ASSIGNMENT 3: SVM, Probabilistic Models, Boosting

Full Marks: 100

In this assignment we will cover the following concepts taught in the class:

1. Support Vector Machine
2. Naive Bayes Classifier
3. Gradient Boosting

Q1. Support Vector Machine **(45)**

Problem Statement: Train a Support Vector Machine model to detect fake news articles!

Data Set Description:

Download the dataset here: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

The dataset contains four fields: title of the article, article text, subject of the article, and date of the article. Download both files Fake.csv and True.csv and combine the two datasets with the appropriate class label.

Assignment Tasks: In this assignment, you can scikit-learn SVM package to classify the above data set. You have to study the performance of the SVM algorithms. You have to randomly pick 70% of the data set as training data and the remaining as test data. You have to submit a report in pdf format. The report should contain the following sections:

1. Clean the text with the following preprocessing steps: a) Removal of Punctuations b) Removal of Stopwords c) Stemming using NLTK PorterStemmer d) Lemmatization using NLTK WordNetLemmatizer e) Removal of URLs f) Removal of HTML Tags (5)
2. Use count vectorizer to generate training feature vector. Create vocabulary from the count vectorizer (using training data). (5)
3. Generate training feature vector using the training data and the vocabulary. Train a Linear SVM on the training feature vector. (5)
4. Generate test feature vector using test data and the above vocabulary. Report test accuracy using the above trained linear SVM model. (5)
5. Reduce the vocabulary to ⅔ of the original vocabulary and generate train and test feature vectors using the reduced vocabulary and training and test data respectively. (5)
6. Train another linear SVM model with the new train feature vector and report test accuracy on the new test feature vector. (10)
7. Effect of data cleaning: using the raw train and test data (before preprocessing), generate training and test features from the above count vectorizer and train a linear SVM model on the train feature vector, report test accuracy on the test feature vector to analyze the effect of data preprocessing. (10)

Submission Guidelines: You should name your report file as (e.g., 18CS72P07_1.pdf). The submitted report file should be in pdf and have the following header comments. # Roll # Name # Assignment number. Also, submit the program file as (e.g., 18CS72P07_1.py)

Q2. Naive Bayes Classifier: **(20)**

a. Imagine that you are given the following set of training examples. Each feature can take on one of three nominal values: a, b, or c.

F_1	F_2	F_3	Category
a	c	a	1
c	a	c	1
a	a	c	0
b	c	a	0
c	c	b	0

How would a Naive Bayes system classify the following test example?

$F_1 = a, F_2 = c, F_3 = b$ (10)

b. Consider an example where X_1, X_2, and X_3 are all Boolean features and Y is a Boolean label. X_1 and X_2 are truly independent given Y and X_3 is a copy of X_2 (meaning that X_3 and X_2 always have the same value). Suppose you are now given a test example with $X_1 = T$ and $X_2 = X_3 = F$. You are also given the probabilities:

$$P(X_1 = T|Y = T) = p$$
$$P(X_1 = T|Y = F) = 1 - p$$
$$P(X_2 = F|Y = T) = q$$
$$P(X_2 = F|Y = F) = 1 - q$$
$$P(Y = T) = P(Y = F) = 0.5$$

Prove that the Naive Bayes decision rule for classifying the test example positively is:

$p \geq (1 - q)^2 / (q^2 + (1 - q)^2)$ (10)

Submission Guidelines: You should name your report file as (e.g., 18CS72P07_2.pdf). The submitted report file should be in pdf and have the following header comments. # Roll # Name # Assignment number.

Q3. Gradient Boosting **(35)**

Problem Statement: Diabetes classification using XGBoost Classifier.
Dataset: Please download the dataset from the link:
https://www.dropbox.com/s/c1q3qix77hclbf2/diabetes.csv?dl=0

The dataset consists of 9 features like “Glucose”, “BloodPressure” etc. The target variable is the field named as “Outcome” which holds 2 values, 0 or 1. Use a 80-20 train-test split for the experiment.

Assignment Tasks: In this assignment, please use the XGBClassifier available under XGBoost python package to perform classification on the Diabetes dataset. Please vary the following parameters and report the test set accuracies for each combination below: (35)

1. Learning rate = 0.1, objective = logistic regression
2. Learning rate = 0.1, objective = hinge loss
3. Learning rate = 0.3, objective = logistic regression, max_depth = 2
4. Learning rate = 0.3, objective = logistic regression, max_depth = 8
5. Learning rate = 0.7, objective = logistic regression
6. Learning rate = 0.7, objective = hinge loss
7. Learning rate = 0.7, objective = hinge loss, max_depth = 8
8. Learning rate = 0.3, objective = logistic regression, L1 regularisation = 0.2, max_depth = 8
9. Learning rate = 0.3, objective = logistic regression, L2 regularisation = 0.2, max_depth = 8
10. Learning rate = 0.3, objective = logistic regression, split finding algorithm = Approximation Algorithm (present in the original paper https://arxiv.org/pdf/1603.02754.pdf)

Submission Guidelines: You should name your report file as (e.g., 18CS72P07_3.pdf). The submitted report file should be in pdf and have the following header comments. # Roll # Name # Assignment number. Also, submit the program file as (e.g., 18CS72P07_3.py)