**Name: Usama Husnain**

**Customer Churn Project**

**Information About Dataset:**

In this task, our main goal is to analysis and predict customer churn for a telecommunication company and to develop a model that accurately predicts the customer churn. In the Dataset, I’ve 21 columns with 18 Categorical columns and 3 Numerical Columns. If you want to download the Dataset here is the Link(**(**<https://www.kaggle.com/datasets/blastchar/telco-customer-churn>).

**EDA (Exploratory Data Analysis):**

In this part, I’ve found many relationships with the help of seaborn and matplotlib modules.

A graph with numbers and bars

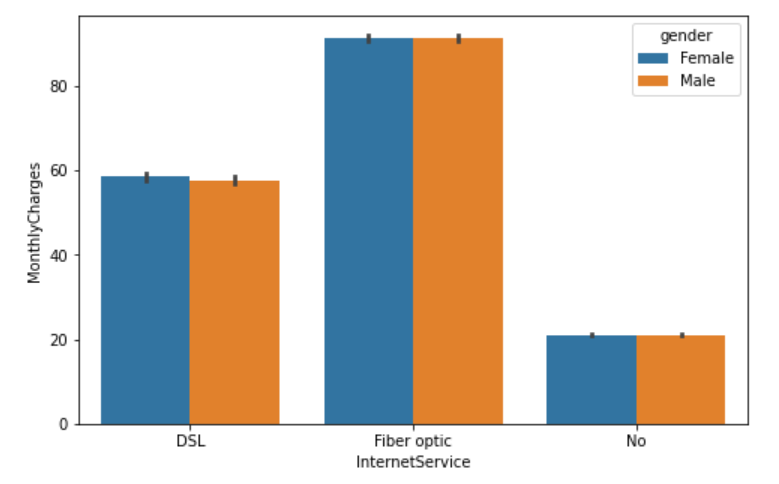
Description automatically generated

In this Figure, I’ve found the relationship between the payment method and Churn.

A graph of different colored rectangular shapes

Description automatically generated

In this Figure, I’ve found the relationship between the phone-Service and Total-Charges with Gender.



In this Figure, I’ve found the relationship between Internet-Service and Monthly-Charges with Gender.

A graph of different colored bars

Description automatically generated

In this Figure, I’ve found the relationship between the Fiber-Optic and Total-Charges with Contract.

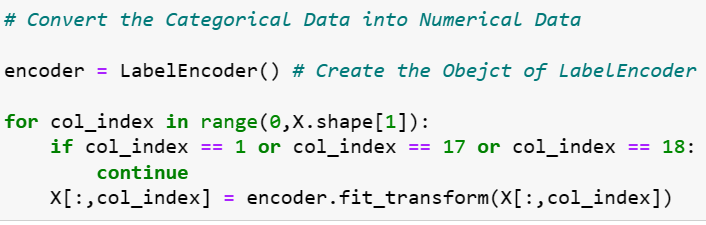
A graph with blue and orange bars

Description automatically generated

In this Figure, I’ve found the relationship between the Senior-Citizen and Payment-Method with Churn.

**Dataset Pre-processing:**

In the Processing, I have handled many problems like missing values, Handle the Categorical Data and Normalize the Data. For the Categorical Data, I’ve used Label-Encoder Method and For Normalize the Data, I’ve used the Min-Max-Scaler Method.



In this Code, I’ve Implemented the Method that handles the Categorical Data.

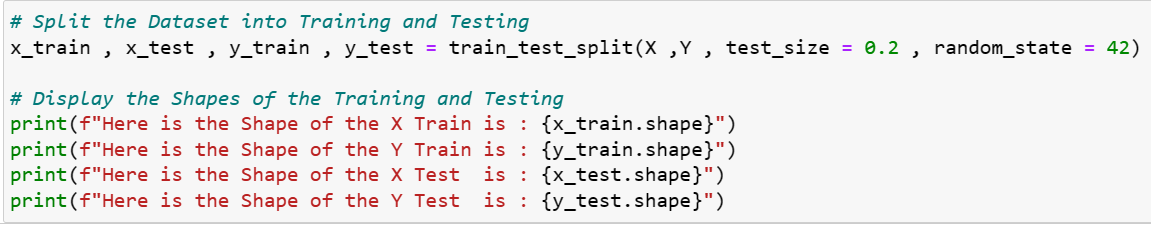
A white background with blue text

Description automatically generated

In this Code, I’ve Implemented the Method that Normalize the Data into [0,1].

**Machine Learning Models with Imbalance Dataset**

In this Part, Firstly, I’ve splitting the dataset into training and testing with the help of train\_trest\_split function. Secondly, I’ve implemented the four Machine Learning Models (Logistic Regression, Decision Tree Classifier, Random Forest Classifier and Support Vector Classifier). Thirdly, I’ve trained these models in the training dataset. After trained the model I’ve predicted the results in the Test dataset.



In this Code, I’ve split the dataset into training and testing.

A screenshot of a computer program

Description automatically generated

In this Code, I ‘ve defined the function that finds out all results of each model.

**Logistic Regression Results:**

A close-up of a logistic report

Description automatically generated

In this Code, I’ve calculated the results for Logistic Regression with the help of model\_res() function.

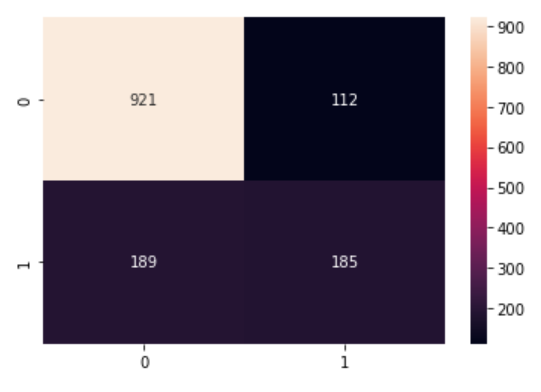


In this Output, I’ve displayed the Accuracy of the Logistic Regression Model.

A screenshot of a graph

Description automatically generated

In this Output, I’ve displayed the Classification Report of the Logistic Regression Model.



In this output, I’ve displayed the Confusion Matrix for the Logistic Regression Model.

**Decision Tree Classifier:**

A close-up of a computer screen

Description automatically generated

In this Code, I’ve Calculated the Results for the Decision Tree Classifier with the help of model\_res() function.



In this Output, I’ve displayed the Accuracy of the Decision Tree Classifier Model.

A number of numbers on a white background

Description automatically generated

In this Output, I’ve displayed the Classification Report of the Decision Tree Classifier Model.

A black and white squares with white text

Description automatically generated

In this Output, I’ve displayed the Confusion Matrix of the Decision Tree Classifier Model.

**Random Forest Classifier:**

A close-up of a computer screen

Description automatically generated

In this Output, I’ve Calculated the Results of Random Forest Classifier with the help of model\_res() function.



In this Output, I’ve displayed the Accuracy of the Random Forest Classifier Model.

A screenshot of a graph

Description automatically generated

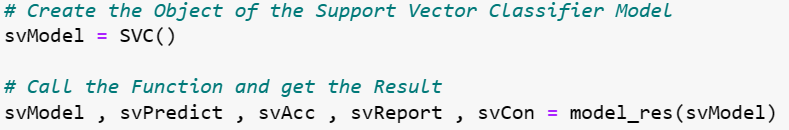
In this Output, I’ve displayed the classification report of the Random Forest Classifier Model

A graph of numbers and a number

Description automatically generated

In this Output, I’ve displayed the Confusion Matrix of the Random Forest Classifier Model.

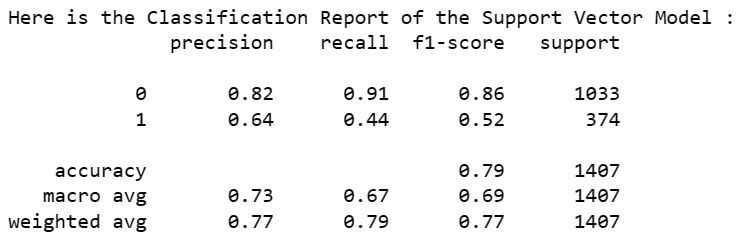
**Support Vector Classifier:**



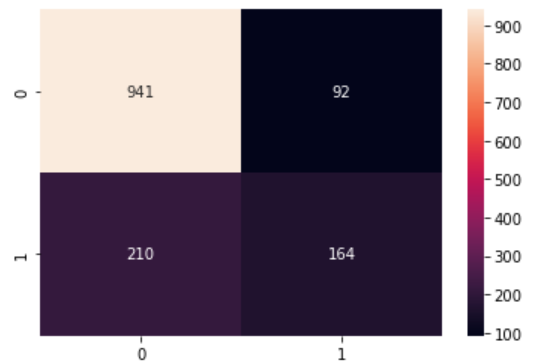
In this Code, I’ve calculated the results of Support Vector Classifier with the help of model\_res() function.



In this Output, I’ve displayed the Accuracy of the Support Vector Classifier Model.



In this Output, I’ve displayed the Classification Report of the Support Vector Classifier Model.



In this Output, I’ve Displayed the Confusion Matrix of the Support Vector Classifier Model.

**Model Evaluation:**

A graph showing different colored rectangular shapes

Description automatically generated

In this Plot, I’ve displayed the Accuracy of each Model with Imbalanced Dataset.

**Machine Learning Models with Balance Dataset & KFOLD Method**

In this Part, Firstly, I’ve splitting the dataset into training and testing with the help of KFold function. Secondly, I’ve handled the Imbalanced Dataset with the help of SMOTE Method and implemented the four Machine Learning Models (Logistic Regression, Decision Tree Classifier, Random Forest Classifier and Support Vector Classifier) too. Thirdly, I’ve trained these models in the training dataset. After trained the model I’ve predicted the results in the Test dataset.

A close-up of a white background

Description automatically generated

In this Code, I’ve handled the Imbalance Dataset with the help of SMOTE Technique.

A white background with black and blue text

Description automatically generated

In this Code, I’ve defined the Function for the Cross Validation and the Calculate the Model Results.

**Logistic Regression Model:**

A close-up of text

Description automatically generated

In this Code, I’ve Calculate the Results for the Logistic Regression Model with Balanced Dataset.



In this Output, I’ve displayed the Accuracy Score of the Logistic Regression Model.

**Decision Tree Classifier Model:**

A white background with black and blue text

Description automatically generated

In this Code, I’ve Calculate the Results for the Decision Tree Model with Balanced Dataset.



In this Output, I’ve displayed the Accuracy Score of the Decision Tree Model.

**Random Forest Classifier Model:**

A white background with black and blue text

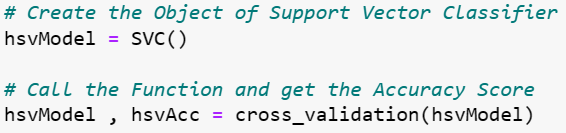
Description automatically generateds

In this Code, I’ve calculated the Results for the Random Forest Classifier Model with Balanced Dataset.



In this Output, I’ve displayed the Accuracy Score for the Random Forest Classifier Model.

**Support Vector Classifier Model:**



In this Code, I’ve Calculated the Results for the Support Vector Classifier Model with Balanced Dataset.



In this Output, I’ve displayed the Accuracy of the Support Vector Classifier Model.

**Model Evaluation:**

A graph of different colored rectangular shapes

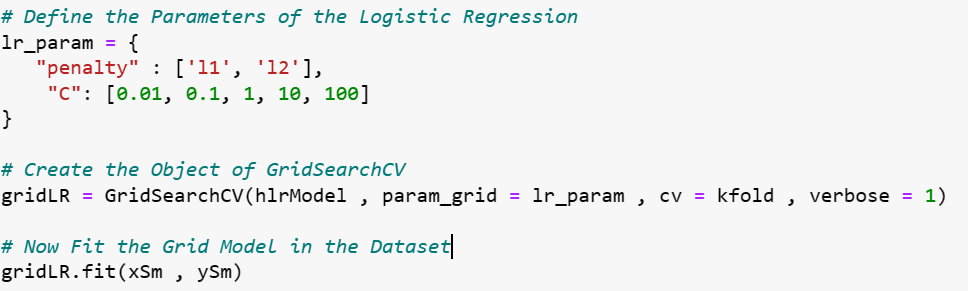
Description automatically generated

In this Plot, I’ve displayed the Accuracy of each Model with Imbalanced Dataset and KFold Method.

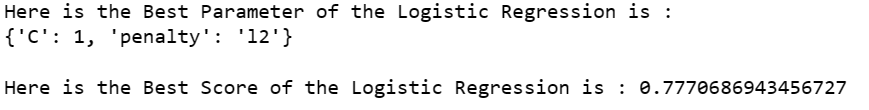
**Optimization Machine Learning Models with Balance Dataset & KFOLD Method**

In this Part, I’ve optimized the Previous Four Machine Learning Models and got the Results.

**Logistic Regression Model:**

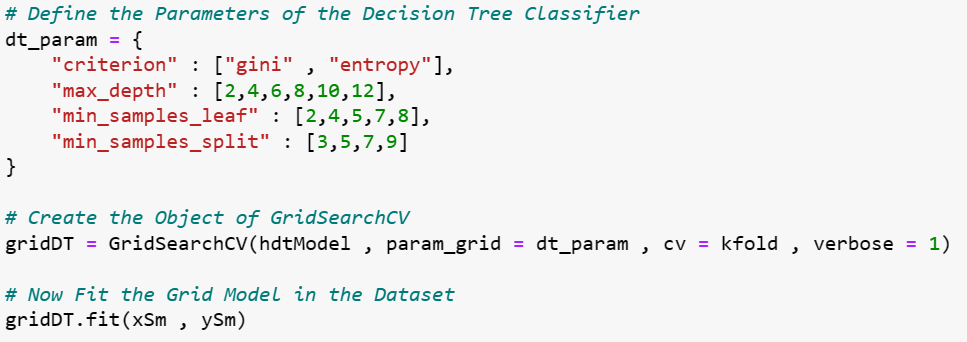


In this Code, Firstly, I’ve defined the Parameters for the Logistic Regression after that I’ve applied the GridSearchCV for training and testing the Model after that I’ve fit the GridSearchCV in the Dataset.

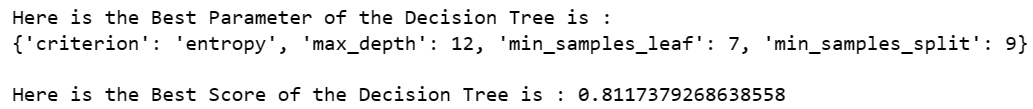


In this Output, I’ve displayed the Best Parameter of Logistic Regression Model and displayed the Accuracy Score too.

**Decision Tree Classifier Model:**

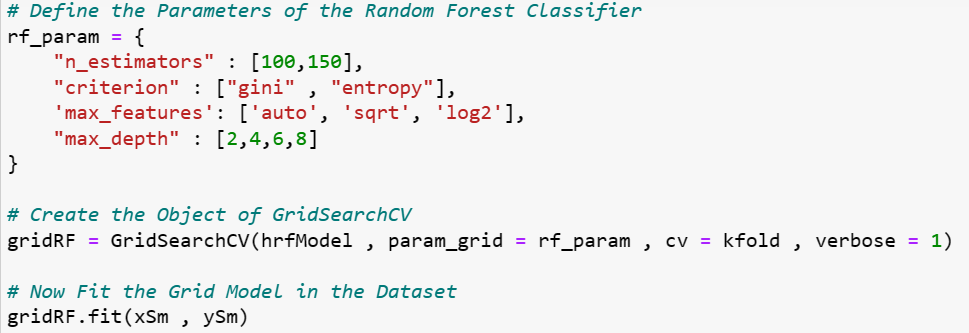


In this Code, Firstly, I’ve defined the Parameters for the Decision Tree Classifier after that I’ve applied the GridSearchCV for training and testing the Model after that I’ve fit the GridSearchCV in the Dataset.

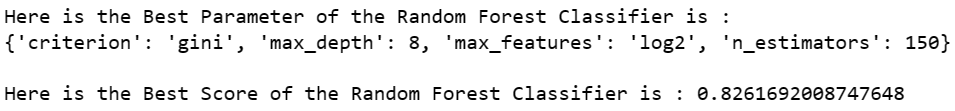


In this Output, I’ve displayed the Best Parameter of Decision Tree Classifier and displayed the Accuracy Score too.

**Random Forest Classifier Model:**

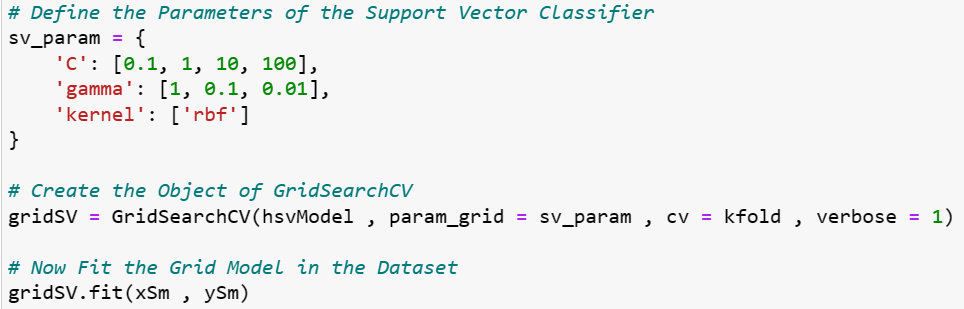


In this Code, Firstly, I’ve defined the Parameters for the Random Forest Classifier after that I’ve applied the GridSearchCV for training and testing the Model after that I’ve fit the GridSearchCV in the Dataset.

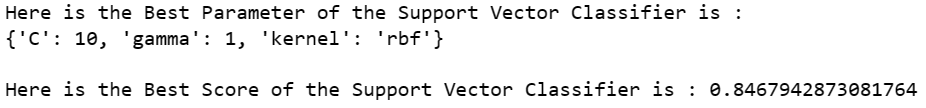


In this Output, I’ve displayed the Best Parameter of Random Forest Classifier and displayed the Accuracy Score too.

**Support Vector Classifier Model:**

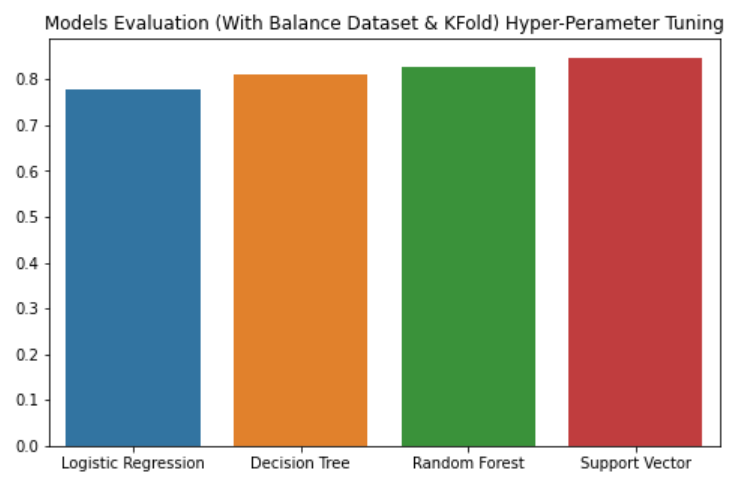


In this Code, Firstly, I’ve defined the Parameters for the Support Vector Classifier after that I’ve applied the GridSearchCV for training and testing the Model after that I’ve fit the GridSearchCV in the Dataset.



In this Output, I’ve displayed the Best Parameter of Support Vector Classifier and displayed the Accuracy Score too.

**Model Evaluation:**



In this Plot, I’ve displayed the Accuracy of each Model with balanced Dataset and KFold Method and Optimization Model.