MOKSH SHUKLA
180433

Assignment 1
CS771A: Introduction to Machine Learning

2021-08-30

Problem 1

Cleaning

First of all I start by cleaning the data, handling the missing values in data features.

- Replace '?' with Nan
- Replacing the Nan values of continuous columns with their mean value and in the categorical column *'num-of-doors'* with the element with highest frequency
- I drop the variables *'fuel-type', 'aspiration', 'num-of-doors', 'engine-location','compression-ratio', 'stroke'* from my dataframe as these features cannot effect the price value of car.
- We can handle the categorical variables by either one hot encoding them or label encoding them$^{\#}$.

# : I find that the model RMSE in both the cases fall to near about the same values at optimum k.

Train/Test Split

I divide my dataset in 70:30 for training and testing purposes. I am not using validation set split since kNN Regression does not involve any training in particular.

Part a

In this I write a simple *kNN Regressor* which has the following algorithm:

1. Take a data value(which lies in n-dimensional vector space) from test dataset and calculate its euclidean distance from each of the points in training dataset
2. Store these distances in an array and subsequently sort it.
3. Select the k smallest distances from this sorted array which represents the k nearest neighbours to test data point.
4. Now with these k nearest neighbours take the mean value of their target variable and assign it to the test data point.

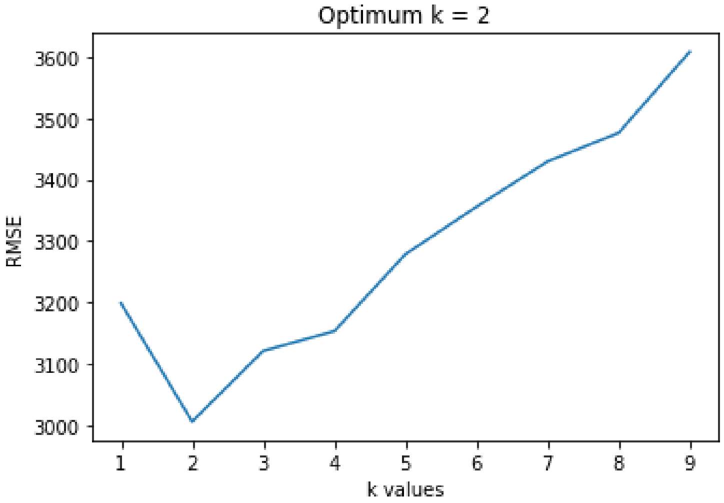
```
def kNNReg(X_train, y_train, X_test, k):
  y_pred = []
  for i in range(X_test.shape[0]):
    temp_dist = []
    for j in range(X_train.shape[0]):
      dist = np.linalg.norm(X_train[j]-X_test[i], 2)
      temp_dist.append(dist)
    temp_dist = np.array(temp_dist)
    idx = np.argsort(temp_dist)
    yhat = 0
    for m in range(k):
      yhat = yhat + y_train[idx[m]]
    y_pred.append(yhat/k)

  return y_pred

}
```

RMSE value on test set with k = 2 is 2591.288133248815

Part b

In this part I use kFold Cross Validation to find optimum value of k.

1. I iterate over k between 1 to 10 (range selected randomly).
2. For each value of k we perform kfold CV and get the average model RMSE score.
3. I plot RMSE vs k to determine optimum value of k which is the minima point in this curve.



Problem 2

Part a

Decision Tree accuracy with *maxLeafSize = 100, maxDepth = 12* is 83.35%.

- We create two classes *DecisionTree* which helps in creating the tree and *DecisionNode* which incorporates operations performed at the Node.
- The *DecisionTree* class initializes the tree with the *maxDepth*, *maxLeafSize* and also takes in a list *catCol* which signifies whether a column data feature is categorical or not.
- For any general node in the decision tree, we first find the *stump* feature value for splitting based on if its a categorical or numeric feature using information gain criteria.
- After getting the *stump* feature value we create 2 children nodes and call recursive function on that for numeric columns.
- For categorical columns we iterate over all the possible unique items in a particular category, create nodes for each one of them and call the *train* function for each one of these possible nodes.

Part b

On Tuning the hyperparametres *maxLeafSize, maxDepth* of the tree the highest accuracy comes out to be 83.54% with *maxLeafSize = 110, maxDepth = 6*