NLP 220 Assignment 2

University of California, Santa Cruz

Part A: Feature Engineering for SVM, Logistic Regression & Decision Tree (20 pt.)

In this first problem, you’ll implement features which can be used with the Support Vector Machine, Logistic Regression and Decision Tree classifiers implemented in scikit-learn.

Dataset

You are given an e-commerce dataset (ecommerceDataset.csv).

Get that from Data sub-folder of Files section in Canvas.

Dataset Format

The dataset contains categories of items and item description from an ecommerce website.

Dataset Tasks & Deliverables (5 pt.)

1. You will need to train/test 4-class classification problem using this dataset.
2. Create a train/test split using the train\_test\_split method from sklearn.model\_selection, and having it shuffle the data. Use a fixed random seed of your choice for reproducibility. Use 70% of the data for training, 10% for validation  and the remaining 20% for test.
3. Print or plot the distribution of your classes/labels, and include these in the report.

Feature Engineering

For this multi-class classification problem, apply 3 feature engineering techniques of your choice to a SVM, Logistic Regression and Decision Tree classifier. Use the implementations of each included in scikit-learn. In total, you’ll create 3 distinct Logistic Regression, 3 distinct SVM and 3 distinct Decision Tree classifier models in this section.

Feature Engineering Deliverables (12pt)

1. Briefly (1-2 sentences each) describe the features you created.
2. Compare the accuracy and F1 scores across your 9 models on the test data. Compute macro-average F1 for each case. Print confusion matrix.
3. Discuss any features/models that outperformed others. If some feature(s) out-performed others in one or both models, consider/discuss why, if there is a particular reason you might expect this. There may not be an immediately clear reason, there is no requirement to analyze further here or provide a reason if doing so would take significant analysis of the data/models.

**Training/Inference time comparison (3)**

Compare training and prediction/inference time of each of your classifiers. Which one would you recommend based on accuracy and train/inference time?

Hyper-parameter Exploration - 10 pt

For this part, you need to try different hyper-parameters for each classifier and compare the accuracy of the models for such settings. For feature engineering, fix the best set of features you found in the previous step

OneVsRest Exploration - 15 pt

Now model the problem as OneVsRest (This is one method to evaluate multiclass models by comparing each class against all the others at the same time. In this scenario we take one class and consider it as our “positive” class, while all the others (the rest) are considered as the “negative” class.).

Since this is a 4-class classification problem, you will get 4 OneVsRest accuracy (macro-F1) scores. Implement OneVsRest and find all the individual accuracy  (macro-F1)  scores and then find the average accuracy  (macro-F1). 

Plot the ROC curve and Precision-Recall curve for each case.