ANALYSIS REPORT

NAME:-Anand Sharma
MT19059

Question [1]

PREPROCESSING:-

- Null values of column pm2.5 are filled with mode of the column to balance the column
- Used LabelEncoder for transforming text data column name-cbwd . converted text to numeric value
- drop the column with cloumn name -no as it contains only index which is not of use

ASSUMPTIONS:-

- used column name month for classification task
- used column name pm2.5 for regression task
- Used year 2010 and 2012 for training data and year 2011 and 2013 for testing data
- Used accuracy for evaluation for classification task and MSE for evaluation for Regression task
- In Random Forest no of features =sqrt(p) in classification and p/3 in regression
- samples are taken as 10 % of training data for Bagging and Random Forset

Visualization:-

- shape of data is (43824,12)
- Training Data Shape:-(17544,12)
- Testing Data shape :-(17520,12)

Methodology/Algorithm:-

- **CLASSIFICATION:-**

1. Find the best split at each node by computing gini index at each node . The minimum gini index is taken to split the data on node . Gini index is simply calculated by formula

(len(left)*gini(left)+len(right)*gini(right))/total_len. Where gini=1-(Pi)**2

2. Accuracy is simply calculated by checking actual value and predicted value of model

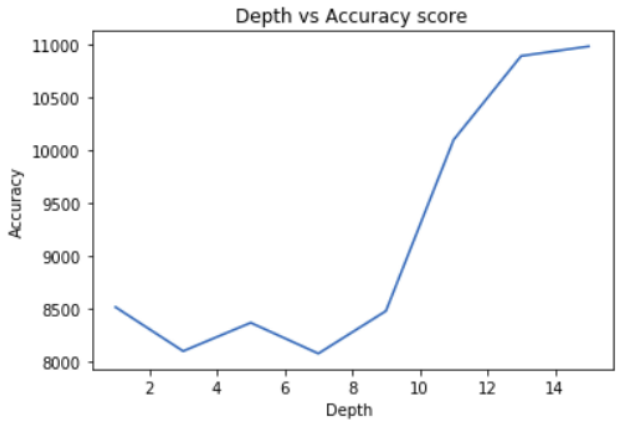
- **REGRESSION:-**

1. Best split is choosen by calculating variance and taking minimum variance as split value at each node . This will help us to minimize the variance and reduce MSE of model
2. MSE (Mean squared error) is calculated for evaluation of model by summing the square of difference of actual and predicted value/total_length

- Best split at each node is calculated . For each column and each unique values in row is choosen to find best split and then minimum is taken
- Main Tree is constructed using recursive approach with the help of dictionary for constructing tree . If stopping conditions are not met we recursively construct tree by taking node as split condition and left part and right part for prediction and model is fitted by tree
- Stopping condition are checked by 2 conditions:-Max depth of Tree and Minimum no of points in leaf node /terminal node
- When we get testing data it will follow tree and go according to condition of node to either left or right
- Finally at terminal/leaf node value is predicted by mean in regression and most occourring in Classification

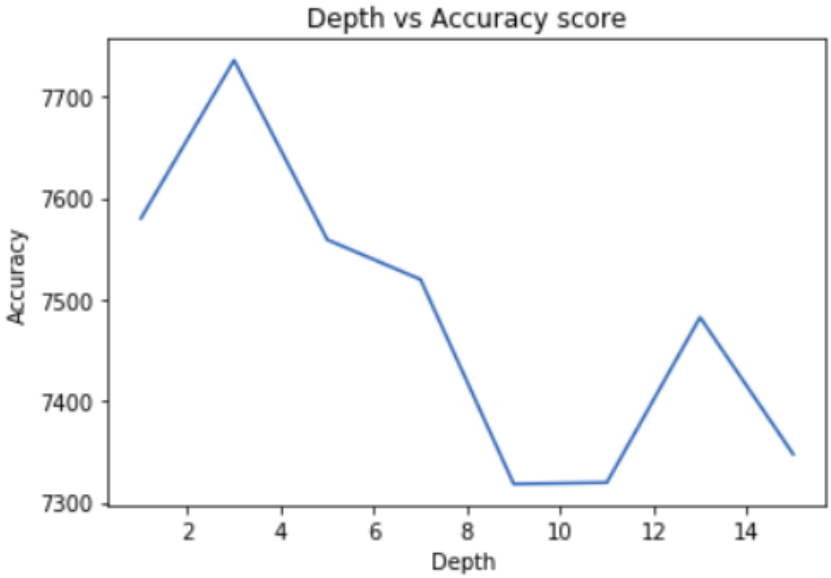
Results:-
REGRESSION:-

DECISION TREE REGRESSION:-



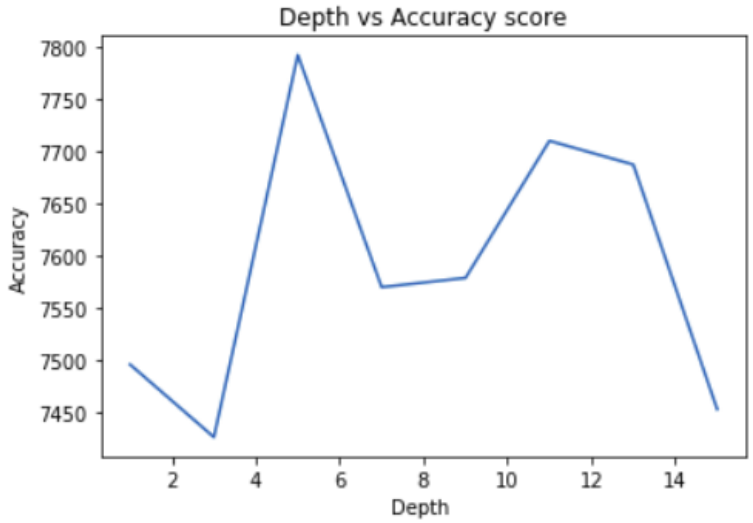

```
[8512.814756618802, 8096.411411723466, 8365.02391140759, 8073.4
98282149024, 8476.398118444982, 10094.16430717441, 10888.173608
782463, 10979.032638111205]
```

Bagging Decision Tree Regression max_depth=5 min_size=1 times=10



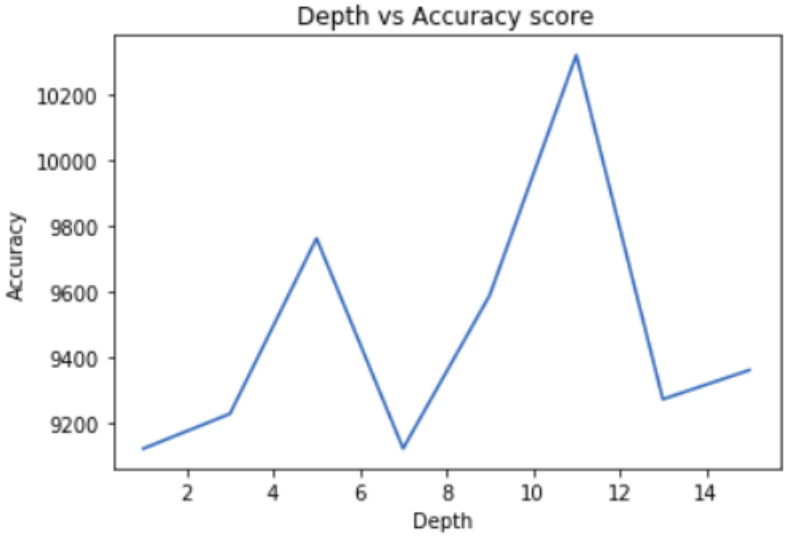

```
[7579.916107663811, 7735.715343372247, 7558.784739572034, 7519.
686348370448, 7318.234683155646, 7319.540622260736, 7482.387465
8820205, 7347.39553967722]
```

Bagging Decision Tree Regression max_depth=5 min_size=10 times=20



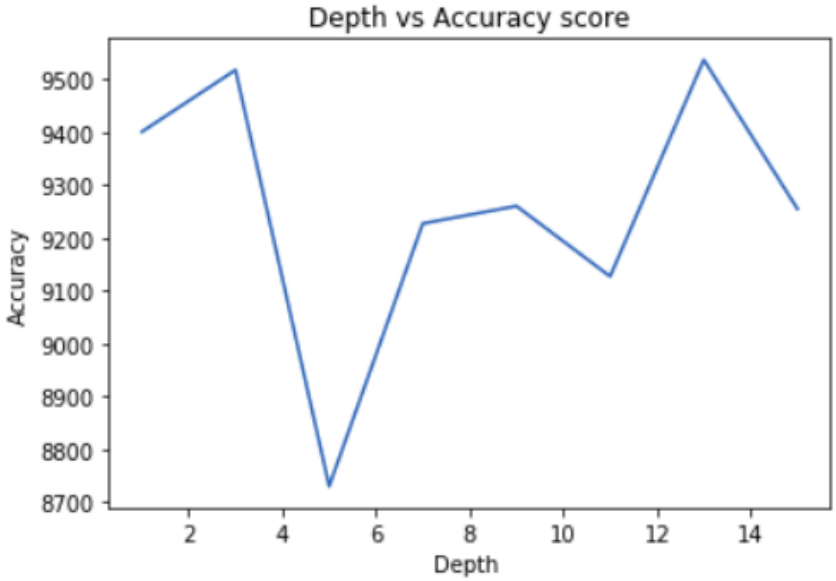

```
[7496.06023183, 7426.272393853065, 7793.022866754833, 7570.1987
98060734, 7579.08355791239, 7710.628019174745, 7687.80066576699
8, 7453.034422948188]
```

Random Forest Tree Regression with n=10 and min_no_of_sample=1 max_depth=5




```
[9124.979485898437, 9230.404470390155, 9762.133045948807, 9125.
337342495392, 9589.61439554842, 10317.504588656733, 9274.548906
54307, 9362.912225974223]
```

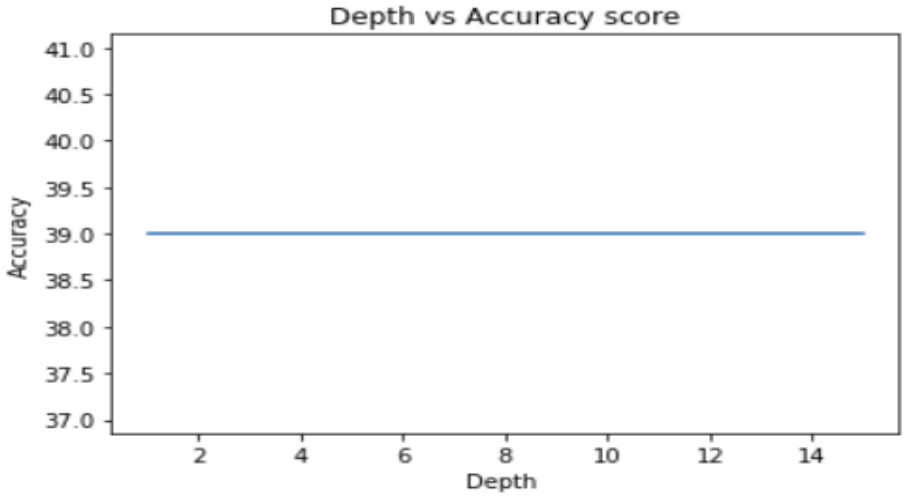
Random Forest Tree Regression with n=20 and min_no_of_sample=1 max_depth=5




```
[9400.80554687616, 9517.909574365221, 8729.507486684915, 9227.0
94542961187, 9260.085065475287, 9126.753501445914, 9537.2298024
83926, 9254.70887266683]
```

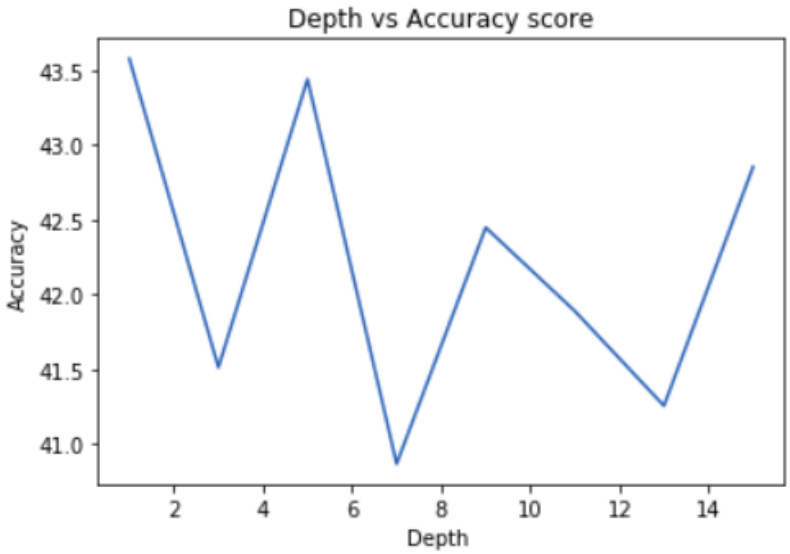
CLASSIFICATION

DECISION TREE CLASSIFICATION



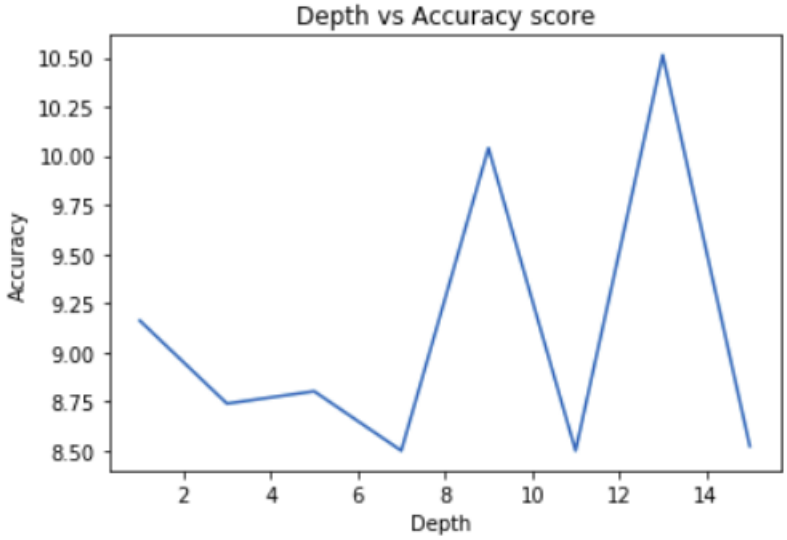

```
[39.00114155251141, 39.00114155251141, 39.00114155251141, 39.001
14155251141, 39.00114155251141, 39.00114155251141, 39.0011415525
1141, 39.00114155251141]
```

Bagging Decision Tree Classification with n=20 and min_no_of_sample=10




```
[43.57876712328767, 41.51255707762557, 43.44178082191781, 40.867
5799086758, 42.4486301369863, 41.88926940639269, 41.255707762557
07, 42.853881278538815]
```

Random Forest Tree Regression with n=50 and min_no_of_sample=10 max_depth=5




```
[9.16095890410959, 8.738584474885846, 8.801369863013699, 8.49885
8447488585, 10.039954337899545, 8.498858447488585, 10.5136986301
36986, 8.521689497716894]
```

Observations/Inferences:-

REGRESSION:-

- **with increasing max depth of decision tree beyond 7 it basically overfits the data and MSE increses with depth . With depth of range (2,7) it performs same approx . But after 7 it overfits and MSE increases rapidly**
- **When we use Bagging in Decision Tree we are able to reduce MSE from 8011 to 7319 (talking about minimum value) . Bagging makes the model more good performance by increasing depth upto certain level .It is not overfitting the data and MSE decreases with increase in depth**
- **Both graphs are nearly same with increase in n_estimatoprd from 10 to 20 . Slight variations are observed . It means that our bagging model has reduced MSE to minimum level (approx) and further no reductions are possible**
- **With Random Forest we observed that at depth of 5 and 7 our model has lowest MSE . It is not letting the model to overfit as we can see that model performance reamins same with increase in depth**

CLASSIFICATION

- **Normal Decision Tree classification has given the accuracy of 39. % for all depths . It means that with increase in depth there is no change of accuracy . When we evaluate on training data accuracy increases with depth but with testing data it remains nearly constant . Case of overfitting**
- **When We use Bagged Decision Tree Classification we can observe that at depth of 1 and 5 we achieved accuracy of 43 % which is more than normal decision tree**
- **Testing accuracy remain fluctuating with increase in depth as it is reliable model and performance is better compared to others**
- **In Random Forest accuracy is very low when compared to others classifiers basically no of features reduced from 12 to 4 which has let our model to perform poor**
- **with feature increased to 8 it also performs better**

QUESTION 2:-

Dataset Used:-

[[1, 0, -45], [2, 1, -51], [3, 2, -58], [4, 3, -63], [5, 4, -36], [6, 5, -52], [7, 6, -59], [8, 7, -62], [9, 8, -36], [10, 9, -43], [11, 10, -55], [12, 11, -64]]

Methodology/Algorithm:-

- Implemented Gaussian process Regression from scratch
- Values of l and sigmaf are evaluated by tuning the parameters by maximizing and optimization algorithm
- Optimize used:-np.sum(np.log(np.diagonal(L))) + \
  0.5 * Y_train.T.dot(lstsq(L.T, lstsq(L, Y_train)[0])[0]) + 0.5 * len(X_train) * np.log(2*np.pi)
- Using values of l and sigmaf mean and covarince are calculated
- First K matrix is evaluated using ans=math.exp(-0.5*((Xi-Xj)**2)/(l**2)) , ans=ans*(sigmaf)
- Kstar and Kstarstar is calculated
- Finally mean and covariance are calculated using
- temp1=np.matmul(Kstar,np.linalg.pinv(K))
- mean=np.matmul(temp1,Y)
- temp=np.matmul(temp1,Kstar.T)
- cov=Kstarstar-temp

Results:-

- X_train [0 2 4 6 8 10 11]

Y_train [-45 -58 -36 -59 -36 -55 -64]

- l_opt 7.1721672681317274 sigma_f_opt 4447.048169948306

K shape is (7, 7)
K shape is (8, 8)
K shape is (9, 9)
K shape is (10, 10)
K shape is (11, 11)

X= 1 Y = [-54.87935135]
X= 3 Y = [-53.33536361]
X= 5 Y = [-51.45986727]

```
X=  7 Y = [-51.26745822]
X=  9 Y = [-52.19383199]
```