IZMIR UNIVERSITY OF ECONOMICS

SOFTWARE ENGINEERING

CE 475 – FUNDAMENTALS AND APPLICATIONS OF MACHINE LEARNING

Project Report

Umutcan Berk HASRET

20140601028

TABLE OF CONTENTS

1. IDENTIFICATION AND SIGNIFICANCE OF PROBLEM

This paper is prepared with the purpose of being a class project, CE 475. The aim of the project is to predict 20 data points by applying a machine learning algorithm and to train the system with the data that was provided by the class instructors. Throughout the paper, I have tried to apply different machine learning algorithms. I was aiming to reach two goals by doing this;

1. Figuring out the model and it's the underlying pattern that was used to generate the data. By this way, reaching the best prediction
2. To experience the differences between the certain algorithms and their consequences from the first hand.

The flow of this paper will be like the following;

In Methodology section, I'll be discussing the advantages and disadvantages of the mentioned algorithms. I will be also providing the how satisfying the results of the implemented algorithm as well along with the information while I'm giving.

In Implementation section, I'll be explaining how the process started and how I did them.

In Results section, I'll be providing visual contents to support my obtained results.

Last but not least, I will be sharing my opinions and experiences that I have obtained throughout the semester.

2.METHODOLOGY

As the first consideration, when we look at to a data we need to be precise about what are we trying to obtain from the it. As I stated above, the task of the project was to predict 20 new data points and since we were asked to predict some data points, it would be fair to say that the given problem is a regression problem rather than a classification problem.

Now that we know what kind of problem we are facing here; the phase of data analysis begins. There are 6 independent variables and 1 dependent variable in the given dataset. We are trying to see whether there is a correlation exists between these independent and dependent variables. My first approach was to implement multiple linear regression algorithm. I performed the implementation with two ways:

1. Adding **x1** to **x6** to the designed model that will be used for estimating the dependent variable, so every variable will play a role in the model
2. Using backward elimination technique to eliminate variables that has nothing to do with estimation of the dependent variable

Variable elimination is done for reducing the complexity of the model. Although there are couple of methods for variable elimination, the one that I used was the backward elimination since it's the easiest implementation practice when we compare it with forward selection and bi-directional selection methods.

Backward elimination technique suggests, if there are multiple variables in the given dataset, we should eliminate the ones with the highest p values. (Highest p value suggests the likelihood of that event will occur during the normal life). If the p value is below our confidence interval, it would mean that we can reject the null-hypothesis. By applying the mentioned algorithm, I extracted the variables that has nothing to do with the dependent variable, y in the model. Overall results of the multiple linear regression were poor. Which is the main reason why we move on to the second algorithm attempt; Polynomial Regression.

It is pretty like the multiple linear regression implementation in terms of coding experience however, the underlying idea is of course different. As the name suggests, in linear regression we are trying to fit a model to a given data set that is linear. In this case test errors are inevitable. Whereas, in Polynomial Regression, mentioned errors can be eradicated. As always in life, every benefit comes with a cost. The model that we have fitted on the system might explain the existed data so well but can do tremendously bad in the new data. This phenomenon is called "Overfitting". So, to wrap up the mentioned advantages and disadvantages of polynomial regression, it is an advantageous method to explain in some specific case data however linear models does a better job usually.

I tried to form 2nd degree, 3rd degree, 4th degree and 5th degrees of polynomial models to explain the data. My observation during this process, the incoming results were slightly better than the regular "Multiple Linear Regression" model especially (2nd degree), however as the degree of the model increases the value of the R^2 of that model decreased too. This means, our model has trained itself so well with the given train data, it would provide a poor performance while trying to predict new data. Because of the performance it has provided, I decided to change my approach to the question here and researched about non-linear regression models. Decision Tree was some sort of pioneer of this field so, I tried to implement this.

The reason why I tried this algorithm was the will of getting a better prediction score. After the implementation what I have achieved with Decision Trees was a great success comparing to the other two I have implemented so far. (Results will be reviewed in the Results section.)

While I was doing research about the Decision Trees, I ran into Random Forest Regression. The advantages that is provided by Random Forest Regression made me curious about it. It basically created lots of decision trees, their contents will be selected randomly. It uses the Bagging method which is a combination of learning models to improve the overall results accuracy.

3.IMPLEMENTATION

Since all the work that was performed throughout the term and the richness of the source, the language choice for my project is Python. I used Spyder as an IDE. To implement the mentioned algorithms in the 2nd chapter, the following libraries were used;

1. Pandas
2. Numpy
3. sklearn.linear_model ❼ LinearRegression
4. sklearn.model_selection ❼ train_test_split && cross_val_score
5. statsmodels.formala
6. sklearn.metrics ❼ r2_score && mean_squarred_error
7. sklearn.tree ❼ DecisionTreeRegressor
8. sklearn.preprocessing ❼ PolynomialFeatures
9. sklearn.ensemble ❼ RandomForestRegressor

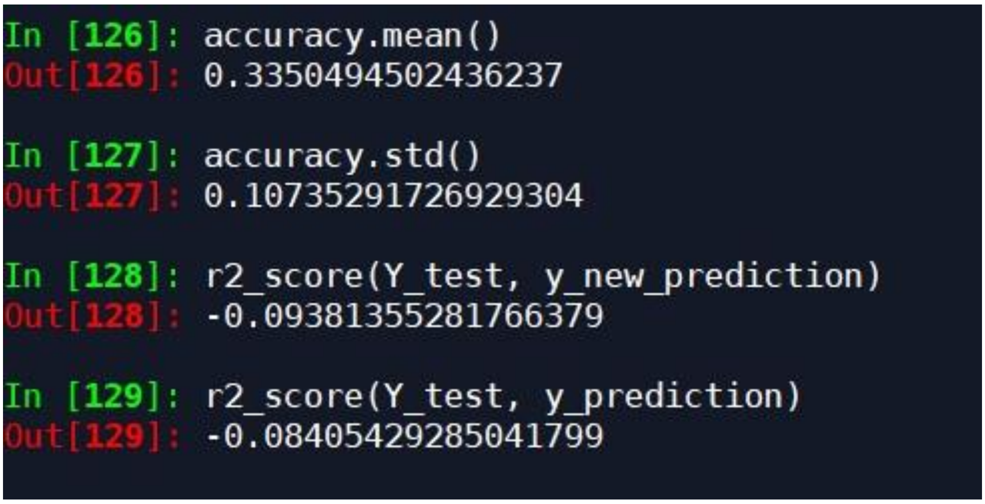
Since the names are pretty much self-explanatory, I will not be explaining every single library by one by one.

4.RESULTS

While we are interpreting the results, there will be a few constraints. To begin with, we are looking for the R^2 values to be big and MSE value to be small. I calculated mean and standard deviation after applying cross-validation technique each time to see the results.

Additionally, I realized that **x2** and **x6** have the same data in it, so it made me think; what would happen if I do not take one of them into consideration of the model. According to the result of my research of the Internet on this topic, people tend to avoid taking redundant data into consideration. The results that you will see below is not the consequence when I extract one of the redundant variables. The only time that I do that is in Random Forest Algorithm and as you will also see, taking one of the redundant variables out of the equation gave a slight increase on the accuracy of the overall model’s prediction performance.

Multiple Linear Regression:

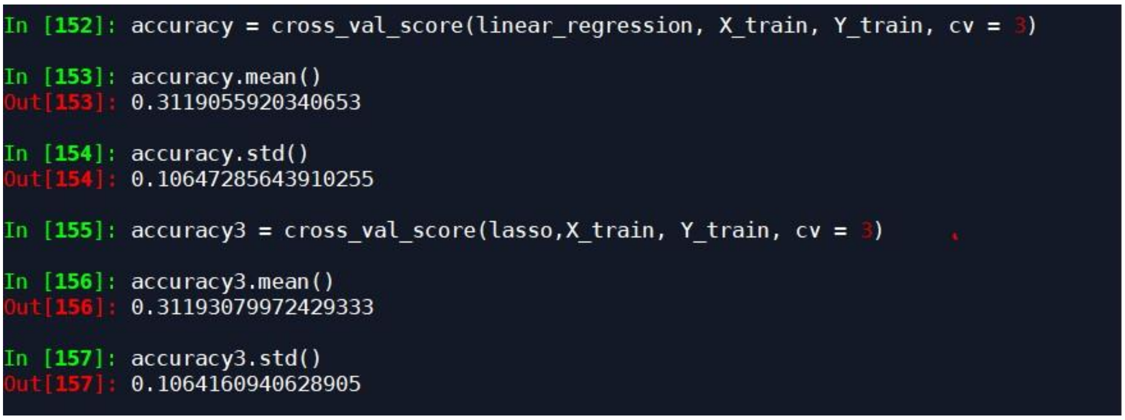


```
In [126]: accuracy.mean()
Out[126]: 0.3350494502436237

In [127]: accuracy.std()
Out[127]: 0.10735291726929304

In [128]: r2_score(Y_test, y_new_prediction)
Out[128]: -0.09381355281766379

In [129]: r2_score(Y_test, y_prediction)
Out[129]: -0.08405429285041799
```

Mean Squared Error: 935.77152765

Here, we see the implemented Multiple Linear Regression model's scores. According to it, this was not a really good model since the R² values are pretty low, meaning this will not help during prediction. The reason why we see two different r2_score with different results is I tried to see the effects of usage of Backward Elimination while doing multiple linear regression. **In [128]** is the one with Backward Elimination and **In [129] is** the results of when we avoid to apply any stepwise elimination and taking every variable into consideration.

Polynomial Regression:

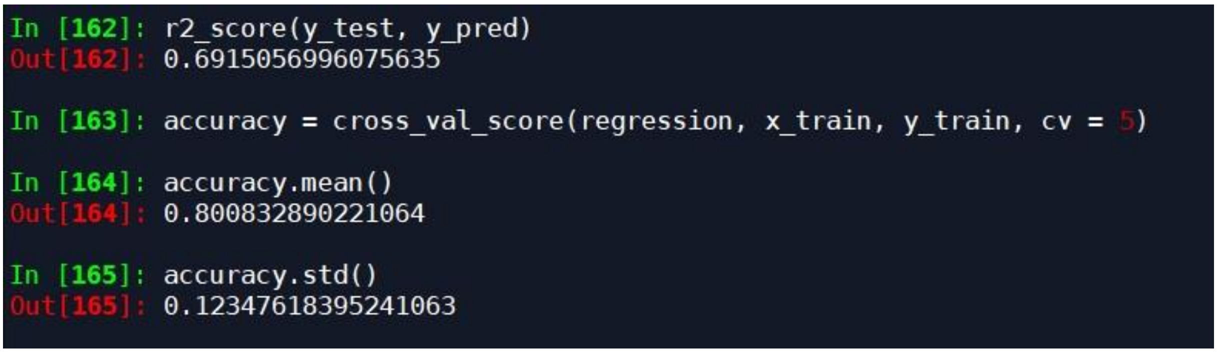


```
In [152]: accuracy = cross_val_score(linear_regression, X_train, Y_train, cv = 3)

In [153]: accuracy.mean()
Out[153]: 0.3119055920340653

In [154]: accuracy.std()
Out[154]: 0.10647285643910255

In [155]: accuracy3 = cross_val_score(lasso,X_train, Y_train, cv = 3)

In [156]: accuracy3.mean()
Out[156]: 0.31193079972429333

In [157]: accuracy3.std()
Out[157]: 0.1064160940628905
```

Mean Squared Error: 855.538677911

Here we are seeing the outcome of the Polynomial Regression Algorithm. I have also tried 3rd, 4th and 5th degree polynomial models but 2nd degree provided the best performance among them. As you might notice by now, there is a slight increase comparing with the Multiple Linear Regression while the chosen degree of the polynomial model is 2nd. That slight increase is growing even more when we also use the Lasso Regression to adjust the coefficient values of the independent variables.

Decision Trees:

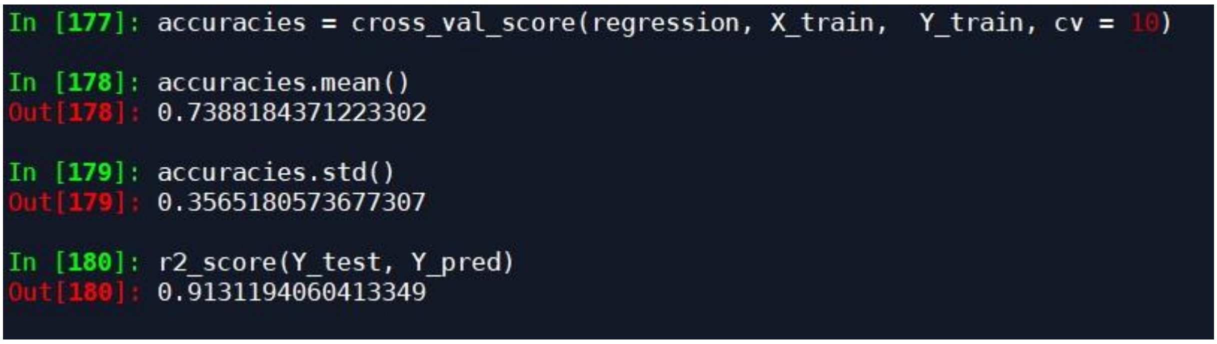


```
In [162]: r2_score(y_test, y_pred)
Out[162]: 0.6915056996075635

In [163]: accuracy = cross_val_score(regression, x_train, y_train, cv = 5)

In [164]: accuracy.mean()
Out[164]: 0.800832890221064

In [165]: accuracy.std()
Out[165]: 0.12347618395241063
```

Mean Squared Error: 496.960360592

When we use the Decision Tree algorithm, as you can see the increase on the R^2 can be noticed immediately. This is the best prediction score by now.

Random Tree Regression:



```
In [177]: accuracies = cross_val_score(regression, X_train,  Y_train, cv = 10)

In [178]: accuracies.mean()
Out[178]: 0.7388184371223302

In [179]: accuracies.std()
Out[179]: 0.3565180573677307

In [180]: r2_score(Y_test, Y_pred)
Out[180]: 0.9131194060413349
```

Mean Squared Error: 263.002478995

When we examine this and compare the R^2 values with the other algorithms that we have implemented, it is crystal clear this one is the best prediction algorithm with respect to the given dataset. I will be performing my predictions using this algorithm.

5.CONCLUSION

Throughout this course, I can easily say that I am confident on the root cause of why are we applying machine learning algorithms, what are the purposes of using them. I can explain every theory in a nutshell that we have seen so far. It was a great experience, especially this project was so great about putting every piece of the puzzle together. It helped me to develop a new way of thinking.

About the project, I have implemented four different algorithms to see how they differ from each other and to decide which one to use while doing the predictions. Finally, I decided to use Random Forest Regression since it had the highest R^2 value with the lowest MSE.

Umutcan Berk HASRET

20140601028