**CE475 PROJECT**

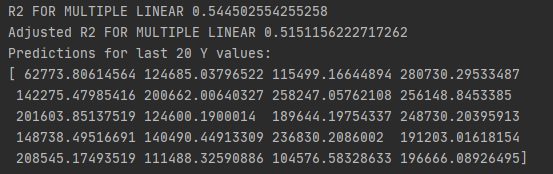
**EGE SEVİNÇ 20180601035**

**ATTEMPT 1: Multiple Linear Regression**

**Methodology:**

This method is a good starting point because it is simple yet the information it provides can be extremely beneficial.The relationship between data is mostly never linear(course slides,nonlinear.pdf page1) but what this method does is that it creates a foundation for us.When we try new methods we can compare R2 values to Multiple Linear Regression and continue if we see improvement.

**R2 , Adjusted R2 and Predictions:**

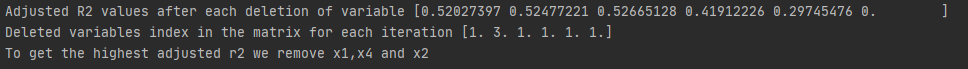


**Attempt 2: Backward Stepwise Selection and**

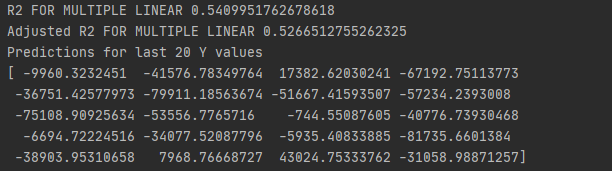
**Multiple Linear Regression**

**Methodology:**

Since we have 6 variables , trying to eliminate some of them is a good idea because some of them can be useless or even have negative effects for us(too many features can complicate the model).Too many variables also increases computational requirements so its a good idea to eliminate some.To select which variables to use and not to use in my model i decided to use Backward Stepwise Selection as a variable selection method.This method will start with all variables then remove one each iteration.I also added a deleted and adjusted\_rsqr calculator to my method so i can see which variables got deleted in which order and what happens to my adjusted R2 after these deletions.



So we delete X1,X2,X4 .I observed that after these 3 eliminations my adjusted R2 starts to get lower if i delete more.So i continued with 3 variables: X3,X5,X6.To check if my Backward Stepwise selection method is right i did Multiple Linear Regression to my new matrix with X3,X5,X6.



And the adjusted R2 score actually goes from 0.515(first attempt) to 0.526 so my new model has improved compared to my first attempt.I also predicted last 20 Y values and they are shown.

**Attempt 3: Dimension Increase and K-fold Cross validation(backward selection after dimension increase to be sure of my variables)**

**Methodology:**

So far we only worked with linear models but in reality the real answer is almost never linear so we now have to change our dimensions to get better results and move beyond linearity(course slides,nonlinear.pdf).We may have x3 multiplied with x5 or x6, we may have x5 multiplied with x6 or we can have all of them multiplied with each other.We could have even more multiplications but these gave me the best results and i don’t want to overcomplicate my model.So when we add these new dimensions our new matrix is this.



Now we treat the model like a Multiple Linear Regression model.

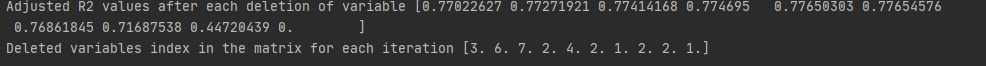
**And our new scores for our new matrix:**



As you can see R2 increasing is expected because we are adding new dimensions but there is a considerable increase in Adjusted R2 so we continue with this new matrix.

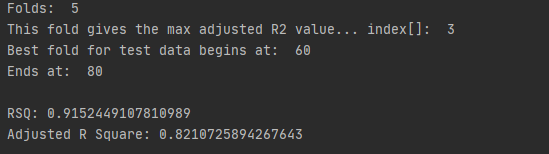
As a bonus:

I then checked with backward stepwise selection if we have any unnecessary variables in a different python file named check\_bigmatrix.



Turns out we didn’t and removing any variable would further decrease my adjusted R2 so i continued with all my variables.

Now we will use k-fold cross validation to see which foldsize and folds give us the best adjusted R2 score.I modified my code to give me the fold that results in the highest adjusted R2.



I got the best results when using 5 folds(foldsize 20).And the best fold out of these 5 folds was at index 3.Which starts at 60th row and ends at 80th.This is my test data and the rest will be my train data for the rest of my project because this method saw important increase in adjusted R2.

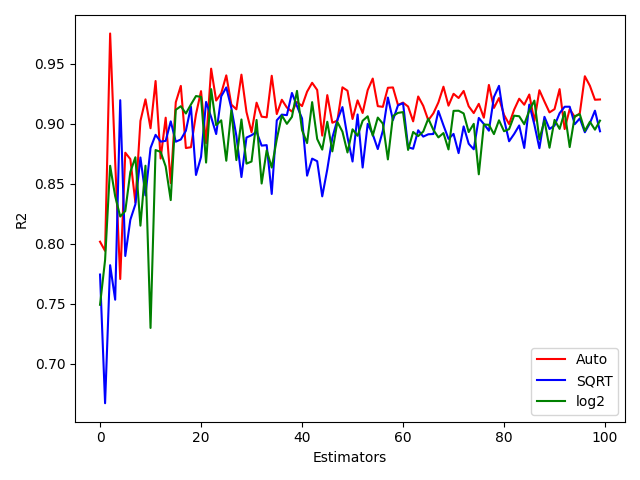
So that now im sure of my variables and test/train data i can move on to the next attempt.

**Attempt 4:**

**Random Forest**

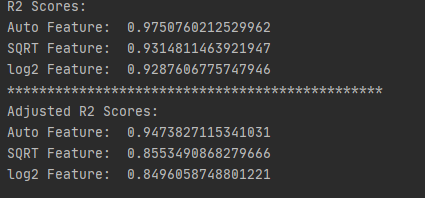
**Methodology:**

Now i wanted to try something that is explainable and can be graphically displayed.As said by our course slides (trees.pdf page30) decision trees are close to human decision-making thus perfect for interpretability.But trees have a disadvantage;they don’t have the same accuracy as some of the other models.Random forests construct multiple decision trees.Since trees have a habit of overfitting their training set, random forest corrects this by adding randomness to it.As said by our course slides (trees.pdf page37):When building these decision trees each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of p predictors.The split is allowed to use only one of those m predictors.So this method is perfect for me now since i know which data set to use(60 to 80 for test, rest for train) and which variables(as mentioned above).



I used auto,sqrt and log2 as features which i learned from sklearn documentation.

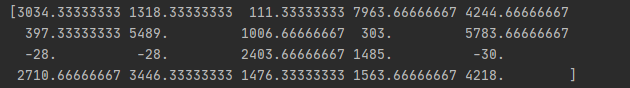
(https://scikit-learn.org/stable/user\_guide.html)



Out of all these Features Auto consistently gave me the best results for both R2 and Adjusted R2 .

So i predict last 20 Y values according to Auto feature

**Last 20 Y values predicted:**



**IMPLEMENTATION**

**Attempt 1:**

**Libraries and Modules:** NumPy,csv

My first attempt is a Multiple Linear Regression.This method is best for start because it gives me an idea of the data set and i can make my next attempts by taking Multiple Linear Regression as a benchmark which is very useful.

My python file has find\_coef , calculate\_ycap , rsqr and adjs\_rsqr methods in it.

**Results:** R2: 0.544… , Adjusted R2 : 0.515… From these calculations i see that i need to increase my R2 and adjusted R2 values.I also predicted 20 Y values in this step even though they are mostly wrong i wanted to see them to give me an idea.

**Attempt 2:**

**Libraries and Modules:** NumPy ,csv

In this attempt i started with Backward Stepwise Selection to eliminate some variables because i thought working with too many variables can be detrimental to my model.My code works in an iterative way:For each iteration my code deletes one variable then checks the adjusted R2 value of remaining variables(multiple linear regression).Then this code will pass the index of the deleted variable to an array called deleted.And the adjusted R2 of each iteration to an array called scores.From this i saw that my adjusted R2 goes up to 0.526 from 0.515 after 3 iterations but starts to go downhill from there.And when i check my deleted array.I see that at first iteration variable at index 1 is deleted which is X1(index 0 is the ones column) and the adjusted R2 goes up.Then at second iteration i see that variable at index 3 is deleted which responds to X4 now(ones,x2,x3,x4,x5,x6) and the adjusted R2 goes up.At third iteration variable at index 1 is deleted which is x2 and the adjusted R2 goes up again.But from now on every deletion of variable decreases the adjusted R2 value by a lot.So it is clear to me that i need to remove X1,X2 and X4.Then finally i try Multiple Linear Regression with these values to check if my calculations in Backward Stepwise Selection is correct.And they turn out to be correct so now my adjusted R2 value went up to 0.526from 0.515 so i keep X3,X5 and X6 as my variables from now on.

My file has the same functions from attempt 1 but with backward stepwise added to it.

**Result:** Adjusted R2 went up from 0.515 to 0.526 i also made my predictions for the last 20 Y values. They should be getting closer to their actual values.But they are still not close enough i did this just to see the effect of 0.11 change in adjusted R2 score.

**Attempt 3:**

**Libraries and Modules:** NumPy , csv

In this attempt i realised i need to create more variables because my work so far has only been linear.To do this we can multiply my existing variables with each other to get .My model so far is(1+X3+X5+X6) new variables i can add are: x3\*x5,x3\*x6,x5\*x6,x3^2,x5^2,x6^2 and x3\*x5\*x6.I could add more dimensions but this gave me best results.So now my new matrix has 10 variables.After creating my new matrix i decided to use k-fold cross validation and modified my code to give me the best fold for my test data and the rest for the train data.My code does this by calculating adjusted R2.I wanted to split my data 80 to 20 so i chose 5 folds.Now my fold size is 20.After this i saw that fold between 60-80 gave me the best result for adjusted R2.

Now my test data is 60:80 and the rest will be my train data.

As a bonus i tried backward stepwise selection again to check if i have any unnecessary variables.And i didn’t so now i know all my variables and which data to use for test/train i can go with last attempt.

My file has codes from the attempts before and after.This attempt covers 2 files.First i increase my dimensions in dimension\_and\_kfold then check if i need to delete any variables in check\_bigmatrix.Since i didn’t need to change my matrix now we continue with kfold in the first file.

**Results:**R2 went up to 0.915 (this is mostly because we add more variables) and Adjusted R2 went up to 0.821 this is a huge leap from my previous attempt.

**Attempt 4:**

**Libraries and Modules:** NumPy , csv,matplotlib.pyplot,from sklearn.ensemble RandomForestRegressor

I tried 3 different trees with Auto,Sqrt and log2 features since Auto is the most consistent and gives the best results i based my predictions on it.Most of this coding is done with sklearn and its RandomForestRegressor.I used number of estimators as 100 it could be more but we may need lots of computation power i found 100 to be consistent enough.

**Results:** I got my final R2 value as : 0.975 and my final Adjusted R2 value as : 0.947 from the tree with Auto feature. I also did my final predictions for 20 y values(image shown above).

**Conclusion**

I started this project some time in the middle of the semester and all i knew was some applications from my labs in this course and theoric information from my lectures.But as i worked more on it i learned a lot of things.My goal was to predict last 20 Y values with the data given to me.I started with a Multiple Linear Regression and i learned how important even the simplest mode l can be for problem solving.Then as lectures progressed in the course i realised that we don’t need every variable and some variables can affect my model negatively so i needed a way to remove variables and i used backward stepwise selection.I learned that we don’t need to have lots of variables we need to have the correct variables.After removing my variables and checking if i did everything right my next move was to increase my dimensions.(1+X3+X5+X6)^2 should be my new variables i also added all 3 multiplied because it proved useful to my model.In this step i learned that variables can be corellated with each other so multiplying them and adding as a new dimension is actually a good idea.After my new matrix is created i checked again if we have any variables to delete.But all of them were useful so i continued with 10 variables.This is more variables than what i started with but since i took the most convenient variables my adjusted R2 increased immensely.And i did K-fold cross validation with my now 10 variables.From this i learned best fold and foldsize.With my test data and variables set i will go with random forest.I learned that this is a easier to explain method which is why i ended my project with it.I learned that even though decision trees are somewhat less accurate than other methods, random forest builds many decision trees (100 in my project) to become more accurate.Also it adds randomness so it is a flexible model where overfitting does not occur most of the time.

As a summary, i am now able to understand and produce different machine learning models for given problems.I also learned how to approach problems in a more strategic way.I am happy about this because i want to learn and practice in A.I. field and this course was very helpful for me.