CSE 472

Machine Learning Sessional

Assignment 2 Report

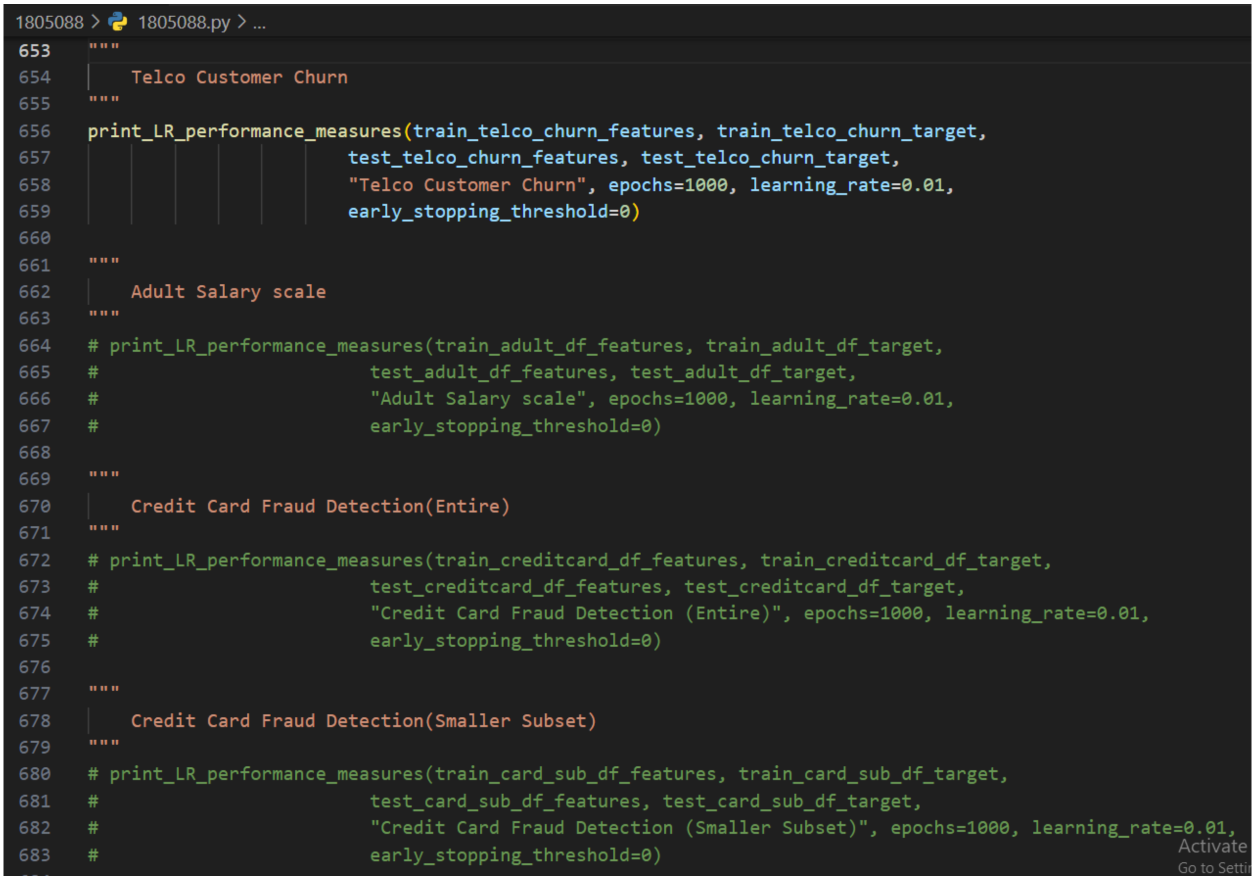
Logistic Regression and Adaboost for Classification

Shehabul Islam Sawraz

Student Id: 1805088

Steps to Run

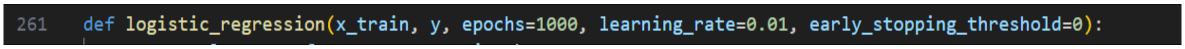
1. Make sure that you have below Python modules installed beforehand
   - **numpy** can be installed with ***pip install numpy***
   - **pandas** can be installed with ***pip install pandas***
   - **scikit-learn** can be installed with ***pip install scikit-learn***
2. In your workspace folder (where the script is located), create a **folder** named **Datasets** and place the **.csv datasets** file inside it
   - Telco Customer Churn(https://www.kaggle.com/blastchar/telco-customer-churn): Download the dataset and place it in the **Datasets** folder
   - Adult Salary Scale(https://archive.ics.uci.edu/ml/datasets/adult): Create a folder named **adult** in the **Dataset** folder and place the downloaded dataset files in the folder named **adult**
   - Credit Card Fraud Detection(https://www.kaggle.com/mlg-ulb/creditcardfraud): Download the dataset and place it in the **Datasets** folder
3. Type **python 1805088.py** to run the script. It will run all experiments and get required metrics as mentioned in specifications. It will print the metrics of the datasets first trained and tested using only logistic regression and AdaBoost sequentially.
4. To run **Logistic Regression** for a **specific dataset**, comment out the function calls unrelated to the dataset



```python
"""
    Telco Customer Churn
"""
print_LR_performance_measures(train_telco_churn_features, train_telco_churn_target,
                        test_telco_churn_features, test_telco_churn_target,
                        "Telco Customer Churn", epochs=1000, learning_rate=0.01,
                        early_stopping_threshold=0)

"""
    Adult Salary scale
"""
# print_LR_performance_measures(train_adult_df_features, train_adult_df_target,
#                          test_adult_df_features, test_adult_df_target,
#                          "Adult Salary scale", epochs=1000, learning_rate=0.01,
#                          early_stopping_threshold=0)

"""
    Credit Card Fraud Detection(Entire)
"""
# print_LR_performance_measures(train_creditcard_df_features, train_creditcard_df_target,
#                          test_creditcard_df_features, test_creditcard_df_target,
#                          "Credit Card Fraud Detection (Entire)", epochs=1000, learning_rate=0.01,
#                          early_stopping_threshold=0)

"""
    Credit Card Fraud Detection(Smaller Subset)
"""
# print_LR_performance_measures(train_card_sub_df_features, train_card_sub_df_target,
#                          test_card_sub_df_features, test_card_sub_df_target,
#                          "Credit Card Fraud Detection (Smaller Subset)", epochs=1000, learning_rate=0.01,
#                          early_stopping_threshold=0)
```

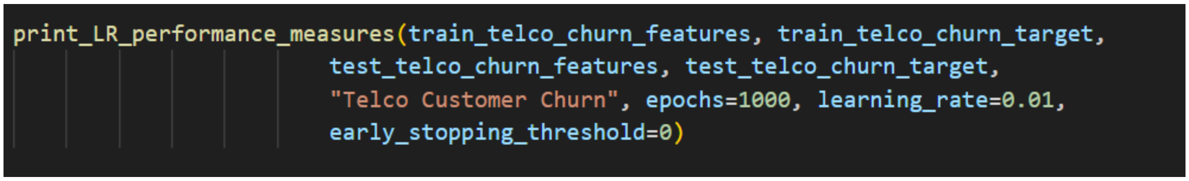
E.g. To run Logistic Regression for **Telco Customer Churn** dataset, comment the other function calls like shown above

5. . You can pass appropriate arguments to ***logistic_regession***



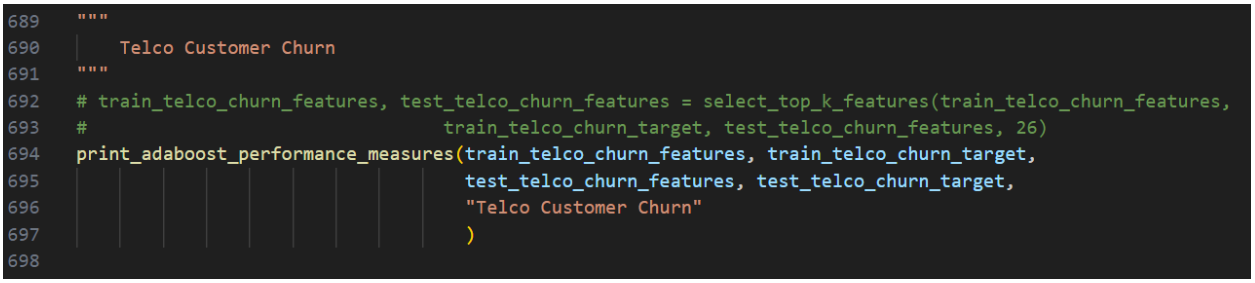
```python
def logistic_regression(x_train, y, epochs=1000, learning_rate=0.01, early_stopping_threshold=0):
```

function using the function parameters shown below:



```python
print_LR_performance_measures(train_telco_churn_features, train_telco_churn_target,
                        test_telco_churn_features, test_telco_churn_target,
                        "Telco Customer Churn", epochs=1000, learning_rate=0.01,
                        early_stopping_threshold=0)
```

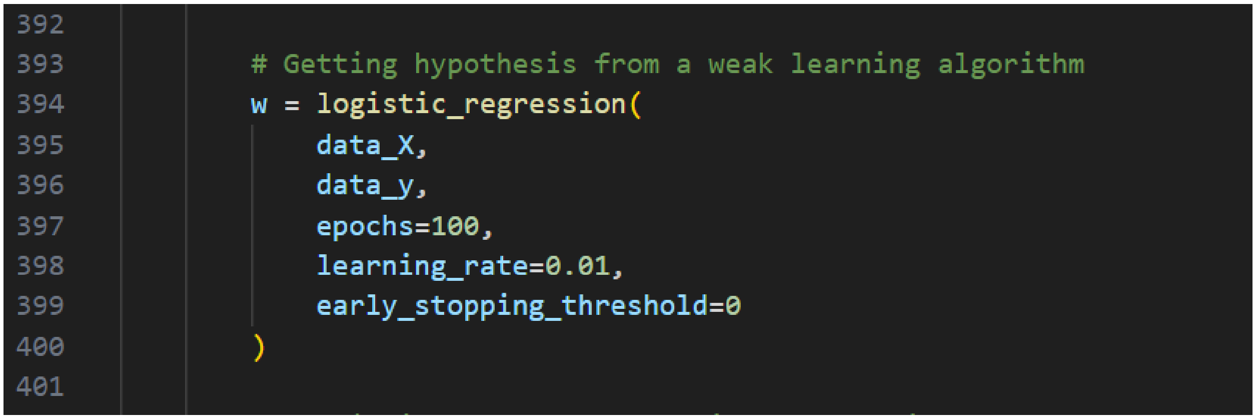
6. You can call **AdaBoost(adaptive_boosting)** function by using this function call:



```
"""
    Telco Customer Churn
"""
# train_telco_churn_features, test_telco_churn_features = select_top_k_features(train_telco_churn_features,
#                                   train_telco_churn_target, test_telco_churn_features, 26)
print_adaboost_performance_measures(train_telco_churn_features, train_telco_churn_target,
                                    test_telco_churn_features, test_telco_churn_target,
                                    "Telco Customer Churn"
                                    )
```

E.g. If you want to use information gain to evaluate attribute importance in order to use a subset of features, you can uncomment the commented line shown above.

7. You can also modify **logistic_regression** function parameters(like epochs, learning rate) to run **adaptive_boosting** more smoothly:



```
            # Getting hypothesis from a weak learning algorithm
            w = logistic_regression(
                data_X,
                data_y,
                epochs=100,
                learning_rate=0.01,
                early_stopping_threshold=0
            )
```

Adult

EPOCHS = 1000, LEARNING RATE = 0.01

Metric	Train	Test
Accuracy	0.821227	0.820158
Precision	0.604598	0.595545
Recall	0.744547	0.743889
Specificity	0.845550	0.843747
False discovery rate	0.395401	0.404454
F1 Score	0.667314	0.661502

EPOCHS = 100, LEARNING RATE = 0.01

Number of boosting rounds	Training	Test
5	0.836368	0.835759
10	0.830901	0.830661
15	0.837197	0.835636
20	0.837934	0.836435

Credit Card Fraud (Smaller Subset)

EPOCHS = 1000, LEARNING RATE = 0.01

Metric	**Train**	**Test**
Accuracy	0.995120	0.994387
Precision	0.990625	0.974683
Recall	0.804568	0.785714
Specificity	0.999812	0.9995
False discovery rate	0.009375	0.025316
F1 Score	0.887955	0.870056

EPOCHS = 100

Number of boosting rounds	**Training**	**Test**
5	0.995120	0.994387
10	0.995120	0.994387
15	0.995120	0.994387
20	0.994998	0.994387

Telco Customer Churn

EPOCHS = 1000, LEARNING RATE = 0.01

Metric	**Train**	**Test**
Accuracy	0.776002	0.783534
Precision	0.569736	0.544513
Recall	0.695595	0.755747
Specificity	0.805737	0.792648
False discovery rate	0.430263	0.455486
F1 Score	0.626406	0.632972

EPOCH = 100

Number of boosting rounds	**Training**	**Test**
5	0.796237	0.801277
10	0.785764	0.792051
15	0.786119	0.791341
20	0.793042	0.801987