❖ Prediction for different bill categories of the dataset using random forest classifier model.

Problem Statement :

Atlantis motor works is a large manufacturing firm engaged in making cars that use biologically derived carbon-hydrogen molecules to power the mobility of today. As a large multinational corporation, the financial team of Atlantis validates and checks thousands of bills and classifies them to the different bill categories according to the below information of the bill:

1. Company Name

2. Financial Department

3. Financial Account Group

4. Vendor Name

5. The Bill amount for the past 4 months.

The Bills data with the required attributes is provided and the Categories are provided for some bills in the last column, for some bills, Category is blank. This will form the required data set for this use case. Please refer to the "DataSet for the Data Science Test.xlsx" file for the same.

As a Data Scientist need to provide the solution approach using ML/Data Science techniques to provide the below output

1. Predict the label for the bills with blank categories.

2. Predict the label for the new set of bills which will be added in future.

Solution Approach :

The problem at hand can be solved using supervised machine learning techniques. Since the data contains labelled examples, we can train a classification model to predict the category of a bill based on its attributes. We can then use this model to predict the category of the bills with blank categories and also for new bills that will be added in the future.

The following are the steps that explains the solution approach :

1. **Data pre-processing :**

The first step is to pre-process the data by cleaning it, removing any unnecessary columns and transforming the categorical variables into numerical variables. This can be done using techniques such as one-hot encoding or label encoding.

2. **Splitting the data :**

Next, we need to split the data into a training set and a test set. The training set will be used to train the classification model and the test set will be used to evaluate its performance.

3. **Feature Selection:**

We need to identify which attributes are most important in predicting the category of a bill. We can use feature selection techniques such as correlation analysis or information gain to identify the most important features.

4. **Data Balancing:**

Check the distribution of the target variable to determine if the dataset is imbalanced

If the dataset is imbalanced, use techniques like oversampling or undersampling to balance the data.

5. **Choosing a classification algorithm :**

We need to choose a suitable classification algorithm based on the characteristics of the data and the performance of the algorithm on the training set. Some popular classification algorithms are:

- Logistic Regression
- Decision Trees
- Random Forests
- Support Vector Machines
- Naive Bayes

6. **Training the model :**

After selecting the classification algorithm, we can train the model using the training set.

7. Evaluating the model :

We need to evaluate the performance of the model on the test set to ensure that it is not overfitting or underfitting. We can use metrics such as accuracy, precision, recall, and F1-score to evaluate the performance of the model.

8. Hyperparameter tuning :

We can further improve the performance of the model by tuning its hyper parameters. This can be done using techniques such as grid search or random search.

9. Predicting categories for new bills :

Once the model has been trained and evaluated, we can use it to predict the categories for the bills with blank categories and for new bills that will be added in the future.

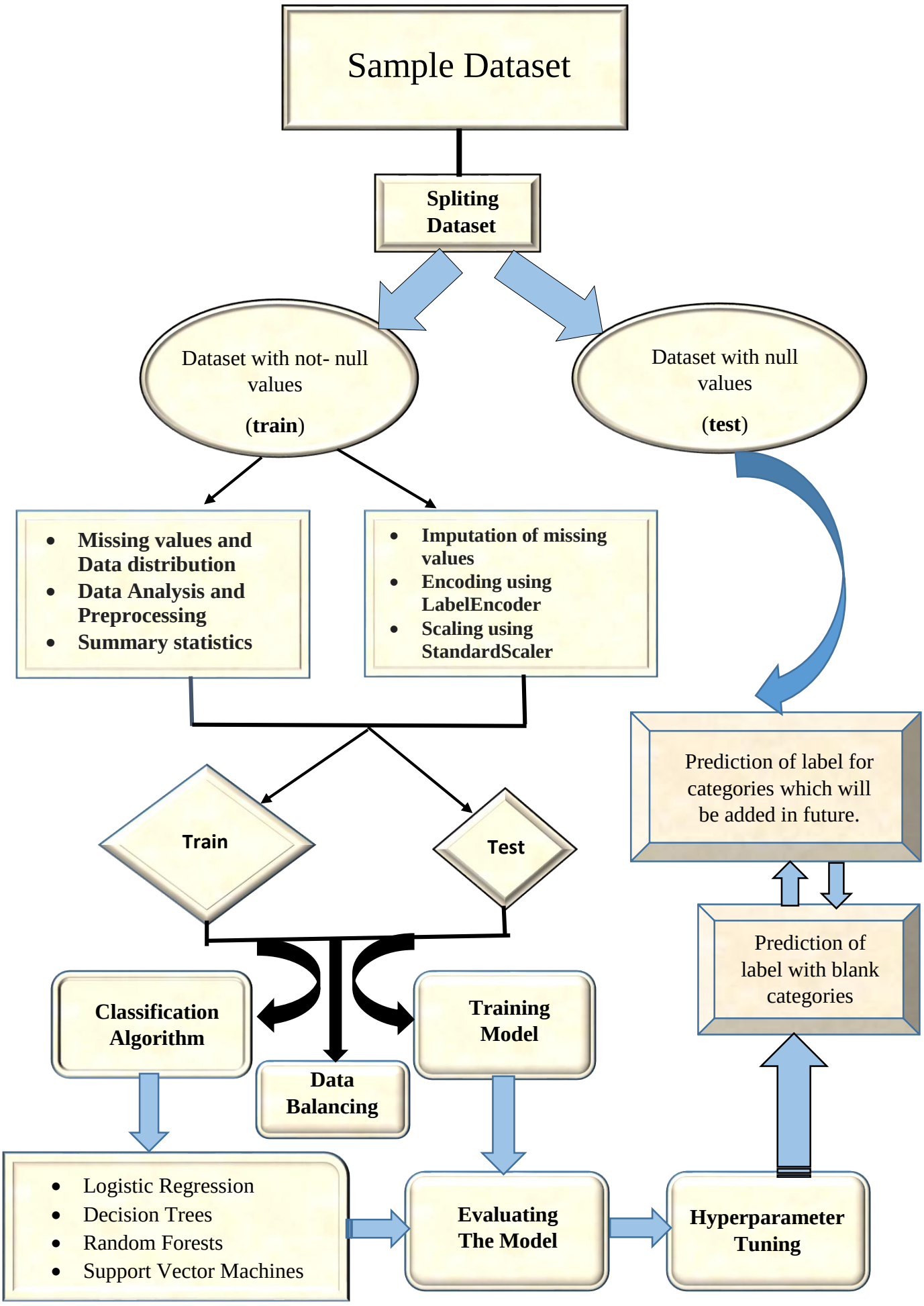
❖ **List of ML Algorithms:**

- Logistic Regression
- Decision Trees
- Random Forests
- Support Vector Machines
- Naive Bayes
- Gradient Boosting
- Neural Networks
- K-Nearest Neighbors (KNN)
- Principal Component Analysis (PCA)
- Lasso and Ridge Regression.

Note : The choice of algorithm will depend on the characteristics of the data and the problem at hand. We may need to try multiple algorithms and compare their performance before selecting the final algorithm.

The following is a flowchart that explains the solution approach :

❖ Mechanism :


Sample Dataset
Spliting Dataset
Dataset with not- null values
(train)
Dataset with null values
(test)
Missing values and Data distribution
Data Analysis and Preprocessing
Summary statistics
Imputation of missing values
Encoding using LabelEncoder
Scaling using StandardScaler
Prediction of label for categories which will be added in future.
Train
Test
Prediction of label with blank categories
Classification Algorithm
Data Balancing
Training Model
Logistic Regression
Decision Trees
Random Forests
Support Vector Machines
Evaluating The Model
Hyperparameter Tuning

❖ Model Used To Predict The Label For Categories:

Random Forest Classifier is a popular machine learning algorithm used for both classification and regression tasks. Here are some reasons why Random Forest Classifier is commonly used:

1. **High Accuracy :** Random Forest Classifier is known for its high accuracy and ability to handle large and complex datasets. It is an ensemble method that combines the results of multiple decision trees, which can reduce overfitting and increase the model's generalization performance.
2. **Robustness to outliers and missing values :** Random Forest Classifier can handle outliers and missing values in the input features. It does not require data normalization or standardization and can handle both categorical and continuous variables.
3. **Feature Importance :** Random Forest Classifier can provide feature importance measures, which can help in feature selection and identifying the most important variables for the classification task.
4. **Reducing bias and variance :** Random Forest Classifier can reduce the bias and variance of individual decision trees by randomly selecting subsets of features and samples to train each tree. This can help prevent overfitting and improve the model's accuracy and generalization performance.

Overall, Random Forest Classifier is a powerful and versatile machine learning algorithm that can be used for a wide range of classification tasks. Its ability to handle complex datasets, outliers, missing values, and provide feature importance measures make it a popular choice for many data scientists and machine learning practitioners.

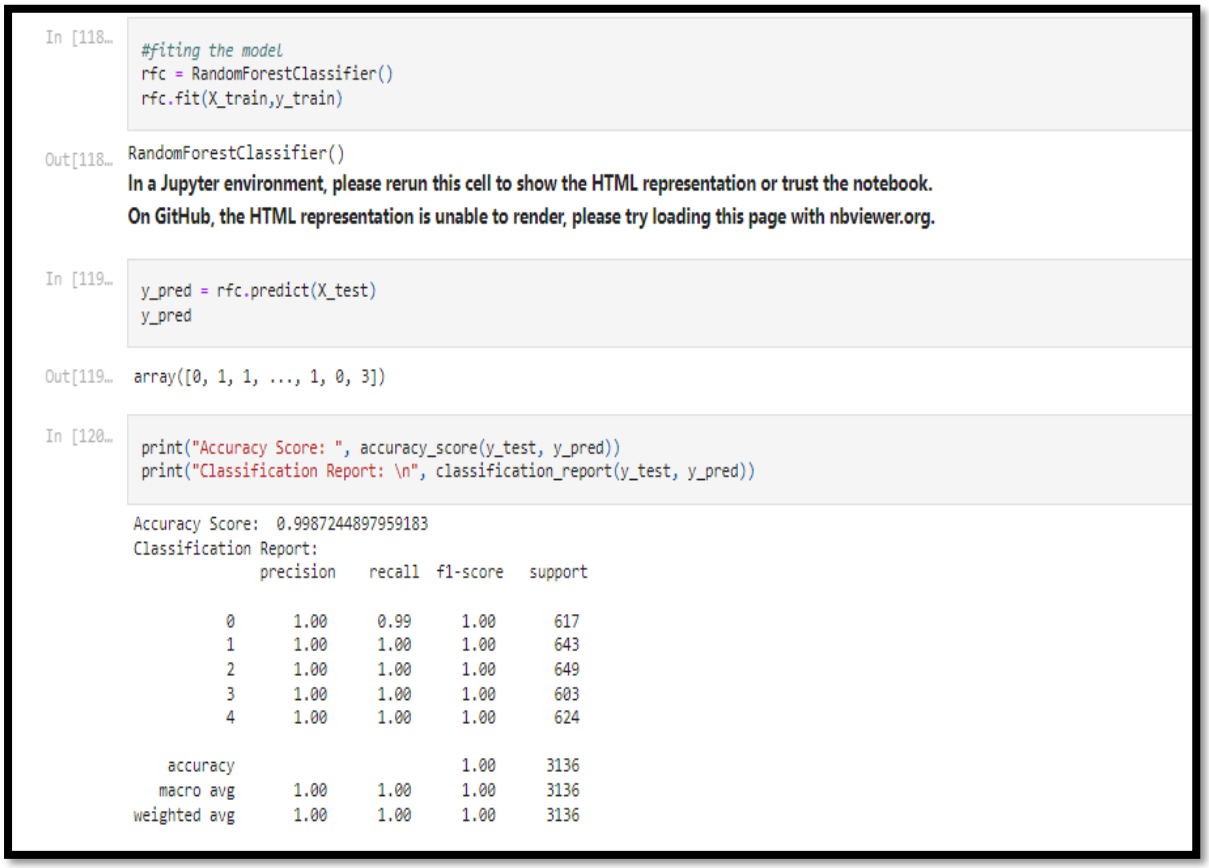
❖ There are several ways to measure the accuracy of machine learning models used in the solution approach. Here are a few common metrics:

1. **Accuracy Score :** This is the most straightforward metric and is calculated as the proportion of correctly classified samples to the total number of samples. It is suitable for balanced datasets but can be misleading in imbalanced datasets where the majority class dominates the classification.
2. **Precision and recall :** Precision and recall are two important metrics used in binary classification tasks. Precision measures the proportion of correctly predicted positive samples out of all predicted positive samples,

while recall measures the proportion of correctly predicted positive samples out of all true positive samples. These metrics can be useful in imbalanced datasets where the objective is to correctly identify the positive samples.

3. **F1 Score :** This is the harmonic mean of precision and recall, and is often used as a summary metric for binary classification tasks. It balances both precision and recall and is useful in imbalanced datasets where both precision and recall are important.

❖ Model Fitting Result :



In [118…
```
#fiting the model
rfc = RandomForestClassifier()
rfc.fit(X_train,y_train)
```

Out[118…
```
RandomForestClassifier()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [119…
```
y_pred = rfc.predict(X_test)
y_pred
```

Out[119…
```
array([0, 1, 1, ..., 1, 0, 3])
```

In [120…
```
print("Accuracy Score: ", accuracy_score(y_test, y_pred))
print("Classification Report: \n", classification_report(y_test, y_pred))
```

```
Accuracy Score:  0.9987244897959183
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       617
           1       1.00      1.00      1.00       643
           2       1.00      1.00      1.00       649
           3       1.00      1.00      1.00       603
           4       1.00      1.00      1.00       624

    accuracy                           1.00      3136
   macro avg       1.00      1.00      1.00      3136
weighted avg       1.00      1.00      1.00      3136
```

```
param_grid = {'n_estimators': [100, 200, 300], 'max_depth': [5, 10, 15, 20], 'min_samples_split': [2, 5, 10]}
grid_search = GridSearchCV(rfc, param_grid, cv=5)
grid_search.fit(x_rs, y_rs)
print(grid_search)

# Print the best hyperparameters and the best score
print("Best Hyperparameters: ", grid_search.best_params_)
print("Best Score: ", grid_search.best_score_)
```

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [5, 10, 15, 20],
                         'min_samples_split': [2, 5, 10],
                         'n_estimators': [100, 200, 300]})
Best Hyperparameters:  {'max_depth': 15, 'min_samples_split': 5, 'n_estimators': 200}
Best Score:  0.9737882653061225
```

In [122…

```
# Evaluate the model's performance on the testing dataset
parameters = {'max_depth': 15, 'min_samples_split': 10, 'n_estimators': 100}

rfc1 = RandomForestClassifier(**parameters)
rfc1.fit(X_train,y_train)
y_pred = rfc1.predict(X_test)
# y_pred = grid_search.predict(X_test)
print("Accuracy Score: ", accuracy_score(y_test, y_pred))
print("Classification Report: \n", classification_report(y_test, y_pred))
```

```
Accuracy Score:  0.9990433673469388
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       617
           1       1.00      1.00      1.00       643
           2       1.00      1.00      1.00       649
           3       1.00      1.00      1.00       603
           4       1.00      1.00      1.00       624

    accuracy                           1.00      3136
   macro avg       1.00      1.00      1.00      3136
weighted avg       1.00      1.00      1.00      3136
```

So, Accuracy score for the random forest classifier model is 0.99. Therefore, Random Forest Classifier is best fitted model to predict the label for blank and new set of bills for categories.

Code Of Implementation :

1. Predict the label for the bills with blank categories
2. Predict the label for the new set of bills which will be added in future

Github Link :

https://github.com/mayurpawar24/Prediction-For-Different-Bill-Categories.

❖ Limitations/Assumptions :

There are some limitations and assumptions of the solution approach for predicting the category of bills using machine learning techniques:

Data Quality : The quality of the input data can have a significant impact on the performance of the machine learning algorithm. If the data is noisy or contains errors, it can affect the accuracy of the predictions.

Data Bias : The input dataset may contain bias, which can affect the performance of the model. For example, if the dataset contains more data from one category than others, it can lead to bias in the model.

Feature Selection : The performance of the machine learning algorithm can depend heavily on the quality of the features selected. If the features are not relevant or informative, the model may not be able to make accurate predictions.

Model Selection : The performance of the model depends on the choice of machine learning algorithm and its hyperparameters. Choosing the wrong algorithm or hyperparameters can lead to poor performance.

Assumptions : The solution approach assumes that the input data is representative of the real-world scenario and that the relationship between the input features and output labels is consistent over time.

Generalizability : The solution approach assumes that the trained model will generalize well to new, unseen data. However, this may not always be the case, especially if the new data is significantly different from the training data.

Overall, while machine learning techniques can be powerful tools for predicting the category of bills, they are not without limitations and assumptions. It is important to consider these limitations and assumptions when designing and implementing a machine learning solution.

❖ Conclusion :

The problem of categorizing bills can be solved using supervised machine learning techniques, including data pre-processing, feature selection, data balancing, model selection, training and evaluation, hyperparameter tuning, and prediction. By following these steps, we can accurately predict the category of bills with blank categories and future bills, improving the efficiency of the financial team at Atlantis motor works.

Overall, this solution approach can provide a robust and efficient way to deal with a large number of bills with missing or incomplete information.