Problem 2

1. We know that training a model on a dataset with N samples takes $O(N)$ time. We split the dataset on $\frac{N}{M}$ subsets of M samples each. Then, the first step is to train the $\frac{N}{M} - 1$ subsets and keep the last one as validation data. The complexity time of this step is $O(M \cdot (\frac{N}{M} - 1))$ since this is the number of the training data and also we don't take into account the test time.
   The second step is to repeat Step 1 $\frac{N}{M}$ times so that all of the chunks have been as validation data exactly once. This step results to $O(\frac{N}{M})$ time and overall we have $O(\frac{N}{M} \cdot M(\frac{N}{M} - 1)) = O(N(\frac{N}{M} - 1))$.
   The last step is to take the average of the $\frac{N}{M}$ results we have obtained from Step 2 which results to $O(\frac{N}{M})$ and since we average only once the total time is $O(N(\frac{N}{M} - 1) + \frac{N}{M}) = O(\frac{N^2}{M})$.

   Now, if $M = 5$ then the complexity time is $O(N^2)$ and if $M = \frac{N}{2}$ then the complexity time is $O(N)$. That shows that for $M = \frac{N}{2}$ the algorithm becomes faster which makes sense since at the Step 2 we only iterate twice(i.e constant number).

2. An advantage of picking M small is that your training data has size $N - M$ (large) and so the model since it fits more data it gives better predictions. Also, the average at Step 3 is taken over $N - M$(large) samples which smoothes out any possible extreme results of Step 2.

Problem 3

0.1 AirFoil Dataset:

k	Estimated Out of Sample Error
3.0	5.188885930831493
5.0	5.025039035761591
10.0	4.8597567584988965
20.0	4.824976669757174
25.0	4.821919099690949

Figure 1: Table corresponding to number of neighbors k and the out of sample error

The best predicted out of sample error is 4.85975 with $k = 10$ and the real error is 4.86297,so the accuracy is $\sim 10^{-2}$,pretty close.

0.2 AirQuality Dataset:

k	Estimated Out of Sample Error
3.0	4.664303568952087
5.0	4.663526595756293
10.0	4.756562417091101
20.0	4.922091178347992
25.0	4.989511957620365

Figure 2: Tablet corresponding to number of neighbors k and the out of sample error

The best predicted out of sample error is 4.6635 with $k = 5$ and the real error is 4.52821, so the accuracy is $\sim 10^{-1}$,which makes sense since the data are $\sim 10^{1}$ more.

Problem 1

1. **Variable Selection:** At a tree node in order to find the optimal split point and attribute, we seek through all possible attributes j and their values s so as to choose the pair(x_j, s) that minimizes an error function, e.g the sum of squared error. This a greedy approach so finding the optimal solution is not guaranteed. Finding the best binary partition in terms of minimum sum of squares is generally computationally infeasible.

$$\min_{j,\ s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

Figure 3: Wish to minimize the squared error over all (j, s) while the regions R_1, R_2 being fixed (feasible) at each node.

2. **Airfoil Dataset:** From the table the best model has $k = 12$ with error ~ 2.3876.

k	Estimated Out of Sample Error
3.0	4.1197938807947025
6.0	3.137399977924945
9.0	2.546506653421633
12.0	2.387650556291391
15.0	2.439688039735099

Figure 4: Tablet corresponding to the max-depth k and its respective sample error

After running it many times I noticed that $k = 15$ model outperforms the others. The difference between these two is very small.

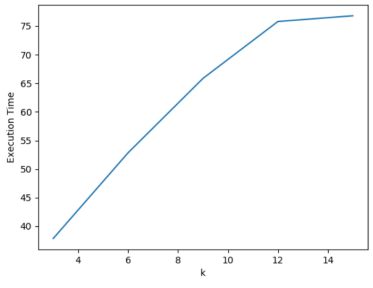

Figure 5: Figure corresponding to the time performance of cross validation on the different models.

3. **AirQuality Dataset:**

k	Estimated Out of Sample Error
20.0	3.674546773229431
25.0	3.6302176534681734
30.0	3.6394116414605673
35.0	3.616172050047334
40.0	3.6404224412197355

Figure 6: Tablet corresponding to the max-depth k and its respective sample error

Here the chosen model has $k = 25$ and sample error 3.63021 but once again $k = 30$ is potentially a good choice too.

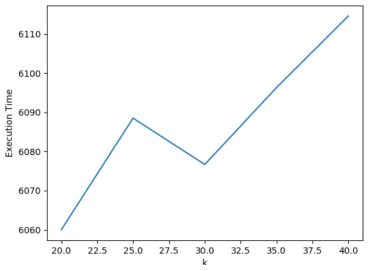

Figure 7: Figure corresponding to the time performance of cross validation on the different models.

1 Problem 4:

1. **Regularization:** Regularization is used so as to control overfitting in linear regression. In Ridge(L2 penalty) we shrink the coefficients β close to zero(small but nonzero) and reduce the model complexity while in Lasso(L1 penalty) except of shrinking coefficients , we manage to apply a feature selection since some coefficients become zero.

2. **Airfoil Dataset:**

alpha	Estimated Out of Sample Error
1e-06	3.825373880600763
0.0001	3.8254436932171334
0.01	3.833842746061066
1.0	4.013457461722741
10.0	4.5099030000093965

Figure 8: Tablet corresponding to the values of alpha and their sample errors

Ridge regression outperformed Lasso regression on the Airfoil Dataset. The best model has $\alpha = 10^{-6}$ and sample error 3.82537.

3. **AirQuality Dataset:** On the contrary , on the AirQuality Dataset the best model of

alpha	Estimated Out of Sample Error
0.0001	4.529082821014492
0.01	4.528998330728452
1.0	4.535453803511389
10.0	4.711218420267439

Figure 9: Tablet corresponding to the values of alpha and their sample errors

Lasso outperformed the best model of Ridge. The best model has $\alpha = 10^{-4}$ and error 4.52908.

2 Problem 5:

For the kaggle competition I used DecisionTreeRegressor with max-depth to be ks and various values for k in kfold.

1. **AirFoil:** ks = $[3, 5, 7, 9, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 40]$
   The best sample error I got is 2.24188 with $ks = 5$ and $k = 15$.(I tried different k but none of them gave me something better).

2. **AirQuality:** ks = $[5, 7, 10, 12, 15, 18, 20, 22, 25, 28, 30, 35, 40]$
   The best sample error I got is 3.3958 with $ks = 20$ and $k = 12$.

3 Problem 6: