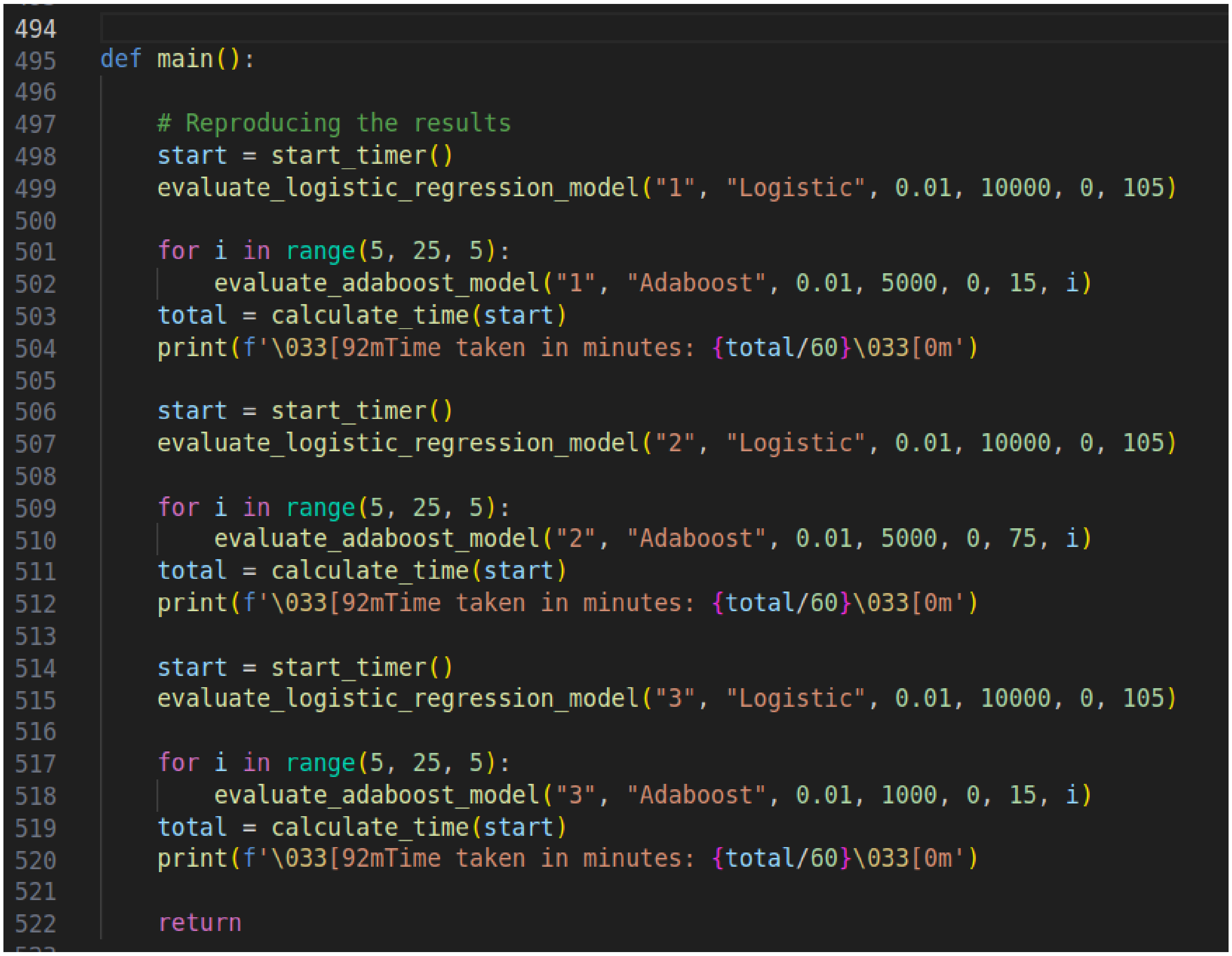
Logistic Regression and Adaboost

Report on Assignment 2

Md. Asif Haider
1805112

Steps to Run



```python
def main():

    # Reproducing the results
    start = start_timer()
    evaluate_logistic_regression_model("1", "Logistic", 0.01, 10000, 0, 105)

    for i in range(5, 25, 5):
        evaluate_adaboost_model("1", "Adaboost", 0.01, 5000, 0, 15, i)
    total = calculate_time(start)
    print(f'\033[92mTime taken in minutes: {total/60}\033[0m')

    start = start_timer()
    evaluate_logistic_regression_model("2", "Logistic", 0.01, 10000, 0, 105)

    for i in range(5, 25, 5):
        evaluate_adaboost_model("2", "Adaboost", 0.01, 5000, 0, 75, i)
    total = calculate_time(start)
    print(f'\033[92mTime taken in minutes: {total/60}\033[0m')

    start = start_timer()
    evaluate_logistic_regression_model("3", "Logistic", 0.01, 10000, 0, 105)

    for i in range(5, 25, 5):
        evaluate_adaboost_model("3", "Adaboost", 0.01, 1000, 0, 15, i)
    total = calculate_time(start)
    print(f'\033[92mTime taken in minutes: {total/60}\033[0m')

    return
```

1. Just run the python code as it is. The main function handles the reproduction of the reported results.
2. Take a look at the logistic regression classifier parameters. The parameters are **dataset number, type of algorithm, learning rate (alpha), training epochs, terminating threshold, top feature count** respectively in that order.
3. The adaboost classifier has an extra parameter titled **boosting round count**.
4. Take a look at the generation time associated, and feel free to comment out any portion of the three evaluation dataset as necessary.

1. Telco Customer Churn Dataset

Logistic Regression
Learning Rate = 0.01
Epoch = 10000
Terminating Threshold = 0
All Features

Performance Measure	Train Set	Test Set
Accuracy	**0.805**	**0.80**
True Positive Rate (Sensitivity)	0.544	0.553
True Negative Rate (Specificity)	0.899	0.889
Positive Predictive Value (Precision)	0.661	0.643
False Discovery Rate	0.339	0.357
F1 Score	0.597	0.595

Adaboost (Logistics Regression as Weak Learner)
Learning Rate = 0.01
Epoch = 5000
Terminating Threshold = 0
15/26 Features
Accuracy

Number of Boosting Rounds	Train Set	Test Set
5	0.795	0.795
10	**0.792**	**0.796**
15	0.782	0.783
20	0.788	0.780

2. Adult Salary Dataset

Logistic Regression
Learning Rate = 0.01
Epoch = 10000
Terminating Threshold = 0
All Features

Performance Measure	**Train Set**	**Test Set**
Accuracy	**0.852**	**0.850**
True Positive Rate (Sensitivity)	0.595	0.590
True Negative Rate (Specificity)	0.933	0.931
Positive Predictive Value (Precision)	0.738	0.725
False Discovery Rate	0.262	0.275
F1 Score	0.659	0.651

Adaboost (Logistics Regression as Weak Learner)
Learning Rate = 0.01
Epoch = 1000
Terminating Threshold = 0
75/103 Features
Accuracy

Number of Boosting Rounds	**Train Set**	**Test Set**
5	0.846	0.847
10	0.846	0.847
15	**0.847**	**0.847**
20	0.845	0.846

3. Credit Card Dataset

Logistic Regression
Learning Rate = 0.01
Epoch = 10000
Terminating Threshold = 0
All Features

Performance Measure	**Train Set**	**Test Set**
Accuracy	**0.999**	**0.999**
True Positive Rate (Sensitivity)	0.563	0.561
True Negative Rate (Specificity)	0.999	0.999
Positive Predictive Value (Precision)	0.874	0.846
False Discovery Rate	0.126	0.154
F1 Score	0.685	0.675

Adaboost (Logistics Regression as Weak Learner)
Learning Rate = 0.01
Epoch = 1000
Terminating Threshold = 0
15/30 Features
Accuracy

Number of Boosting Rounds	**Train Set**	**Test Set**
5	0.99890	0.99882
10	0.99890	0.99882
15	0.99892	0.99884
20	**0.99892**	**0.99886**

Observations

1. Adaboost performance tends to fall down a little bit compared to the base learner logistic regression
2. Increasing the boosting rounds does not always ensure performance improvement.
3. Usually 10-15 rounds of boosting is good enough for a stable performance.
4. Hyperparameter tuning plays a role in changing the performance metrics, and there was no fixed rule that applies to all datasets and algorithms.
5. Decreasing the number of features in boosting does not lower the performance of a solo logistic regression classifier in a big scale.