SACRAMENTO STATE

CSC 177- Data Analytics and Mining

Project 3: Classification Project

Team Challengers (23):

1. Srujay Reddy Vangoor
2. Vaibhav Jain
3. Bashar Allwza
4. Varun Bailapudi
5. Uddayankith Chodagam

Classification

What is Classification?

Classification is a technique used in data mining and machine learning to predict the class label of a data point based on its features or attributes. The goal of classification is to build a model that can learn from existing labeled data and accurately predict the class labels of new, unlabeled data.

There are various algorithms that can be used for classification, such as decision trees, logistic regression, k-nearest neighbor, support vector machines, and neural networks. The choice of algorithm depends on the nature of the data and the problem at hand.

About the Dataset

Churn Rate is the percentage of subscribers to a service who discontinue their subscriptions to the service within a given time period. For a company to expand its clientele, its growth rate, as measured by the number of new customers, must exceed its churn rate.

This is a Classification Problem in which you'll classify a customer based on his/her Credit Score, Region, Gender, Age, Tenure, Balance, Salary etc. whether he/she will EXIT(1) or NOT(0).

Data Preprocessing

Data preprocessing is an essential step before running clustering models because it can significantly impact the accuracy and effectiveness of the clustering results.

```
In [13]: df.isna().any()

Out[13]: RowNumber          False
         CustomerId         False
         Surname            False
         CreditScore        False
         Geography          False
         Gender             False
         Age                False
         Tenure             False
         Balance            False
         NumOfProducts      False
         HasCrCard          False
         IsActiveMember     False
         EstimatedSalary    False
         Exited             False
         dtype: bool
```

```
In [14]: df.nunique()

Out[14]: RowNumber          10000
         CustomerId         10000
         Surname             2932
         CreditScore          460
         Geography              3
         Gender                 2
         Age                   70
         Tenure                11
         Balance             6382
         NumOfProducts          4
         HasCrCard              2
         IsActiveMember         2
         EstimatedSalary     9999
         Exited                 2
         dtype: int64
```

Feature Selection

Feature selection is a technique used in classification to identify the most relevant features (or attributes) that can contribute to the accuracy of the model. It involves selecting a subset of the original features from the dataset that are most relevant to the classification task, while discarding the irrelevant or redundant ones.

The primary aim of feature selection is to improve the performance and efficiency of the classification model by reducing the dimensionality of the input space. This can help to reduce the risk of overfitting, speed up the training process, and simplify the interpretation of the model.

In [22]:
```
df["NewAGT"] = df["Age"] - df["Tenure"]
df["CreditsScore"] = pd.qcut(df['CreditScore'], 10, labels = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
df["AgeScore"] = pd.qcut(df['Age'], 8, labels = [1, 2, 3, 4, 5, 6, 7, 8])
df["BalanceScore"] = pd.qcut(df['Balance'].rank(method="first"), 10, labels = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
df["EstSalaryScore"] = pd.qcut(df['EstimatedSalary'], 10, labels = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
df["NewEstimatedSalary"] = df["EstimatedSalary"] / 12
```

In [23]:
```
df.head()
```

Out[23]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	...	IsActiveMember	EstimatedSalary	Exite
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	...	1	101348.88	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	...	1	112542.58	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	...	0	113931.57	
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	...	0	93826.63	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	...	1	79084.10	

5 rows × 21 columns

In [24]:
```
df = pd.get_dummies(df, columns =["Geography", "Gender"], drop_first = True)
```

K-Nearest Neighbors

K-Nearest Neighbors

The k-nearest neighbors (KNN) algorithm is a data classification method for estimating the likelihood that a data point will become a member of one group or another based on what group the data points nearest to it belong to.

The k-nearest neighbor algorithm is a type of supervised machine learning algorithm used to solve classification and regression problems. However, it's mainly used for classification problems.

Results

```
KNeighborsClassifier(n_neighbors=17)
[[1195  378]
 [ 358 1215]]
Accuracy on test data is 0.77
F1 score on test data is 0.76
Precision Score on test data is 0.77
Recall score on test data is 0.76
              precision    recall  f1-score   support

           0       0.77      0.76      0.76      1573
           1       0.76      0.77      0.77      1573

    accuracy                           0.77      3146
   macro avg       0.77      0.77      0.77      3146
weighted avg       0.77      0.77      0.77      3146
```

Naive Bayes Classifier

Naive Bayes Classifier

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Naive Bayes Classifier (Contd.)

Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

Gaussian Naive Bayes

When working with continuous data, an assumption often taken is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. The likelihood of the features is assumed to be

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Results

```
GaussianNB()
[[1131  442]
 [ 343 1230]]
Accuracy on test data is 0.75
F1 score on test data is 0.74
Precision Score on test data is 0.77
Recall score on test data is 0.72
              precision    recall  f1-score   support

           0       0.77      0.72      0.74      1573
           1       0.74      0.78      0.76      1573

    accuracy                           0.75      3146
   macro avg       0.75      0.75      0.75      3146
weighted avg       0.75      0.75      0.75      3146
```

Support Vector Machine

Support Vector Machine

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression.

The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane.

Results

```
SVC(gamma='auto', random_state=12345)
[[1290  283]
 [ 355 1218]]
Accuracy on test data is 0.80
F1 score on test data is 0.80
Precision Score on test data is 0.78
Recall score on test data is 0.82
              precision    recall  f1-score   support

           0       0.78      0.82      0.80      1573
           1       0.81      0.77      0.79      1573

    accuracy                           0.80      3146
   macro avg       0.80      0.80      0.80      3146
weighted avg       0.80      0.80      0.80      3146
```

Decision Tree Classifier

Decision Tree Classifier

A decision tree classifier is a type of supervised machine learning algorithm used for classification tasks. It works by creating a tree-like model of decisions and their possible consequences. The tree consists of nodes that represent a feature or attribute of the data, branches that represent the decision rules based on that feature, and leaves that represent the outcomes or class labels.

During training, the algorithm determines which features are most informative in making decisions and creates a tree structure that optimizes classification accuracy. In the testing phase, the model can be used to predict the class label of a new data point by following the decision rules in the tree.

Results

```
DecisionTreeClassifier(random_state=12345)
[[1350  223]
 [ 418 1155]]
Accuracy on test data is 0.80
F1 score on test data is 0.81
Precision Score on test data is 0.76
Recall score on test data is 0.86
              precision    recall  f1-score   support

           0       0.76      0.86      0.81      1573
           1       0.84      0.73      0.78      1573

    accuracy                           0.80      3146
   macro avg       0.80      0.80      0.80      3146
weighted avg       0.80      0.80      0.80      3146
```

Logistic Regression

Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

Results

```
LogisticRegression(random_state=12345)
[[1114  459]
 [ 364 1209]]
Accuracy on test data is 0.74
F1 score on test data is 0.73
Precision Score on test data is 0.75
Recall score on test data is 0.71
              precision    recall  f1-score   support

           0       0.75      0.71      0.73      1573
           1       0.72      0.77      0.75      1573

    accuracy                           0.74      3146
   macro avg       0.74      0.74      0.74      3146
weighted avg       0.74      0.74      0.74      3146
```

Thank you!!