

## Environment Preparation:

```
pip install pandas==1.5.3 numpy==1.25.0 scikit-learn==1.4.2 nltk==3.8.1 joblib==1.3.2  
openpyxl==3.0.10 statsmodels==0.14.4
```

## Script 1:

**Name:** BCG\_classify\_customers.py

**Description:** This script processes **customer, activity, and complaints data** to predict customer churn using a **pre-trained machine learning model and combining it with a NLP based** random forest classifier. The output csv file will include the the customers complaint, the churn prediction, the classification of the problem, weather the problem was resolved or not, and weather the problem is short-term or long-term. It supports **Logistic Regression, LDA, and AdaBoost** models and applies the necessary preprocessing for each.

## Libraries:

- Pandas
- Joblib
- Argparse
- Scikit-learn
- Nltk

## Input:

- activity\_data.csv (Customer activity data)
- customer\_data.csv (Customer demographic and subscription details)
- complaints\_data.xlsx (Customer complaints data)
- logistic\_regression.pkl (Pre-trained Logistic Regression model for churn prediction)
- lda\_model.pkl (Pre-trained LDA model for churn prediction)
- adaboost\_model.pkl (Pre-trained AdaBoost model for churn prediction)
- logistic\_scaler.pkl (Scaler for data normalization - only for Logistic Regression)
- lda\_scaler.pkl (Scaler for data normalization - only for LDA)
- ada\_features.pkl (List of selected features used during training)
- logistic\_features.pkl (List of selected features used during training)
- lda\_features.pkl (List of selected features used during training)
- Random\_forest\_classifier.pkl (NLP classifier)

## Output:

- predictions\_logistic.csv (Churn predictions when using Logistic Regression)
- predictions\_lda.csv (Churn predictions when using LDA)
- predictions\_adaboost.csv (Churn predictions when using AdaBoost)

### **Example Execution Logistic Regression:**

```
python scripts/BCG_classify_customers.py \  
--activity sample_data/activity_data.csv \  
--customer sample_data/customer_data.csv \  
--complaints sample_data/complaints.xlsx \  
--model_type logistic \  
--model models/logistic_regression/logistic_regression.pkl \  
--scaler models/logistic_regression/logistic_scaler.pkl \  
--nlp models/nlp/random_forest_classifier.pkl \  
--features models/logistic_regression/logistic_features.pkl \  
--output predictions/predictions_logistic.csv
```

### **Example Execution Linear Discriminant Analysis (LDA):**

```
python scripts/BCG_classify_customers.py \  
--activity sample_data/activity_data.csv \  
--customer sample_data/customer_data.csv \  
--complaints sample_data/complaints.xlsx \  
--model_type lda \  
--model models/lda/lda_model.pkl \  
--scaler models/lda/lda_scaler.pkl \  
--nlp models/nlp/random_forest_classifier.pkl \  
--features models/lda/lda_features.pkl \  
--output predictions/predictions_lda.csv
```

### Example Execution AdaBoost:

```
python scripts/BCG_classify_customers.py \  
--activity sample_data/activity_data.csv \  
--customer sample_data/customer_data.csv \  
--complaints sample_data/complaints.xlsx \  
--model_type adaboost \  
--model models/adaboost/adaboost_model.pkl \  
--features models/adaboost/ada_features.pkl \  
--nlp models/nlp/random_forest_classifier.pkl \  
--output predictions/predictions_adaboost.csv
```

### Script 2 - Model Training:

**Name:** BCG\_train\_model.py

Description: This script processes **customer, activity, and complaints data** to **train or retrain a churn prediction model** using new data. It **preprocesses the data, trains the selected machine learning model (Logistic Regression, LDA, or AdaBoost), evaluates its performance, and saves the trained model along with necessary preprocessing artifacts**. This script is useful to retrain the model with newly available data.

#### Libraries:

- Pandas
- Joblib
- Argparse
- Scikit-learn
- Nltk

#### Input:

- activity\_data.csv (Customer activity data)
- customer\_data.csv (Customer demographic and subscription details)
- complaints\_data.xlsx (Customer complaints data)

## Outputs:

- Logistic Regression (`--model\_type logistic`)
  - logistic\_regression.pkl (Trained Logistic Regression model)
  - logistic\_scaler.pkl (Trained scaler for data normalization)
  - LOGISTIC\_features.pkl (List of selected features for Logistic Regression )
- LDA (`--model\_type lda`)
  - lda\_model.pkl (Trained LDA model)
  - lda\_scaler.pkl (List of selected features used during training)
  - lda\_features.pkl (List of selected features for LDA)
- AdaBoost (`--model\_type adaboost`)
  - adaboost\_model.pkl (Trained AdaBoost model)
  - ada\_features.pkl (List of selected features for AdaBoost)
- NLP (`--model\_type NLP`)
  - Random\_forest\_classifier.pkl (Trained model for NLP classification)

## Example Execution Logistic Regression:

```
python scripts/BCG_train_model.py \  
  
--activity sample_data/activity_data.csv \  
  
--customer sample_data/customer_data.csv \  
  
--complaints sample_data/complaints.xlsx \  
  
--model_type logistic \  
  
--output_model models/logistic_regression/logistic_regression.pkl \  
  
--output_scaler models/logistic_regression/logistic_scaler.pkl \  
  
--output_features models/logistic_regression/logistic_features.pkl
```

## Example Execution Linear Discriminant Analysis (LDA):

```
python scripts/BCG_train_model.py \  
  
--activity sample_data/activity_data.csv \  
  
--customer sample_data/customer_data.csv \  
  
--complaints sample_data/complaints.xlsx \  
  
--model_type lda \  
  
--output_model models/lda/lda_model.pkl \  

```

```
--output_scaler models/lda/lda_scaler.pkl \  
--output_features models/lda/lda_features.pkl
```

### **Example Execution AdaBoost:**

```
python scripts/BCG_train_model.py \  
--activity sample_data/activity_data.csv \  
--customer sample_data/customer_data.csv \  
--complaints sample_data/complaints.xlsx \  
--model_type adaboost \  
--output_model models/adaboost/adaboost_model.pkl \  
--output_features models/adaboost/ada_features.pkl
```

### **Example Execution NLP:**

```
python scripts/BCG_train_model.py \  
--activity sample_data/activity_data.csv \  
--customer sample_data/customer_data.csv \  
--complaints sample_data/complaints.xlsx \  
--model_type nlp \  
--output_model models/nlp/random_forest_classifier.pkl
```

