A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | Alina Marrulyn Lozano Flores |
| *Student Number* | 2024068 |
| *Module Title* | Machine Learning (10 ETCS) |
| *Assessment Title* | Prediction of house pricing |
| *Assessment Due Date* | 21-04-2024 |
| *Date of Submission* | 20-04-2024 |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

**Contents**

[Introduction 2](#_Toc164452407)

[Data Cleaning 2](#_Toc164452408)

[Exploratory data analysis - EDA 2](#_Toc164452409)

[Dummies and scalation 4](#_Toc164452410)

[**Scaling our Data** 6](#_Toc164452411)

[Our Models 6](#_Toc164452412)

[Splitting our data 6](#_Toc164452413)

[Random Forest Regressor 6](#_Toc164452414)

[Feature importance 7](#_Toc164452415)

[Results 7](#_Toc164452416)

[Results with Hypertuning 8](#_Toc164452417)

[Support Vector Regression 8](#_Toc164452418)

[Results 8](#_Toc164452419)

[Results with Hypertuning 9](#_Toc164452420)

[Hypertuning with feature importance SVR 9](#_Toc164452421)

[Cross Validation 10](#_Toc164452422)

# Introduction

Data analysis is widely used in housing to predict prices. Machine learning can be really useful to predict prices in the future or resolve the price of one property that we want to sell.

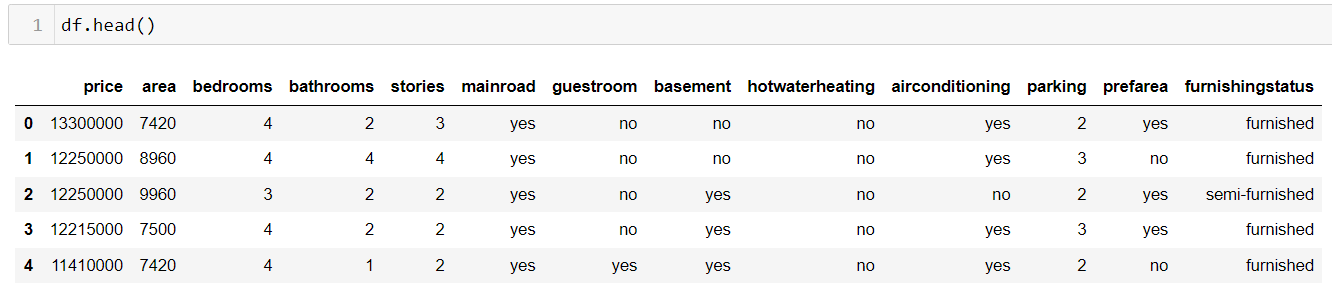
This dataset contains a total of 13 variables and the goal of this project was to predict the prices. We decided to use two different methods, Random Forest Regressor and Support Vector Regression (SVR) and compare the results between each other through Cross Validation (CV), to compare what model is better on this case.

We are going to have three training splits 20%, 25% and 30%, the intention is to compare the results.

# Data Cleaning

# Exploratory data analysis - EDA

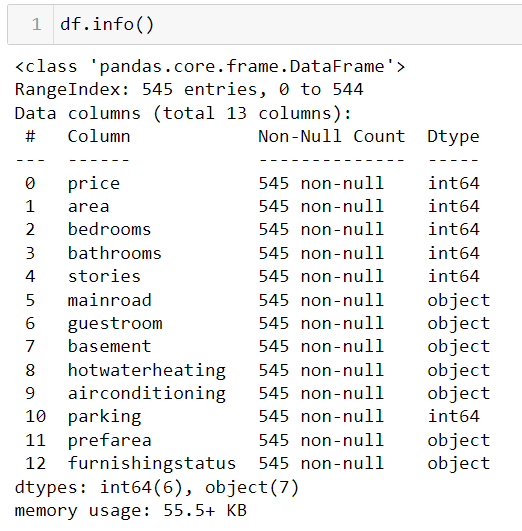
For this analysis our data set is called Housing in format csv, with the function “df.head()” we can see the first five columns:



*Fig 1*

As we can see in the figure 1, we have six numerical values, and seven categorical columns. With the function info, “df.info()” we can observe that we have 545 rows or samples.

With the funciton info( ), we can see the type of columns that we have as well.



*Fig. 2*

All the columns are the same type that is supposed to be, any column with numerical values is type object, and also with the function “nan\_count = df.isna().sum()” we can corroborate that there are not nan values (Fig. 3).

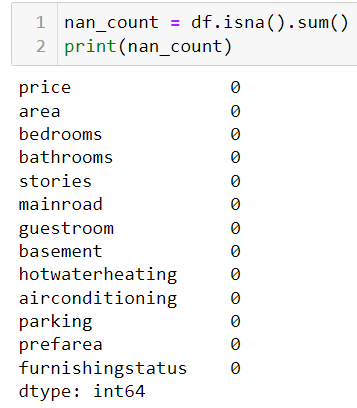


Fig. 3

For categorical data is important to convert this information in number and values, we use dummies to convert our values in 0 and 1, being 0 no and 1 yes,” pd.get\_dummies(df)”

We can observe in the figure 4, that we have now 21 columns because our categorical data is split in 2 instead 1.

# Dummies and scalation



Fig.4

In the figure 5, we can observe the correlation between price and area, the first graph shows the correlation with the outliers and the second graph the correlation without them.

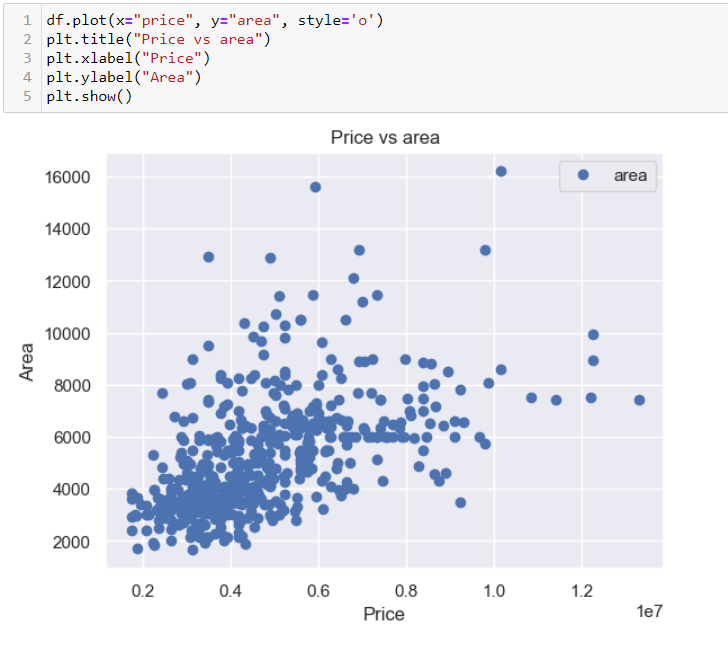
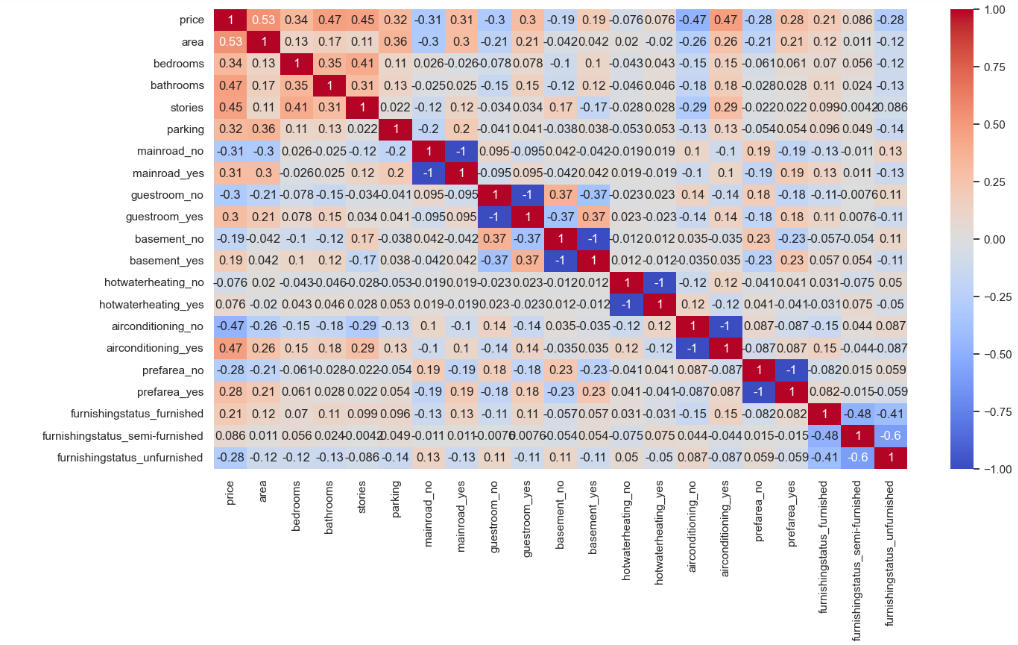


Fig 5

**Heatmap/ Correlation matrix**

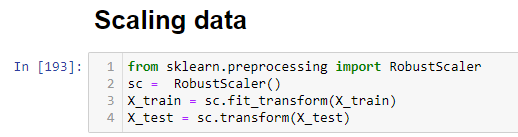
The most important feature to predict the price is the area followed by the feature bedrooms and bathrooms.

****

*Fig.6*

# **Scaling our Data**

We used RobustScaler for our analysis. This is to make sure that the statistical features of our dataset are on the same scale. It is similar ro rhe StandarScaler but the RobustScaler uses the median and quartiles instead the mean and variance. This can help us to remove outliers.

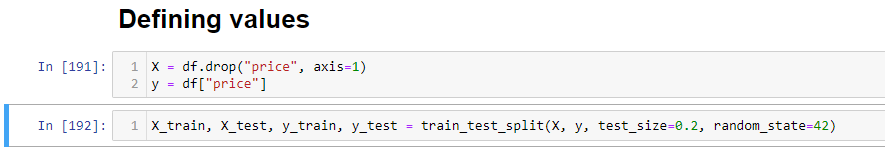
****

*Fig. 7*

# Our Models

## Splitting our data

For this project we are going to split our data 3 times, the test size is going to be 20%,25% and 30% and we are going to compare our results. In the figure 8 we can observe that our test size is 0.2 (20%).

****

*Fig. 8*

We decided to use two methods and after compare them, SVR and Random Forest Regressor, SVR is very recommended in general to predict prices RFR is good to test because is based in decisions and hierarchy, is used for classification and regression tasks.

## Random Forest Regressor

On the figure 9 we can see the parameter of our code:

*N\_estimators* .- The number of decision trees in our code is 300

*Max\_depth* .- The maximum possible depth of each tree is 5

*Max\_features*.- The maximum number of features the model will consider determining a split, on this case *square root*.

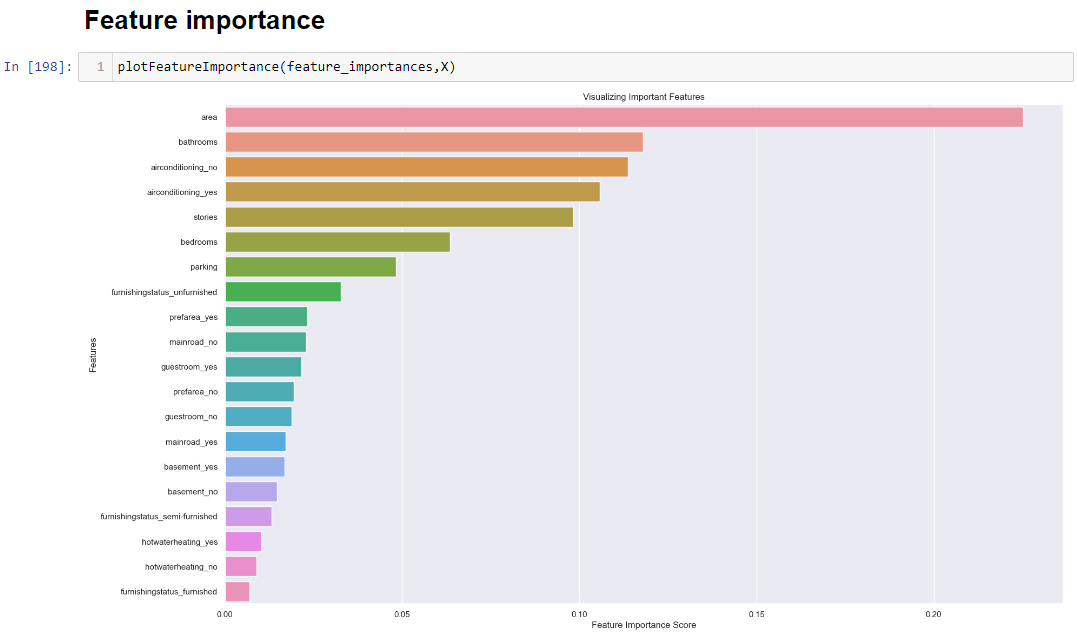
*Random state.-*  We define the random used to shuffle the data before splitting it. On this case is 42

****

*Fig. 9*

### Feature importance

The graph below (fig.10), shows that the most important feature related with our dependent variable is the area followed by the quantity of bathrooms.

****

*Fig.10*

### Results of our RFR

With a test size of 20%, the results of our metrics shows that our training data has a score of 65%, however our test data has a score of 57%, this means that our data could slightly be over fitted.(Fig.11)

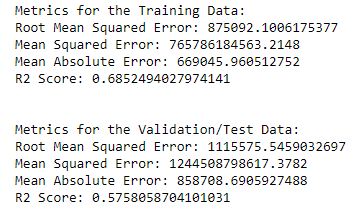
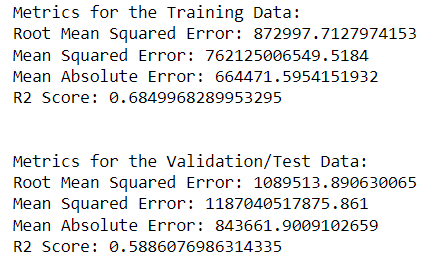


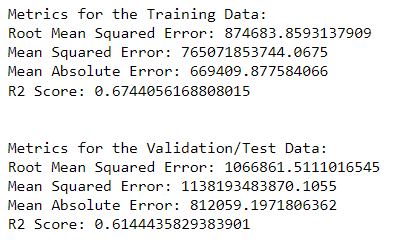
Fig. 11

With a test size of 25%, the results of our metrics shows that our training data has a score of 68%, however our test data has a score of 58%, this means that our data could slightly be over fitted.(Fig. 12).



*Fig. 12*

With a test size of 30%, the results of our metrics shows that our training data has a score of 67%, however our test data has a score of 61% (Fig. 13).



*Fig.13*

### Results with Hypertuning

The hyper parameter optimization allows us to choose the best parameters that our model could have.

The results, with a test *size of 20*%, after tuning the hyper parameters are:

*N\_estimators*: 700

*Max\_depth*: 7

*Max\_features*.- sqrt

*Random state.-*  35

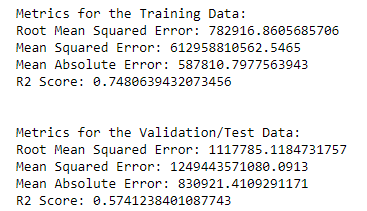
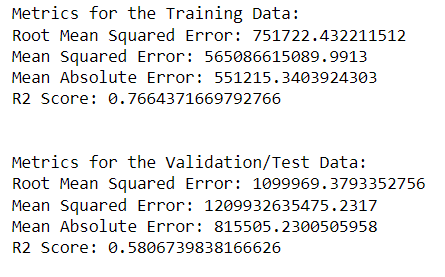


Fig. 14

The results of our metrics (Fig.14) shows that our training data has a score of 74%, however our test data has a score of 57%, this means that our data is over fitted.

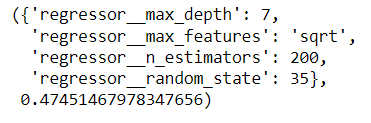
The results, with a test *size of 25*%, after tuning the hyper parameters are:



*Fig.15*

The results of our metrics (Fig.15) shows that our training data has a score of 76%, however our test data has a score of 58%, this means that our data is over fitted.

The results, with a test *size of 25*%, after tuning the hyper parameters are:



*Fig. 16*

*N\_estimators*: 200

*Max\_depth*: 7

*Max\_features*.- sqrt

*Random state.-*  35

The results, with a test *size of 30*%, after tuning the hyper parameters are:

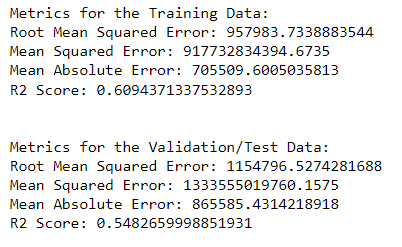


Fig. 17

*N\_estimators*: 600

*Max\_depth*: 5

*Max\_features*.- sqrt

*Random state.-*  60

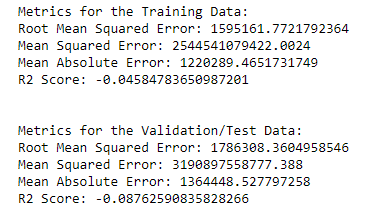
## Support Vector Regression

A linear regression calculates the closest 'fit' (typically the least squared) in the database. The best way of describing this fit is 'R2', which should be as high as possible, with the maximum being 1.00. For use as a forecasting tool, the average forecast Standard deviation is an indicator of usefulness.

This model is generally used to predict prices.

### Results

The results with a test *size of 20%* are:



*Fig. 18*

The results with a test *size of 25%* are:

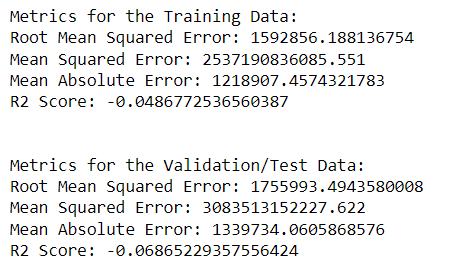


Fig.19

The results with a test *size of 30%* are:

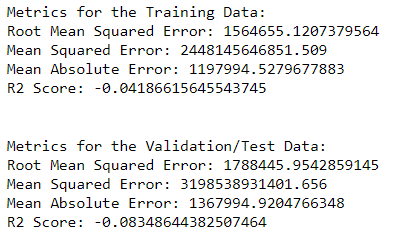


Fig.20

As we can see our result is negative, this suggest to find a new method that complies with the way SVR reads the data or we have to review again our data set.

### Results with Hypertuning

The results, with a test *size of 20*%, after tuning the hyper parameters are:

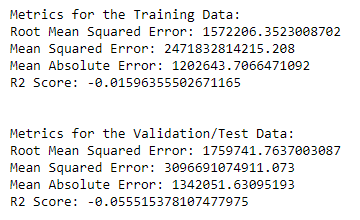
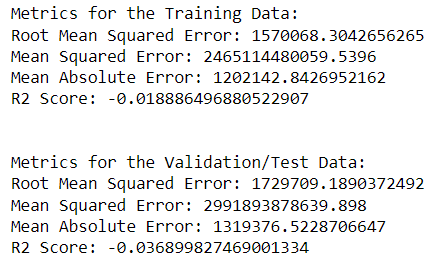
****

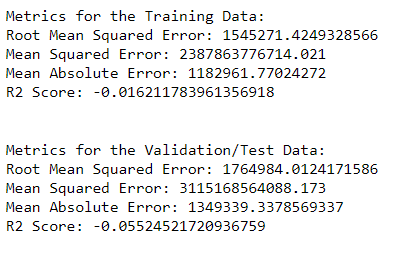
Fig. 18

The results, with a test *size of 25*%, after tuning the hyper parameters are:



*Fig.19*

The results, with a test *size of 30*%, after tuning the hyper parameters are:



*Fig.20*

### Hypertuning with feature importance SVR

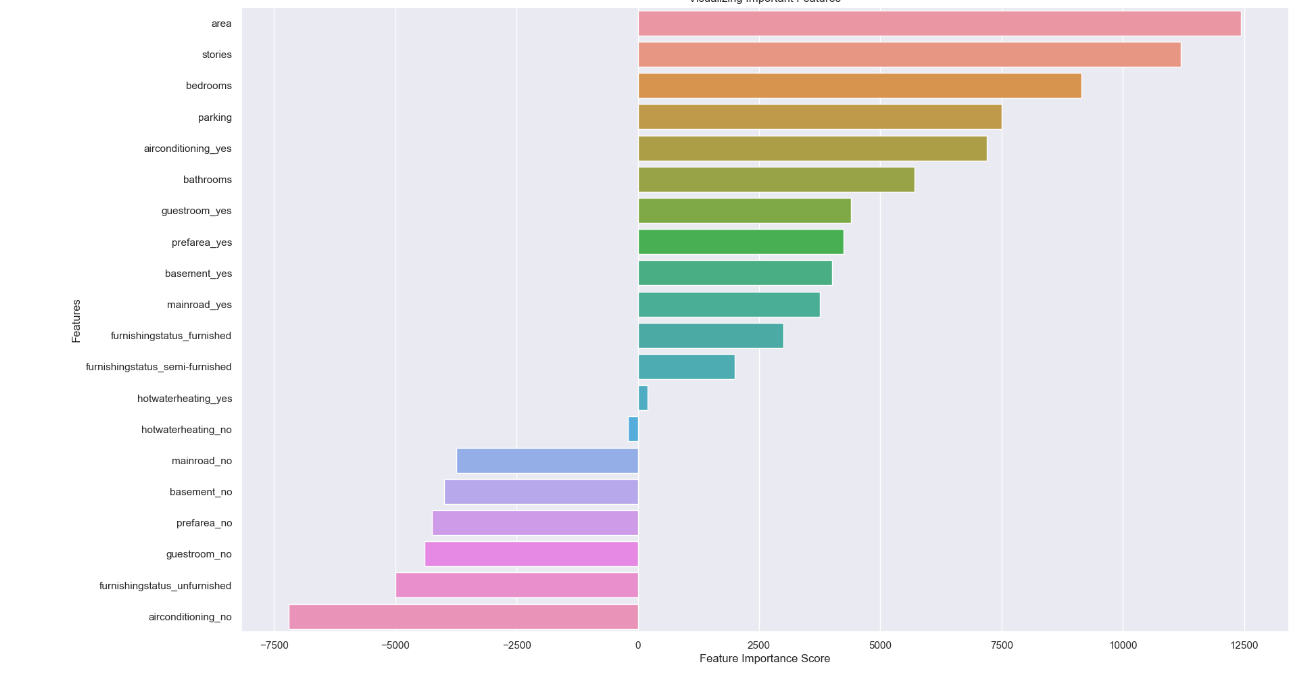
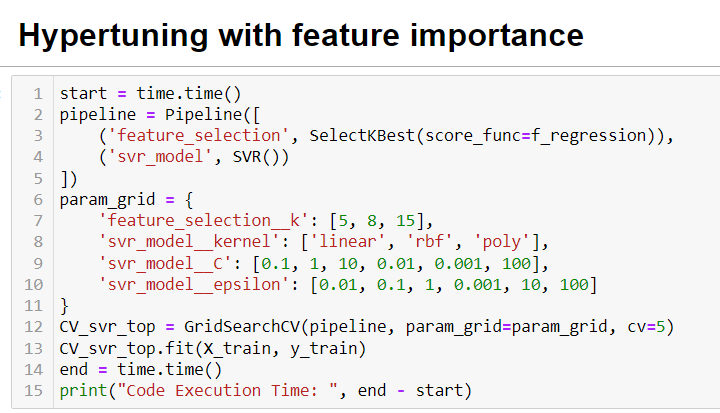


Fig 20.



# Hypertuning with feature importance

Hypertuning with feature importance 30%

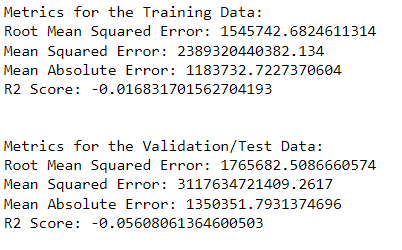
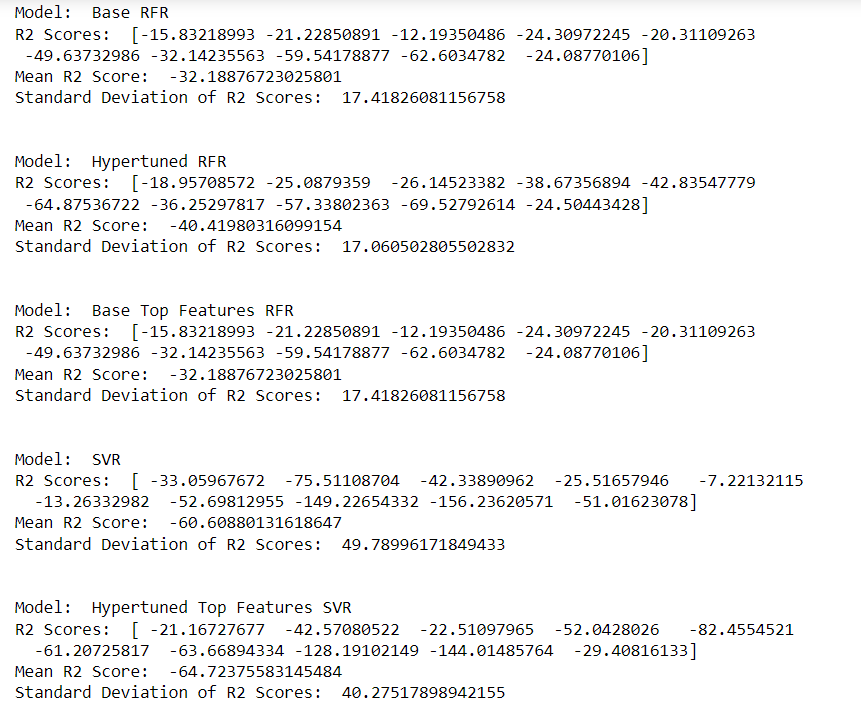


Fig.21

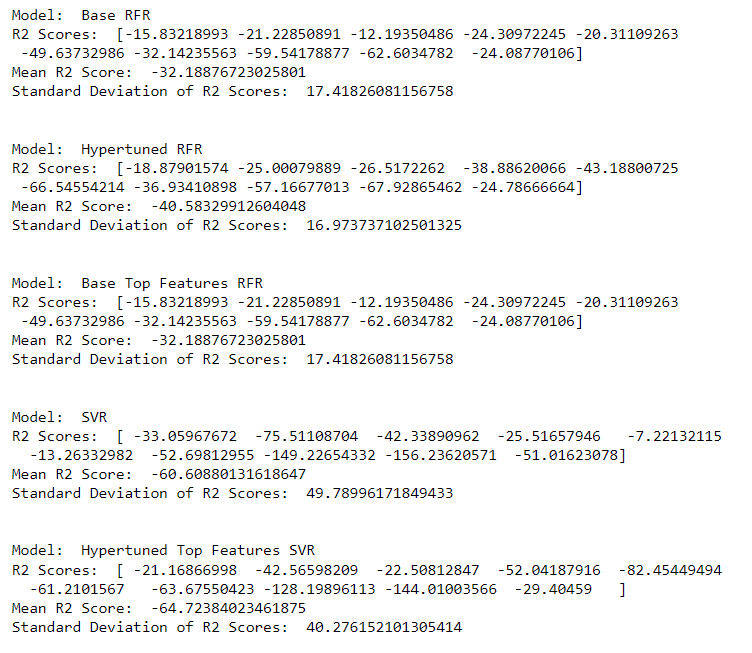
# Cross Validation

Cross validation based in a test size of 20%:

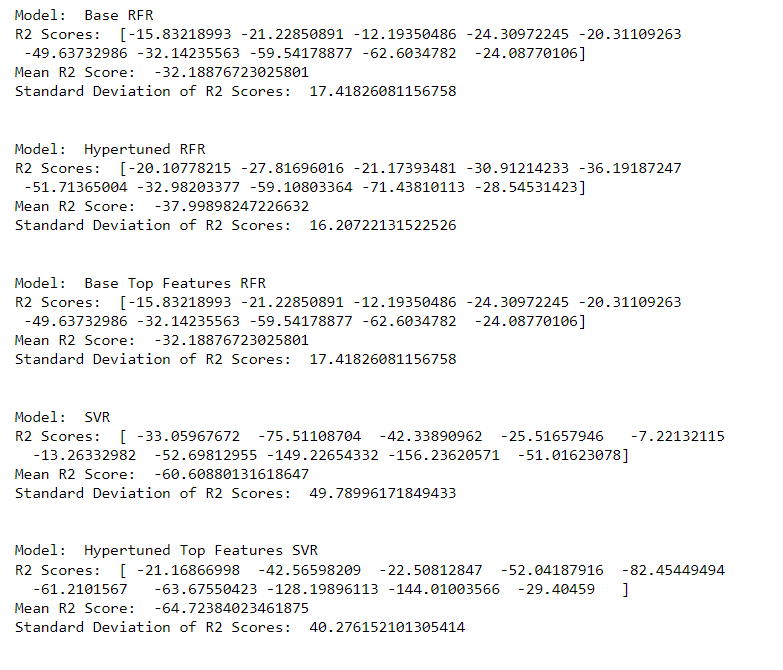


*Fig. 22*

Cross validation based in a test size of 25%:



Cross validation based in a test size of 30%:



# Conclusions

The student would need to consider the following instructions (a - d) during the development of this project.

a) Logical justification based on the reasoning for the specific choice of machine learning approaches.

b) Multiple machine learning models (at least two) using hyperparameters and a comparison between the chosen modelling approaches.

c) Visualise your comparison of ML modelling outcomes. You may use a statistical approach to argue that one feature is more important than other features.

d) Cross-validation methods should be used to justify the authenticity of your ML results.

3. What is the primary purpose of hyperparameter tuning in machine learning? Could you elaborate on specific hyperparameter tuning techniques (e.g., GridSearchCV) applied to machine learning models to find optimal parameters? (25 marks)

4. Interpret and explain the results obtained, discuss overfitting / underfitting / generalisation, provide a rationale for the chosen models and use visualisations to support your findings. Comments in Python code, conclusions of the project should be specified at the end of the report. Harvard Style must be used for citations and references. (25 marks)

● Clearly detail the number of words used in the report. (per section)

# Bibliography