted avg       0.67      0.76      0.66     10148
```

Presion: The ratio of accurately predicted positive observations to all expected positives is known as precision. It measures how accurate positive forecasts are.

Recall: The ratio of accurately predicted positive observations to all observations made during the actual class is known as recall. It evaluates the classifier's capacity to identify every positive sample.

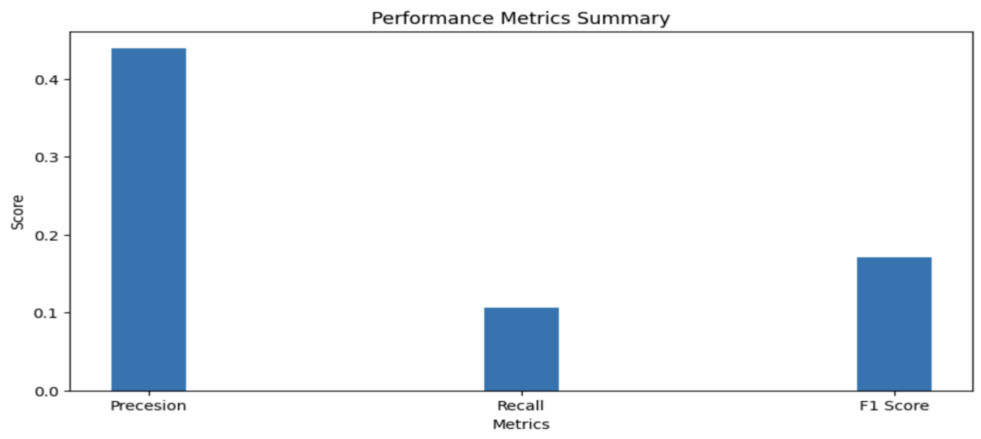
F1 Score: The precision and recall mean is the F1 score. It offers a balance between recall and precision.
F1 score reaches its best value at 1 and worst at 0.



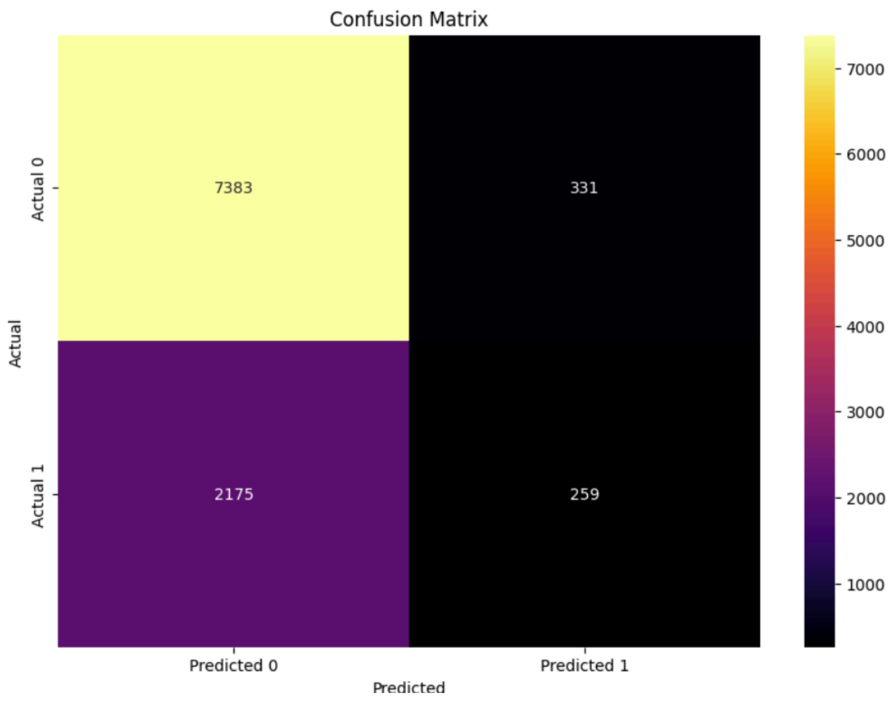
```
print(Cal_precesion_f1_recalll(y_test,y_predict_rf))
```

```
{'Precesion': 0.375, 'Recall': 0.01725554642563681, 'F1 Score': 0.032992930086410056}
```

Visualization :



Confusion matrix: Is a tabular representation of the counts of true positive, true negative, false positive, and false negative predictions that helps illustrate the effectiveness of the random forest model.



In the confusion matrix it can be seen that, when the actual value is 0, the model correctly predicted the value 0, 7383 times. when the actual value is 1, the model correctly predicted the value 1, 259 times. When the value is 0, the model predicted 1, 331 times. When the value is 1, the model predicted 0, 2175 times.

References:

Machine Learning Algorithms : https://scikit-learn.org/stable/

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
https://scikit-learn.org/stable/modules/naive_bayes.html
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
https://scikit-learn.org/stable/modules/tree.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
https://scikit-learn.org/stable/modules/neural_networks_supervised.html
https://imbalanced-learn.org/stable/

Matplotlib :
https://matplotlib.org/stable/plot_types/basic/bar.html#sphx-glr-plot-types-basic-bar-py
Pandas :
https://pandas.pydata.org/docs/user_guide/index.html#user-guide

https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html#pandas.get_dummies

Seaborn :
https://seaborn.pydata.org/generated/seaborn.heatmap.html

Decision Tree :
https://medium.com/@MrBam44/decision-trees-91f61a42c724

Multi Layer Perceptron :
https://medium.com/@jorgesleonel/multilayer-perceptron-6c5db6a8dfa3

Random Forest Classifier :
https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1

Peer Evaluation Form for Final Group Work

CSE 487/587B

- Please write the names of your group members.

Group member 1: Surya Venkata Rohit Moganti

Group member 2: Bhavani Kiran Kukunoor

- Rate each groupmate on a scale of 5 on the following points, with 5 being HIGHEST and 1 being LOWEST.

Evaluation Criteria	Group member 1	Group member 2
How effectively did your group mate work with you?	5	5
Contribution in writing the report	5	5
Demonstrates a cooperative and supportive attitude.	5	5
Contributes significantly to the success of the project .	5	5
TOTAL	20	20

Also please state the overall contribution of your teammate in percentage below, with total of all the three members accounting for 100% (33.33+33.33+33.33 ~ 100%):

Group member 1: 50 %

Group member 2: 50 %