**Writeup for Assignment**

## Overview

I used python package “sklearn” to do this assignment. I do the first preprocessing the dataset by trying to understand what may be the influential features in the dataset, and then converted some categorical and order attributes into numeric labels.

After finishing preprocessing the data, I started to find the best classifier on this dataset. I tried to use 4 types of classifiers, kNN, Neural Network, SVM, and Random Forest to train on the dataset. The random forest classifier got a fairly good accuracy through cross validation with low computational expense.

For the model other than random forest, I did the second time preprocessing on the dataset, and trained the dataset with new features in those four model again. After second preprocessing on the dataset, except for the random forest classifier, the kNN and Neural Network models also had high accuracy. I selected the one with lowest mean cross validation error as the best model to predict the testing data.

Finally, I predicted the testing data with the best model, and predicted result of the 20 testing data is: ['B', 'A', 'B', 'A', 'A', 'E', 'D', 'B', 'A', 'A', 'B', 'C', 'B', 'A', 'E', 'E', 'A', 'B', 'B', 'B'].

## Data exploration

For data exploration, I tried to understand the purpose of this assignment, and learn about what all the values mean as possible at the first place. In first time preprocessing of the dataset, I converted categorical attributes like “num_window”, “user_name”, and “cvtd_timestamp” to numerical values by using dummy variable method.

In the second time preprocessing, I deleted some of features which has less influence on the prediction of the dataset based on the understanding of the purpose of this assignment. Also, in order to get better results from classifiers other than random forest, I normalized the data in each feature. To reduce the feature dimensions, I used the kernel PCA method and observed the change of variance to extract the principal features. I reduced the number of features down to 14 from more than 180 features. Then I performed second time training for kNN, Neural Network, and SVM classifiers to see the performance on the dataset after second time preprocessing.

## Prediction Modeling

I trained the first pre-processed data with all the features into 4 model, kNN, Neural Network, SVM, and Random Forest. The Random Forest model with number of estimators = 5 performed a really good prediction. I got more than 99% accuracy on it.

However, except for the Random Forest, the other models did not give good mean cross validation score. The accuracies are less than 40% for kNN, Neural Network, and SVM models. Since the random forest model did not need much work on features extraction before training. I expected those models would perform well after doing second time preprocessing.

After second preprocessing on dataset, by cross validation score, the kNN with number of neighbor = 1 and random forest with number of estimators = 20 got a fairly good accuracy with low computational expense. The accuracy are 98% and 96% respectively. Although Neural Network classifier also gave a not bad accuracy around 89%, with 2 layers and number of nodes = 40 per layer, that was more computationally expensive compared to the other classifiers. For SVM classifier, it still did not perform well in this dataset.

I selected two models to predict the testing data, Random Forest (with number of estimators = 5) and kNN (with number of neighbor = 1), since they gave the lowest two mean cross validation error with really low error variance. Thus, basically the bounded errors will also be the least two among these four classifiers. In addition, the random forest classifier had better performance on first preprocessing data, so I transform the testing data into the first preprocessing format and predicted them by random forest (Here, I skipped the dummy variables transformation, since the testing data had different number of values in those variables and will cause the input number of features not coincide with the training data), and transformed them to the second preprocessing format and predicted them by kNN classifier. By doing this I could also check the accuracy by two different classifier. The predicted classes for the given 20 testing data are:

Random Forest (# of estimators = 5):
['B', 'A', 'B', 'A', 'A', 'E', 'D', 'B', 'A', 'A', 'B', 'C', 'B', 'A', 'E', 'E', 'A', 'B', 'B', 'B']

kNN (k=1):
['A', 'A', 'A', 'A', 'A', 'E', 'D', 'B', 'A', 'A', 'B', 'A', 'B', 'A', 'E', 'E', 'E', 'B', 'B', 'B']
There were only four different predicted outcomes for these two classifier, so there

might not be a serious issue for the model selection. Since when k = 1 is more likely to overfit the training data, I will prefer Random Forest as the prediction model for this assignment. Thus, the prediction outcome is:
['B', 'A', 'B', 'A', 'A', 'E', 'D', 'B', 'A', 'A', 'B', 'C', 'B', 'A', 'E', 'E', 'A', 'B', 'B', 'B'] for the 20 given testing data.