Lab 1: Data Pre-processing and Handling Missing Values

Problem Statement:

You are working with a dataset of customer details for a retail store, which includes information about their age, gender, income, and purchase history. However, some of the values in the dataset are missing or incorrect. The goal is to clean and pre-process the data to make it suitable for machine learning tasks.

Dataset:

The dataset contains the following columns:

- **CustomerID** (Categorical)
- **Age** (Numerical, with some missing values)
- **Gender** (Categorical)
- **AnnualIncome** (Numerical, with some extreme values)
- **PurchaseAmount** (Numerical, with some missing values)

Sample Data:

CustomerID	Age	Gender	AnnualIncome	PurchaseAmount
1	25	Male	50000	250
2	NaN	Female	75000	NaN
3	45	NaN	120000	500
4	22	Female	50000	100
5	NaN	Male	68000	NaN

Objective:

1. Drop columns with more than **60% missing values** (if there is any).
2. Impute missing values in **numerical columns** using the **mean or median**.
3. Impute missing values in **categorical columns** using the **most frequent value (mode)**.
4. Provide a **pre-processed dataset**.

Importing Necessary Libraries

Making DataFrame

	CustomerID	Age	Gender	AnnualIncome	PurchaseAmount
0	1	25.0	Male	50000	250.0
1	2	NaN	Female	75000	NaN
2	3	45.0	None	120000	500.0
3	4	22.0	Female	50000	100.0
4	5	NaN	Male	68000	NaN

Objective 1:

Objective 2:

	CustomerID	Age	Gender	AnnualIncome	PurchaseAmount
0	1	25.000000	Male	50000	250.000000
1	2	30.666667	Female	75000	283.333333
2	3	45.000000	None	120000	500.000000
3	4	22.000000	Female	50000	100.000000
4	5	30.666667	Male	68000	283.333333

Objective 3:

	CustomerID	Age	Gender	AnnualIncome	PurchaseAmount
0	1	25.000000	Male	50000	250.000000
1	2	30.666667	Female	75000	283.333333
2	3	45.000000	Female	120000	500.000000
3	4	22.000000	Female	50000	100.000000
4	5	30.666667	Male	68000	283.333333

Objective 4:

Label Encoding

Scaling

	CustomerID	Age	Gender	AnnualIncome	PurchaseAmount
0	0.00	0.130435	1	0.000000	0.375000
1	0.25	0.376812	0	0.357143	0.458333
2	0.50	1.000000	0	1.000000	1.000000
3	0.75	0.000000	0	0.000000	0.000000
4	1.00	0.376812	1	0.257143	0.458333

Lab 2: Feature Scaling and Normalization

Problem Statement:

You are working with a dataset related to house prices in different neighborhoods. The dataset contains a wide range of numerical variables such as the number of bedrooms, area in square feet, and the price of the house. You need to scale the data for machine learning purposes, as different features have different ranges.

Dataset:

The dataset contains the following columns: - **HouseID** (Categorical) - **Bedrooms** (Numerical) - **AreaSqFt** (Numerical) - **Price** (Numerical)

Sample Data:

HouseID	Bedrooms	AreaSqFt	Price
1	3	1500	300000
2	2	800	150000
3	4	2500	500000
4	3	1800	350000
5	1	600	120000

Objective:

1. Apply **Min-Max Scaling** to normalize the values of **AreaSqFt** and **Price**.
2. Apply **Standardization (Z-score)** on the **Bedrooms** column.
3. Analyse the impact of scaling on the dataset (Using **Matplotlib**).

Importing Necessary Libraries

Making DataFrame

	HouseID	Bedrooms	AreaSqFt	Price
0	1	3	1500	300000
1	2	2	800	150000
2	3	4	2500	500000
3	4	3	1800	350000
4	5	1	600	120000

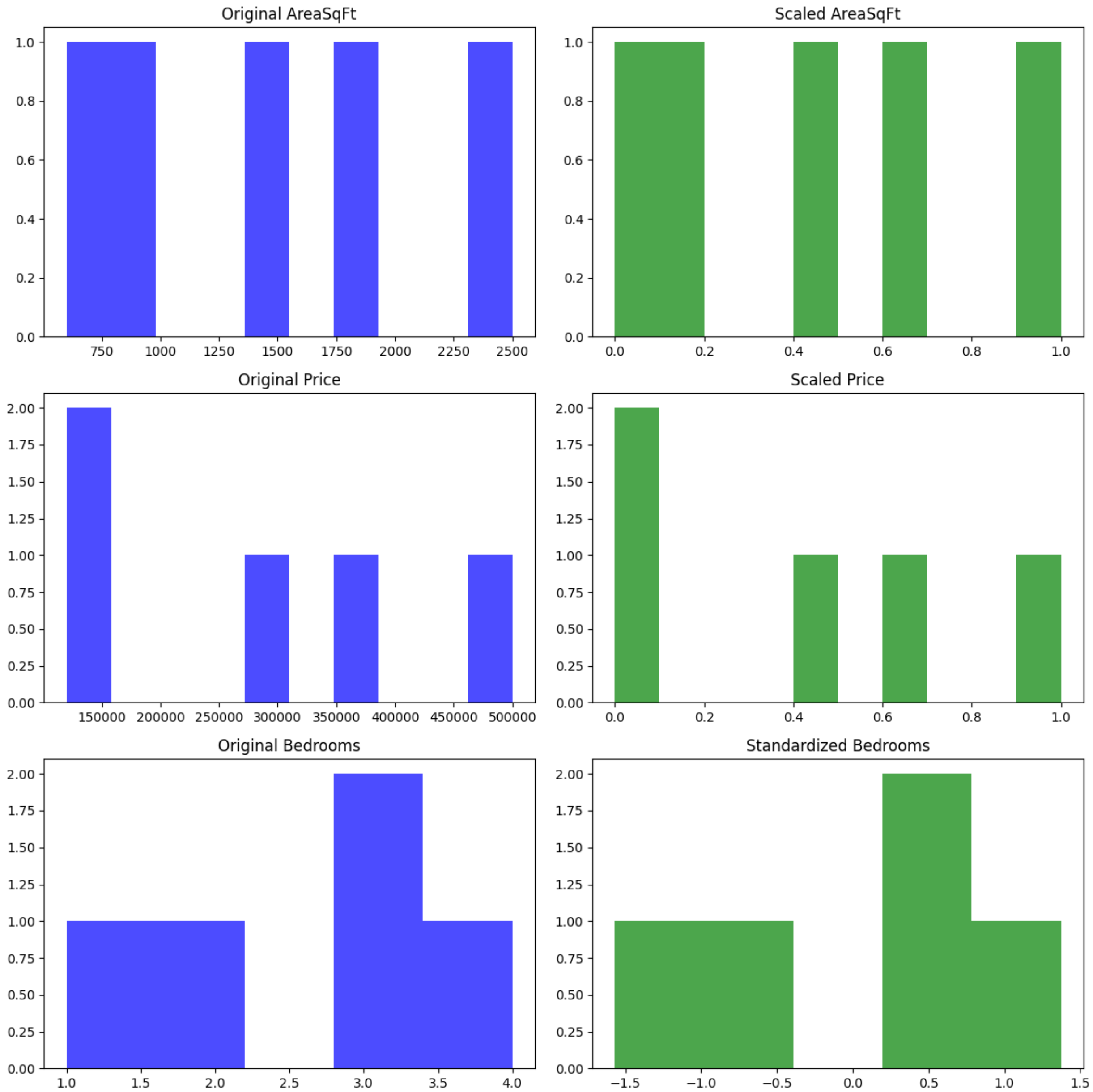
Objective 1

	HouseID	Bedrooms	AreaSqFt	Price
0	1	3	0.473684	0.473684
1	2	2	0.105263	0.078947
2	3	4	1.000000	1.000000
3	4	3	0.631579	0.605263
4	5	1	0.000000	0.000000

Objective 2

	HouseID	Bedrooms	AreaSqFt	Price
0	1	0.392232	0.473684	0.473684
1	2	-0.588348	0.105263	0.078947
2	3	1.372813	1.000000	1.000000
3	4	0.392232	0.631579	0.605263
4	5	-1.568929	0.000000	0.000000

Objective 3


Original AreaSqFt
1.0
0.8
0.6
0.4
0.2
0.0
750 1000 1250 1500 1750 2000 2250 2500
Scaled AreaSqFt
1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.2 0.4 0.6 0.8 1.0
Original Price
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
150000 200000 250000 300000 350000 400000 450000 500000
Scaled Price
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0 0.2 0.4 0.6 0.8 1.0
Original Bedrooms
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1.0 1.5 2.0 2.5 3.0 3.5 4.0
Standardized Bedrooms
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5

Lab 3: Confusion Matrix for Classification Problem

Problem Statement:

You are given a dataset of student exam results, where students have been classified as either “Pass” or “Fail” based on certain criteria. You have built a classification model to predict whether a student will pass or fail, and now you need to evaluate the model’s performance using a confusion matrix and other metrics.

Dataset:

The dataset contains the following columns: - **StudentID** (Categorical) - **ActualResult** (Categorical - Pass/Fail) - **PredictedResult** (Categorical - Pass/Fail)

StudentID	ActualResult	PredictedResult
1	Pass	Pass
2	Fail	Pass
3	Pass	Pass
4	Fail	Fail
5	Pass	Fail

Objective:

1. Construct a confusion matrix based on the `ActualResult` and `PredictedResult` columns.
2. Calculate the following evaluation metrics:
   - **Accuracy**
   - **Precision**
   - **Recall**
   - **F1-Score**
3. Analyze the performance of the model using the **ROC-AUC curve**.

Importing Necessary Libraries

Making DataFrame

	StudentID	ActualResult	PredictedResult
0	1	Pass	Pass
1	2	Fail	Pass
2	3	Pass	Pass
3	4	Fail	Fail
4	5	Pass	Fail

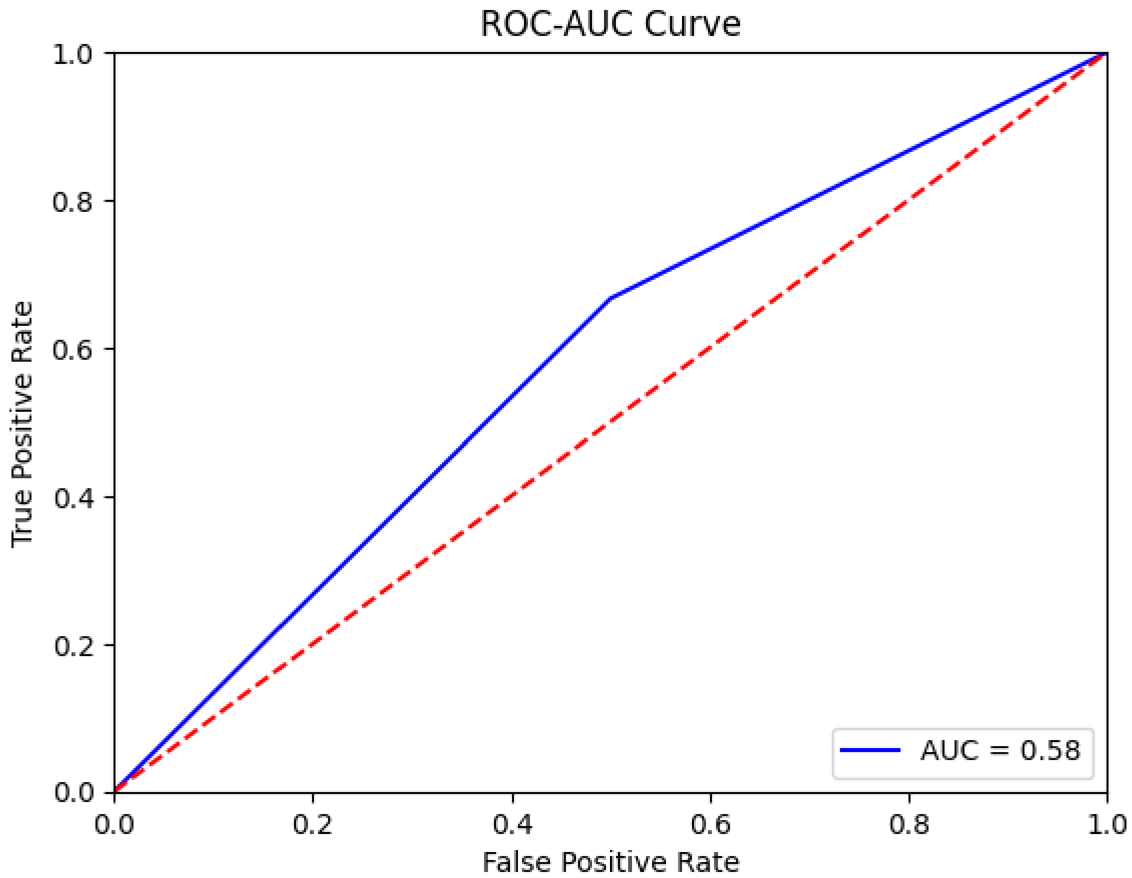
	StudentID	ActualResult	PredictedResult
0	1	1	1
1	2	0	1
2	3	1	1
3	4	0	0
4	5	1	0

Objective 1

Objective 2

```
Evaluation Metrics:
Accuracy: 0.60
Precision: 0.67
Recall: 0.67
F1-Score: 0.67
```

Objective 3


ROC-AUC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC = 0.58

Lab 4: Converting Categorical Data to Numerical Using One-Hot Encoding and Label Encoding

Problem Statement:

A retail company is analyzing the purchasing behavior of their customers using a dataset. Some features are categorical and need to be converted into numerical values to facilitate further analysis. In this task, you will convert the categorical features into numerical values using different encoding techniques.

Dataset:

The dataset contains the following columns: - **CustomerID**: Unique identifier for each customer. - **Gender**: Gender of the customer (Male, Female). - **Age**: Age of the customer. - **City**: The city where the customer lives (New York, Los Angeles, Chicago). - **Product**: The product purchased by the customer (A, B, C). - **PurchaseAmount**: The amount of money spent by the customer.

CustomerID	Gender	Age	City	Product	PurchaseAmount
1	Male	25	New York	A	100
2	Female	30	Los Angeles	B	200
3	Male	35	Chicago	C	150
4	Female	28	New York	A	120
5	Male	40	Los Angeles	B	250

Objectives:

1. **Identify categorical variables** in the dataset.
2. Apply **One-Hot Encoding** to the columns `City` and `Product` to convert them into numerical form.
3. Apply **Label Encoding** to the column `Gender` to convert it into numerical form.

Importing Necessary Libraries

Making DataFrame

	CustomerID	Gender	Age	City	Product	PurchaseAmount
0	1	Male	25	New York	A	100
1	2	Female	30	Los Angeles	B	200
2	3	Male	35	Chicago	C	150
3	4	Female	28	New York	A	120
4	5	Male	40	Los Angeles	B	250

Objective 1

Objective 2

	CustomerID	Gender	Age	PurchaseAmount	City_Chicago	City_Los Angeles	City_New York	Product_A	Product_B	Product_C
0	1	Male	25	100	0	0	1	1	0	0
1	2	Female	30	200	0	1	0	0	1	0
2	3	Male	35	150	1	0	0	0	0	1
3	4	Female	28	120	0	0	1	1	0	0
4	5	Male	40	250	0	1	0	0	1	0

Objective 3

	CustomerID	Gender	Age	PurchaseAmount	City_Chicago	City_Los Angeles	City_New York	Product_A	Product_B	Product_C
0	1	1	25	100	0	0	1	1	0	0
1	2	0	30	200	0	1	0	0	1	0
2	3	1	35	150	1	0	0	0	0	1
3	4	0	28	120	0	0	1	1	0	0
4	5	1	40	250	0	1	0	0	1	0

Lab 5: Predicting Diabetes Using Classification Models

Problem Statement:

You are tasked with predicting whether a patient is likely to be diagnosed with diabetes based on certain diagnostic features like pregnancies, BMI, insulin levels, age, etc. The goal is to preprocess the data, handle missing values, and train classification models to compare their performance.

Objectives:

1. **Data Preprocessing**:
   - Handle missing values in the dataset.
   - Scale the numerical features (e.g., BMI, age, insulin levels).
   - Perform a train-test split (70% training, 30% testing).
2. **Modeling**:
   - Train and evaluate the following classification models:
     - Decision Tree Classifier
     - Random Forest Classifier
3. **Evaluation**:
   - Use the following metrics to evaluate model performance:
     - Accuracy
     - Precision
     - Recall
     - F1-score
     - Confusion Matrix
   - Compare the performance of each model.
   - Provide insights into which model performs better and explain why.

Dataset:

- **Dataset Name**: Pima Indians Diabetes Database
- **Link**: Kaggle - Pima Indians Diabetes Database

Importing Necessary Libraries and Reading CSV File

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Objective 1

Handling Missing Values

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Scaling

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	0.639947	0.848324	0.149641	0.907270	-0.692891	0.204013	0.468492	1.425995	1
1	-0.844885	-1.123396	-0.160546	0.530902	-0.692891	-0.684422	-0.365061	-0.190672	0
2	1.233880	1.943724	-0.263941	-1.288212	-0.692891	-1.103255	0.604397	-0.105584	1
3	-0.844885	-0.998208	-0.160546	0.154533	0.123302	-0.494043	-0.920763	-1.041549	0
4	-1.141852	0.504055	-1.504687	0.907270	0.765836	1.409746	5.484909	-0.020496	1

Train-Test Split

Objective 2

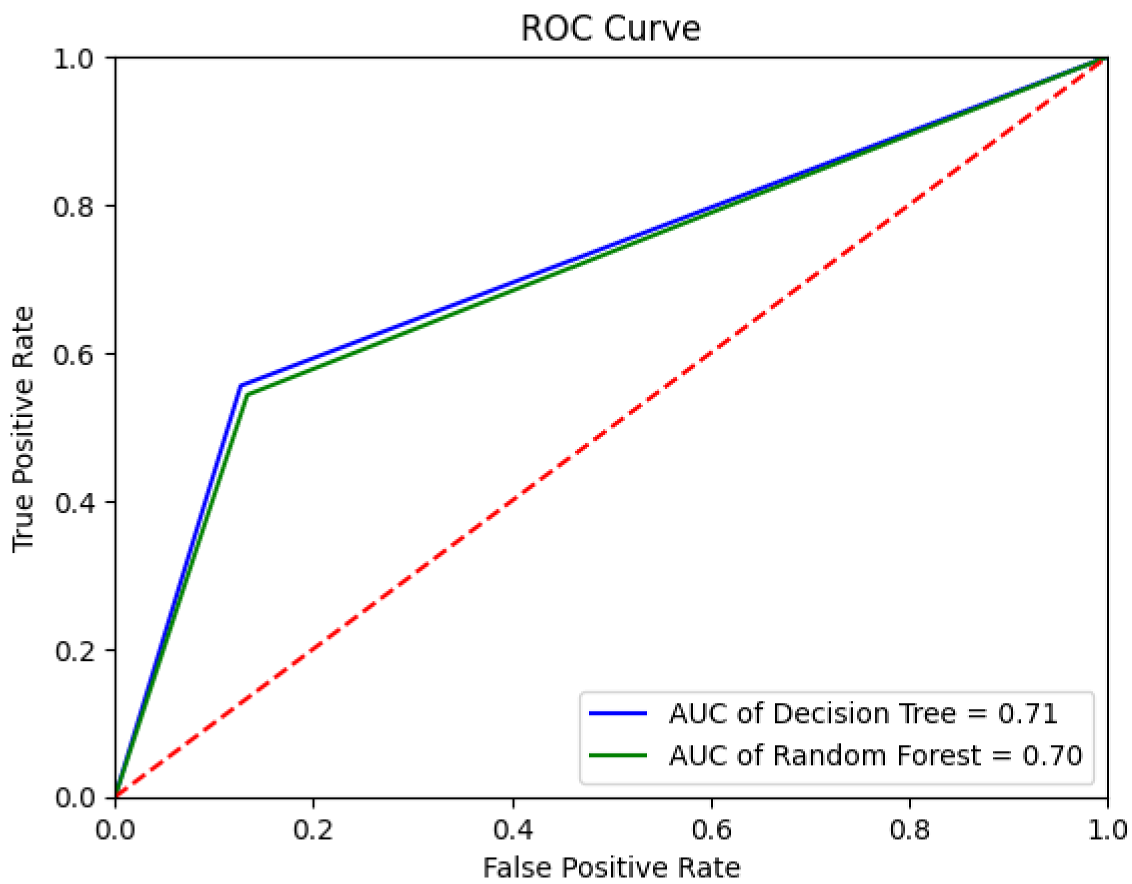
Decision Tree Classifier

Random Forest Classifier

Objective 3

```
Evaluation of Decision Tree
Accuracy Score is: 0.7619047619047619
Precision Score is: 0.703125
Recall Score is: 0.5555555555555556
F1 Score is: 0.6206896551724138
Confusion Matrix:
[[131  19]
 [ 36  45]]
```

```
Evaluation of Random Forest
Accuracy Score is: 0.7532467532467533
Precision Score is: 0.6875
Recall Score is: 0.5432098765432098
F1 Score is: 0.6068965517241379
Confusion Matrix:
[[130  20]
 [ 37  44]]
```



ROC Curve
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
AUC of Decision Tree = 0.71
AUC of Random Forest = 0.70
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate

Lab Question 7: Predicting Customer Churn Using Classification Models

Problem Statement:

A telecommunications company wants to predict customer churn (whether a customer will leave the company). The dataset contains various customer information, including contract type, monthly charges, tenure, etc. Your task is to pre-process the data and train multiple classification models to predict customer churn.

Objectives:

1. **Data Preprocessing:**
   - Handle missing values.
   - Convert categorical features (e.g., contract type) into numeric using encoding techniques.
   - Normalize or scale the numerical features.
   - Perform train-test split (75% training, 25% testing).
2. **Modeling:**
   - Train and evaluate the following models:
     - Naive Bayes
     - Support Vector Machine (SVM)
3. **Evaluation:**
   - Use accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix for model evaluation.
   - Compare model performance and justify which model performs best for predicting churn.

Dataset:

- **Dataset Name:** Telco Customer Churn
- **Kaggle Link:** https://www.kaggle.com/datasets/blastchar/telco-customer-churn

Importing Necessary Libraries and Reading CSV File

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	No
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Yes
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   gender            7043 non-null   object 
 1   SeniorCitizen     7043 non-null   int64  
 2   Partner           7043 non-null   object 
 3   Dependents        7043 non-null   object 
 4   tenure            7043 non-null   int64  
 5   PhoneService      7043 non-null   object 
 6   MultipleLines     7043 non-null   object 
 7   InternetService   7043 non-null   object 
 8   OnlineSecurity    7043 non-null   object 
 9   OnlineBackup      7043 non-null   object 
 10  DeviceProtection  7043 non-null   object 
 11  TechSupport       7043 non-null   object 
 12  StreamingTV       7043 non-null   object 
 13  StreamingMovies   7043 non-null   object 
 14  Contract          7043 non-null   object 
 15  PaperlessBilling  7043 non-null   object 
 16  PaymentMethod     7043 non-null   object 
 17  MonthlyCharges    7043 non-null   float64
 18  TotalCharges      7043 non-null   object 
 19  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(17)
memory usage: 1.1+ MB
```

Objective 1

Handling Missing Values

```
gender               0
SeniorCitizen        0
Partner              0
Dependents           0
tenure               0
PhoneService         0
MultipleLines        0
InternetService      0
OnlineSecurity       0
OnlineBackup         0
DeviceProtection     0
TechSupport          0
StreamingTV          0
StreamingMovies      0
Contract             0
PaperlessBilling     0
PaymentMethod        0
MonthlyCharges       0
TotalCharges        11
Churn                0
dtype: int64
```

```
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

Encoding

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport
0	0	0	1	0	1	0	No phone service	DSL	No	Yes	No	No
1	1	0	0	0	34	1	No	DSL	Yes	No	Yes	No
2	1	0	0	0	2	1	No	DSL	Yes	Yes	No	No
3	1	0	0	0	45	0	No phone service	DSL	Yes	No	Yes	Yes
4	0	0	0	0	2	1	No	Fiber optic	No	No	No	No

One-Hot Encoding

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	PaperlessBilling	MonthlyCharges	TotalCharges	Churn	...	StreamingMovies_No
0	0	0	1	0	1	0	1	29	29	0	...	1
1	1	0	0	0	34	1	0	56	1889	0	...	1
2	1	0	0	0	2	1	1	53	108	1	...	1
3	1	0	0	0	45	0	0	42	1840	0	...	1
4	0	0	0	0	2	1	1	70	151	1	...	1

5 rows × 41 columns

Scaling

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	PaperlessBilling	MonthlyCharges	TotalCharges	Churn	...	StreamingMovies_No
0	0	0.0	1	0	0.000000	0	1	0.11	0.001269	0	...	1
1	1	0.0	0	0	0.464789	1	0	0.38	0.215901	0	...	1
2	1	0.0	0	0	0.014085	1	1	0.35	0.010385	1	...	1
3	1	0.0	0	0	0.619718	0	0	0.24	0.210247	0	...	1
4	0	0.0	0	0	0.014085	1	1	0.52	0.015347	1	...	1

5 rows × 41 columns

Train-Test Split

Objective 2

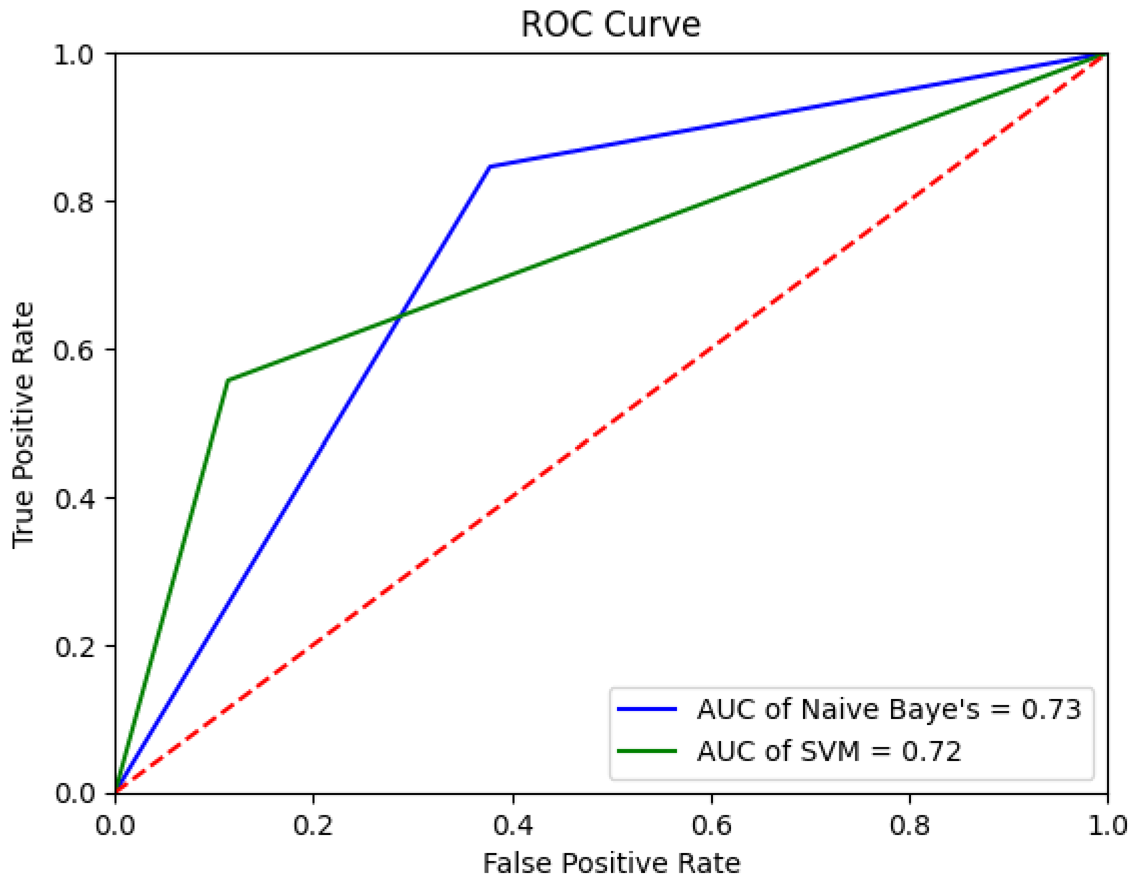
Naïve Baye’s

SVM

Objective 3

```
Evaluation of Naive Bayes
Accuracy Score is: 0.6814562002275313
Precision Score is: 0.44733861834654587
Recall Score is: 0.8458244111349036
F1 Score is: 0.5851851851851851
Confusion Matrix:
[[803 488]
 [ 72 395]]
```

```
Evaluation of SVM
Accuracy Score is: 0.7986348122866894
Precision Score is: 0.6388206388206388
Recall Score is: 0.556745182012848
F1 Score is: 0.5949656750572082
Confusion Matrix:
[[1144  147]
 [ 207  260]]
```


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC of Naive Baye's = 0.73
AUC of SVM = 0.72

Lab Question 8: Predicting Wine Quality Using Regression and Classification Models

Problem Statement:

You are working with a wine company that wants to predict the quality of wine based on various chemical properties such as acidity, alcohol content, sugar level, etc. The dataset contains numerical data on different types of wine and their associated quality rating (ranging from 0 to 10). Your task is to apply both regression and classification techniques to model the wine quality.

Objectives:

1. **Data Preprocessing:**
   - Handle missing values and outliers.
   - Scale or normalize the numerical features.
   - Split the dataset into training and testing sets (80% training, 20% testing).
2. **Modeling:**
   - Apply the following models for classification (quality as a categorical label):
     - Decision Tree Classifier
     - Random Forest Classifier
   - Apply the following models for regression (quality as a continuous label):
     - Decision Tree Regressor
     - Random Forest Regressor
3. **Evaluation:**
   - For classification models, use accuracy, precision, recall, F1-score, and confusion matrix.
   - For regression models, use Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.
   - Compare the performance of both types of models and explain which approach (classification or regression) better suits this problem.

Dataset:

- **Dataset Name:** Wine Quality Dataset
- **Kaggle Link:** https://www.kaggle.com/datasets/yasserh/wine-quality-dataset

Importing Necessary Libraries and Reading CSV File

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Objective 1

Handling Null Values

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

Normalizing the Data

Train-Test Split

Objective 2

Classifier

Decision Tree Classifier

Random Forest Classifier

Regressor

Decision Tree Regressor

Random Forest Regressor

Objective 3

Classifier

```
Decision Tree Classifier
Accuracy Score is: 0.5414847161572053
Precision Score is: 0.2942964872088583
Recall Score is: 0.31363312613312616
F1 Score is: 0.3022829060289604
Confusion Matrix is:
[[ 0  0  0  0  0  0]
 [ 0  1  2  2  1  0]
 [ 1  4 60 30  1  0]
 [ 1  3 35 47 13  0]
 [ 0  0  0 10 16  0]
 [ 0  0  0  1  1  0]]
```

```
Random Forest Classifier
Accuracy Score is: 0.5545851528384279
Precision Score is: 0.3556998016143444
Recall Score is: 0.40028085340585345
F1 Score is: 0.372111440710279
Confusion Matrix is:
[[ 0  0  0  0  0  0]
 [ 1  1  2  1  1  0]
 [ 2  7 59 26  2  0]
 [ 1  2 31 50 13  2]
 [ 0  0  0 10 16  0]
 [ 0  0  0  0  1  1]]
```

Regressor

```
Decision Tree Regressor
Mean Squared Error is: 0.5982532751091703
Root Mean Squared Error is: 0.7734683413748557
R2 Score is: -0.07508052909327612
```

```
Random Forest Regressor
Mean Squared Error is: 0.30738558951965067
Root Mean Squared Error is: 0.5544236552670265
R2 Score is: 0.4476181310396823
```