7/15/2020

Feature Analysis on Covid-19 Dataset

Table of Contents

1. INTRODUCTION 1

2. "4" FEATURE SELECTION PROCESS 1

2.1 DELETION OF NON-NUMERICAL DATA 2
2.2 DELETION OF COLUMNS WITH HIGHER PERCENTAGE OF MISSING VALUES 2
2.3 DELETION OF NON-RELEVANT COLUMNS 3
2.4 DELETION OF FEATURES WITH LESS CORRELATION 4

3. NEW APPROACH FOR FEATURE ANALYSIS 4

3.1 UNIVARIATE AND BIVARIATE ANALYSIS 4
3.1.1 Histograms 4
3.1.2 Kernel Density Estimation 4
3.2 MISSING FEATURES ANALYSIS 4
3.3 IMPUTATION TECHNIQUES FOR MISSING FEATURES 5
3.3.1 Constant or 0 'zero' Imputation 5
3.3.2 Mean value Imputation 7
3.3.3 Simple Imputer function Imputation 9
3.3.4 Iterative Imputer function Imputation 10
3.3.5 Differences in Validations 12
3.4 CORRELATION ANALYSIS ON THE SELECTED DATASET 12
3.5 VALIDATION TESTING AFTER CORRELATION ANALYSIS 14

Table of Figures

Figure 1.....3
Figure 2.....6
Figure 3.....7
Figure 4.....8
Figure 5.....8
Figure 6.....9
Figure 7.....10
Figure 8.....11
Figure 9.....12
Figure 10.....13
Figure 11.....14
Figure 12.....15

1. Introduction

In this document we will understand how we have selected those 4 features for our regression model, and then we will be discussing about the analysis on features which includes univariate analysis, bivariate analysis, correlation between features and we will also discuss the techniques for missing values imputations.

2. "4" Feature Selection Process

The dataset for this project has been taken from "Our World in Data (Covid-19 Cases) ". The dataset has almost 26,000 entries representing data about 33 features for Covid-19 cases. There are sperate cases for each country up to 25th June, 2020. All features are given below:

- ISO code.
- Continent.
- Location (Country Name).
- Date.
- Total cases.
- Total deaths.
- New deaths.
- Total cases per million.
- New cases per million.
- Total deaths per million.
- New deaths per million.
- Total tests.
- New tests.
- Total tests per thousand.
- New tests per thousand.
- New tests smoothed.
- New tests smoothed per thousand.
- Test units.
- Stringency index.
- Population.
- Population density.
- Median age.
- Aged 65 older.
- Aged 70 older.
- GDP per capita.
- Extreme poverty.
- Cvd death rate.
- Diabetes prevalence.
- Female smokers.
- Male smokers.

- Handwashing facilities.
- Hospital beds per thousand.
- Life expectancy.

Out of all 33 features we had selected 4 features for our regression model, and after training our model on those features the model had given us R^2 score of 0.85. Selection of those 4 features has been defined in the steps below:

2.1 Deletion of Non-numerical data

In our dataset we have 5 features having non-numerical data.

- Iso code
- Location
- Continent
- Date

We had deleted those 5 features because of some reasons. In those non-numerical features, we have almost 5%-15% missing data. There are many techniques by which we can fill up those missing entries, but there is a problem with that, when we fill up the missing entries of non-numerical data then it creates bias in our dataset which is a bad sign for the accuracy of your model.

Another reason for deleting those columns is growing number of columns after using one-hot encoding. ML models work with numerical dataset, they don't work well with non-numerical dataset or sometimes it gives us error with non-numerical datasets. For resolving those issues there is a technique called as **One-Hot Encoding.** One-Hot encoding will read the column of non-numerical data and then it will find categories in it, for example: in our location data column we have names of countries so for One-Hot Encoding each country name represents a category like US is one category, UK is another, Spain is also another category and so on. It will make columns with those categories and then it will fill up the entries of those columns with 1 or 0 values. After applying One-Hot encoding the columns will increase. And we have 5 features so after applying One-Hot Encoding we will end up with having more than 650 columns in our dataset which are to much for regression model. In Classification models this is not a big deal but it effects the regression models.

2.2 Deletion of columns with higher percentage of missing values

After deleting those non-numerical features, we have left with 28 features. Now in those 28 features some of them have more than 30% missing values. We have almost 9 features which have more than 30% missing entries. After deleting those entries, we are left with 19 features. The percentage of missing values for each feature is given below:

```
Missing values in percentage:


iso_code                              0.244892
continent                             0.925997
location                              0.000000
date                                  0.000000
total_cases                           0.880080
total_deaths                          0.880080
new_deaths                            0.880080
total_cases_per_million               1.124971
new_cases_per_million                 1.124971
total_deaths_per_million              1.124971
new_deaths_per_million                1.124971
total_tests                          71.952246
new_tests                            74.504477
total_tests_per_thousand             71.952246
new_tests_per_thousand               74.504477
new_tests_smoothed                   69.767353
new_tests_smoothed_per_thousand      69.767353
tests_units                          67.379659
stringency_index                     20.054335
population                            0.244892
population_density                    4.331522
median_age                            9.608173
aged_65_older                        10.882375
aged_70_older                        10.071172
gdp_per_capita                       10.622178
extreme_poverty                      40.303819
cvd_death_rate                        9.550777
diabetes_prevalence                   6.573812
female_smokers                       27.439351
male_smokers                         28.273513
handwashing_facilities               59.340323
hospital_beds_per_thousand           17.261039
life_expectancy                       1.385169
dtype: float64
```

Figure 1

2.3 Deletion of non-relevant columns

Now we will be going to delete those columns which have less relevancy with the "rise in infection'. Some of the features like "total deaths per million", "new deaths per million" are irrelevant for our regression. These features are useful if we are building a regression model for deaths rates by Covid-19. Same is the case with some other features too like "stringency index", 'female smokers', "male smokers", "life expectancy", "gdp per capita", "total deaths", "new deaths" and after deleting these types of features we have left with 8 features.

2.4 Deletion of features with less correlation

Now on those 8 features we had applied correlation analysis which had given us the association of variables with each other. After the analysis of correlation data, we came up with 4 features.

3. New approach for Feature analysis

3.1 Univariate and Bivariate Analysis

First, we have done univariate and bivariate analysis on each and every feature in the dataset. **Univariate** analysis is the simplest form of analyzing data, it works with only one variable. It doesn't deal with causes or relationships and it's major purpose is to describe; It takes data, summarizes that data and finds patterns in the data. **Bivariate analysis** is one of the simplest forms of quantitative statistical analysis. It involves the analysis of two variables for the purpose of determining the empirical relationship between them. Bivariate analysis can be helpful in testing simple associations. Bivariate analysis can help determine to what extent it becomes easier to know and predict a value for one variable.

Univariate analysis of the data is done by the help of **Histograms** and **Kernel density Estimation.** Bivariate analysis involves histograms and scatter plots.

3.1.1 Histograms

A histogram is a plot that lets you discover, and show, the underlying frequency distribution of a set of continuous data. This allows the inspection of the data for its underlying distribution (e.g., normal distribution), outliers, skewness, etc.

3.1.2 Kernel Density Estimation

Kernel density estimation is a really useful statistical tool. It's a technique that let's you create a smooth curve given a set of data. Kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable. KDE is mostly used for numerical data, KDE is shown by the curve line on the graph.

3.2 Missing features Analysis

For missing features analysis, first we have calculated the percentage of missing values in the dataset. We have calculated the missing values percentage by the help of pandas library, pandas library has a function 'isnull()' which is mostly used for the detection of missing value in the dataset. It does not take any parameter. If we simply call that function it will return the number of missing entries for each feature. For calculating the percentage we had used the expression '.sum()*100/len(features)", by the help of this expression we will got the percentage of missing entries for each feature in the dataset.

After getting the percentage of missing values we will then removed the features which are having more than 30%. Now on remaining features we will be using different imputation techniques to fill up the missing data.

3.3 Imputation techniques for missing features

There exists are a lot of imputation techniques, each and every imputation technique has their own pros and cons. For this analysis we are using 4 different imputation techniques, beside these 4 there are bunch of other techniques which are available to use.

3.3.1 Constant or 0 'zero' Imputation

The first technique which we have used is constant or zero imputation technique. This technique is widely used for every type of dataset. In this imputation technique we fill up the numerical data with zero value, and for non-numerical value we have to select a constant value like 'Null', 'None" or you can also select any value of your choice. For filling the missing entries in the dataset we will use the function 'fillna()' from pandas library, this function requires a value which is to be replaced with the missing data entry. After applying this technique we have used one-hot encoding to setup the non-numerical data for training and validation and then we have validated the dataset with our regression model and we have got the following results:

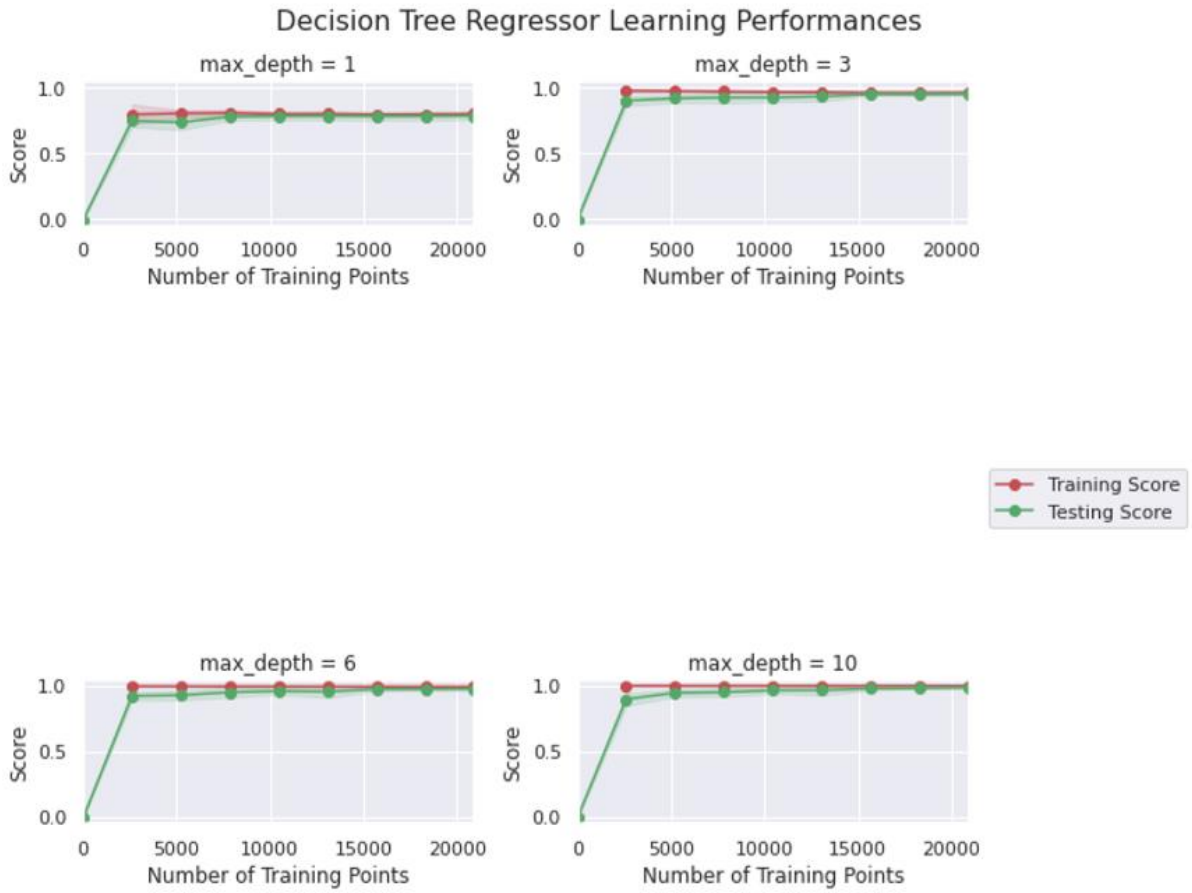
Decision Tree Regressor Learning Performances
max_depth = 1
max_depth = 3
max_depth = 6
max_depth = 10
Score
Number of Training Points
Training Score
Testing Score

Figure 2

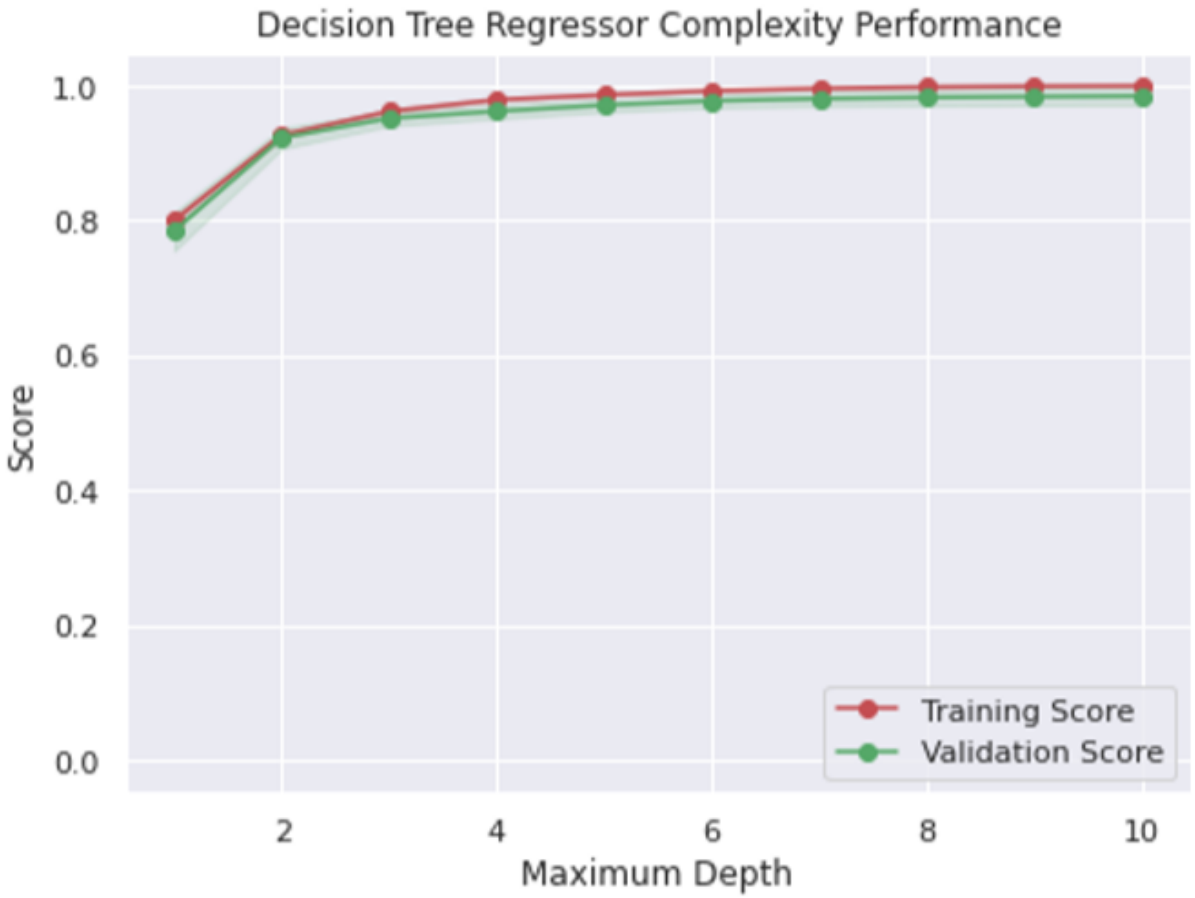

Figure 3

3.3.2 Mean value Imputation

In this technique we will be filling up the missing entries with mean values. For non-numerical data we will again use constant values as we re unable to find the mean value for the non-numerical data. For finding the mean value for each feature we will take help from pandas function ".mean()", this will find the mean values for each feature and replace the missing values with mean value of the column. After applying that imputation technique, we will again use one-hot encoding for setting up the non-numerical data for validation and testing. Results of validation after that technique are as follows:

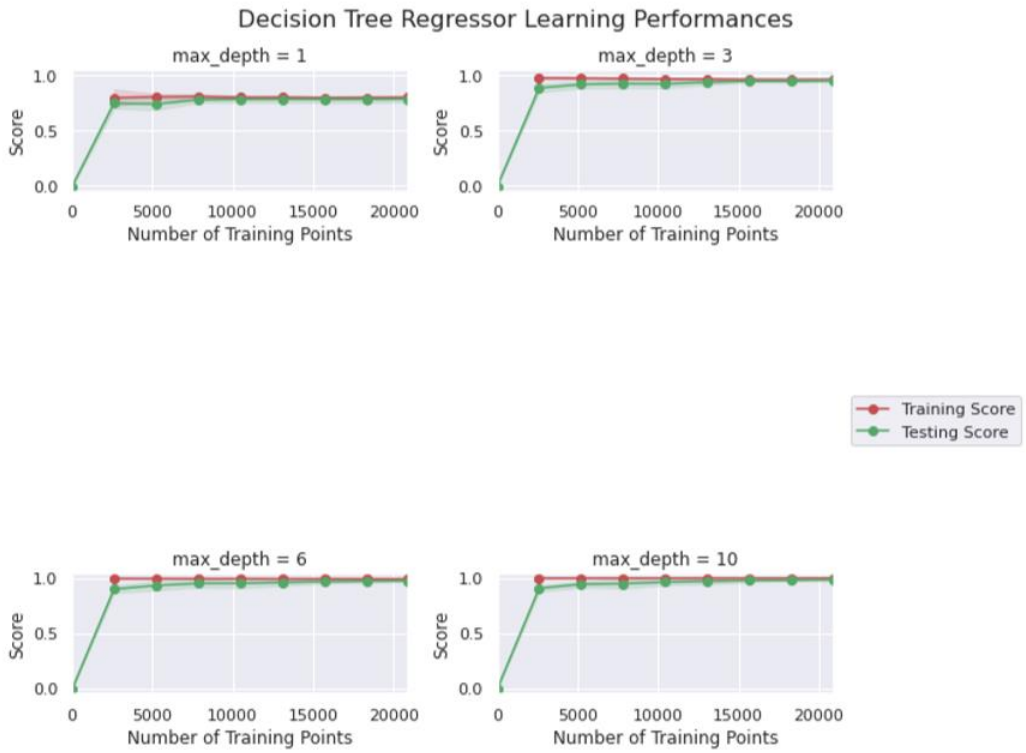

Figure 4

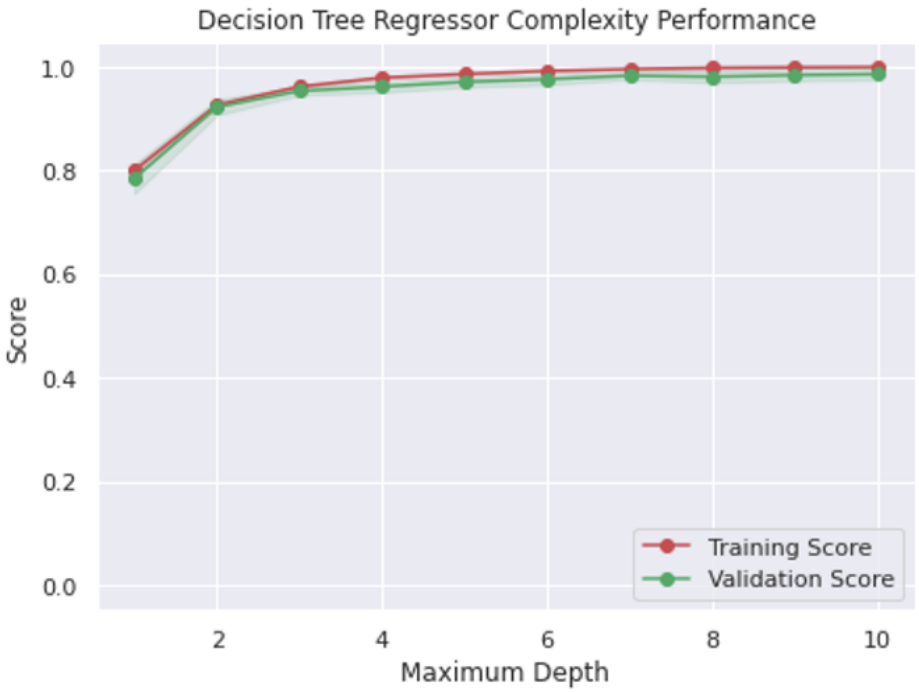

Figure 5

3.3.3 Simple Imputer function Imputation

In this technique we will be filling up the missing entries with the help of simple imputer function provided by sklearn library. This function is widely used for the simple imputation of missing data entries. In this function we have 3 different strategies. The one which we are using is the "most frequent" imputation technique. This will fill up the dataset of numerical and non-numerical dataset with the most frequent values. The results of validations for this technique are given below:

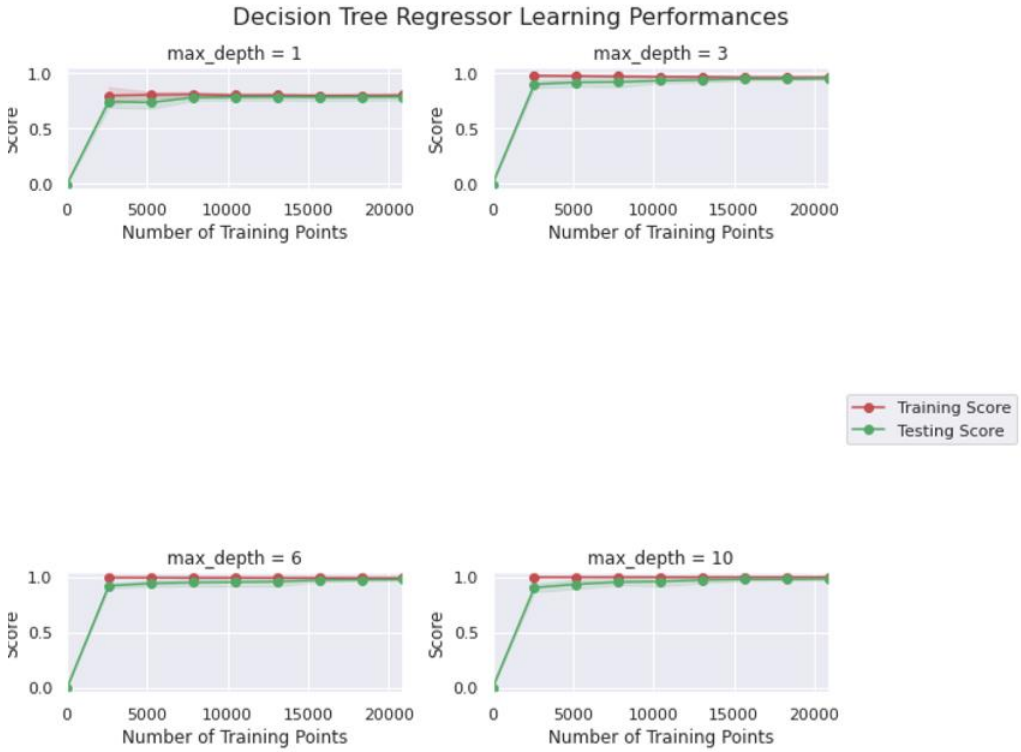

Figure 6

Figure 7

3.3.4 Iterative Imputer function Imputation

In this technique we will be filling up the missing entries with the help of iterative imputer function provided by sklearn library. This is a bivariate imputation technique and it is very much useful for numerical data. It is unable to impute the non-numerical data. This iterative imputer takes two parameters for the imputation process to begin with. The first is the "max iter" and the second one is "random state", max iter tells the imputer that how many iterative analysis it will fill up the missing entries, and the second parameter helps us in the randomization of new fillable values so they all won't be same at any point. The results after that imputations are:

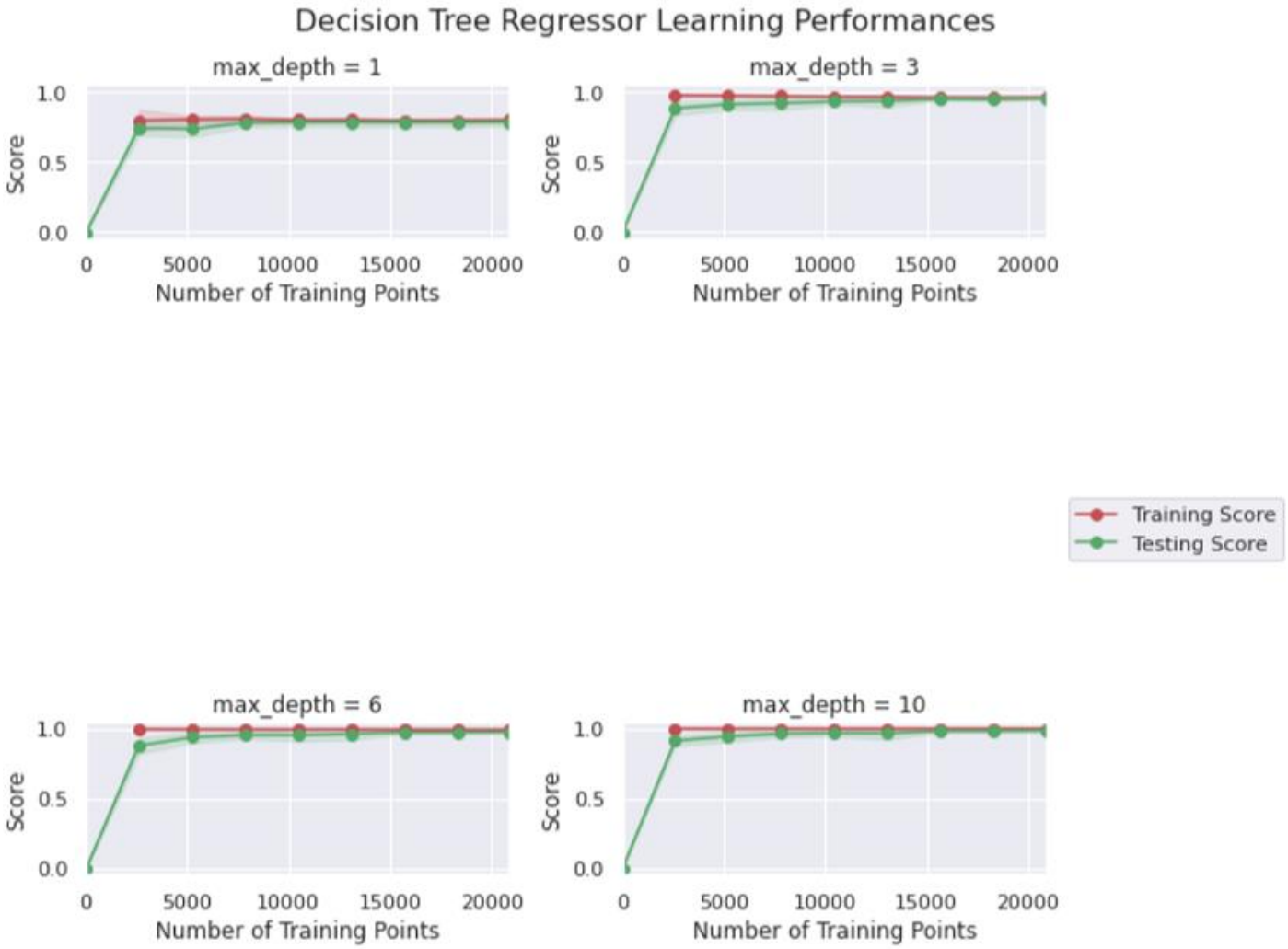
Decision Tree Regressor Learning Performances
max_depth = 1
max_depth = 3
max_depth = 6
max_depth = 10
Score
Number of Training Points
0.0
0.5
1.0
0
5000
10000
15000
20000
Training Score
Testing Score

Figure 8

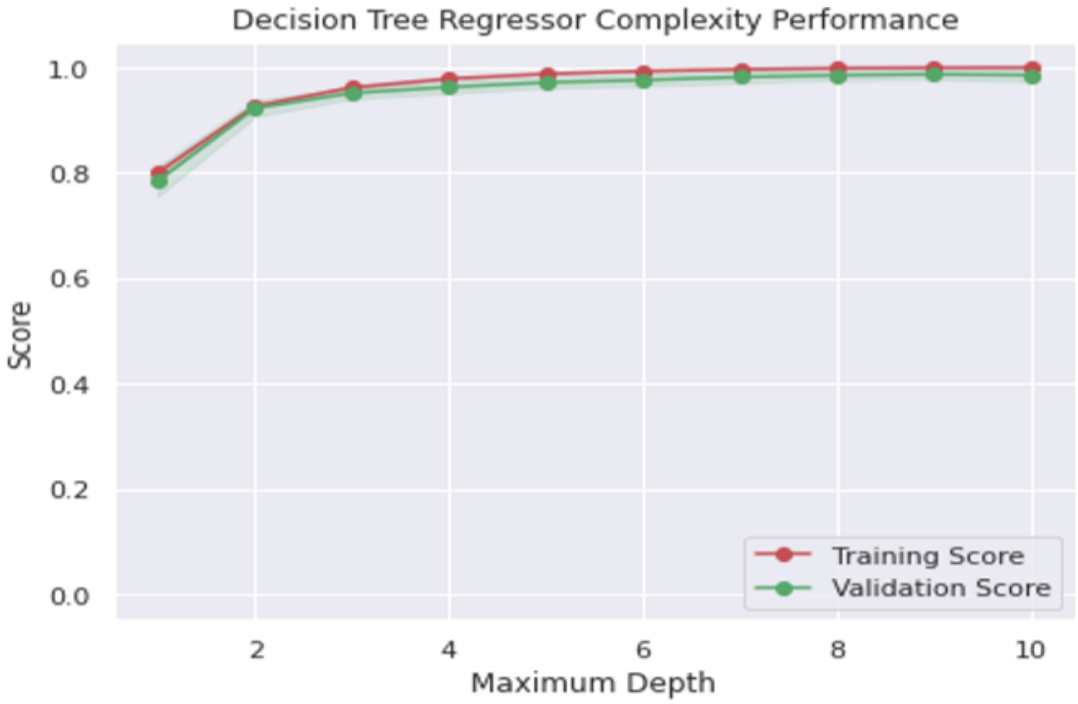

Figure 9

3.3.5 Differences in Validations

For this dataset we have applied imputation techniques and validated those techniques, you will find a slight change or almost no change in the validation graphs of different techniques this is because we had a small number of missing values and filling them not effect the large part of dataset that's why we have a slight change in the validation graphs.

3.4 Correlation analysis on the selected dataset

Correlation analysis is use to find the associations between variables.The correlation coefficient is measured on a scale that varies from + 1 through 0 to - 1. Complete correlation between two variables is expressed by either + 1 or -1. When one variable increases as the other increases the correlation is positive; when one decreases as the other increases it is negative. Complete absence of correlation is represented by 0.

For correlation analysis we are taking the imputed dataset by Iterative Imputer. We are using pandas to calculate the correlation of a dataset, pandas function .corr() is used to calculate the correlation between features. There are 3 methods which can be used to calculate the correlation between features. We are using a standard one Pearson method, other two methods are "kendall" and "spearman".

The heatmap generated by the correlation analysis of the dataset is given below.

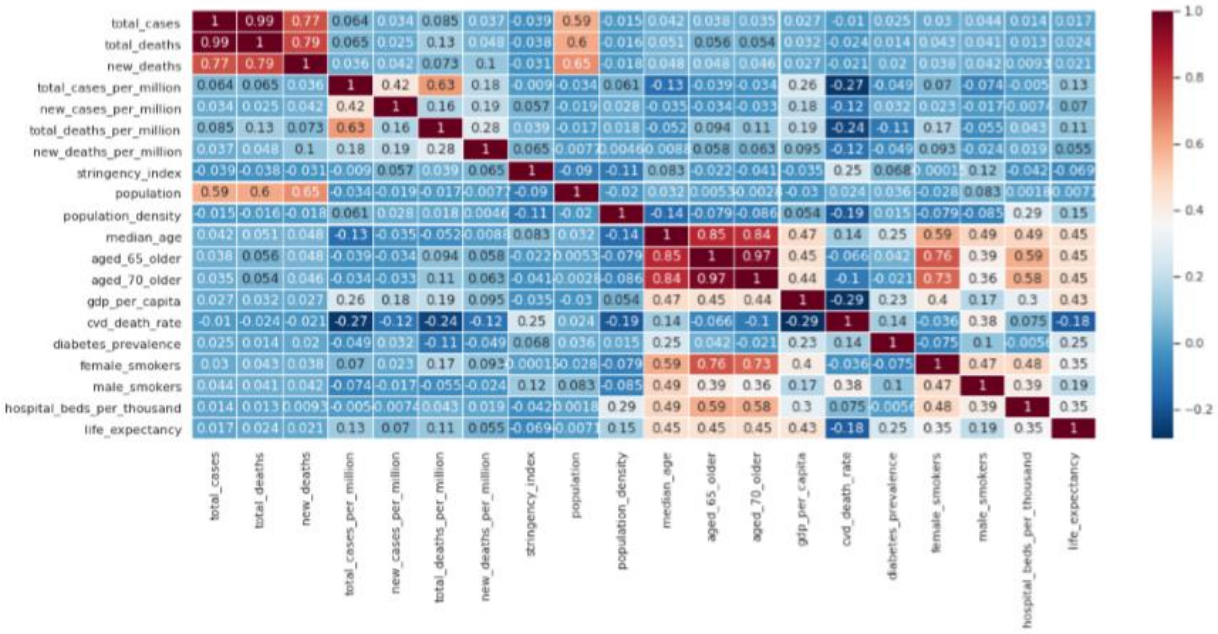

Figure 10

By the help of this heatmap we can easily visualize the correlation of the dataset. After applying the correlation analysis we found 5 features which have good associations, these 5 features are different from the previous 4 we have selected. The 5 selected features are:

- Total cases
- Total deaths
- New deaths
- Population
- Median age

3.5 Validation testing after correlation analysis

Now we will be doing validation testing on the selected 5 features. The validation graphs are given below:

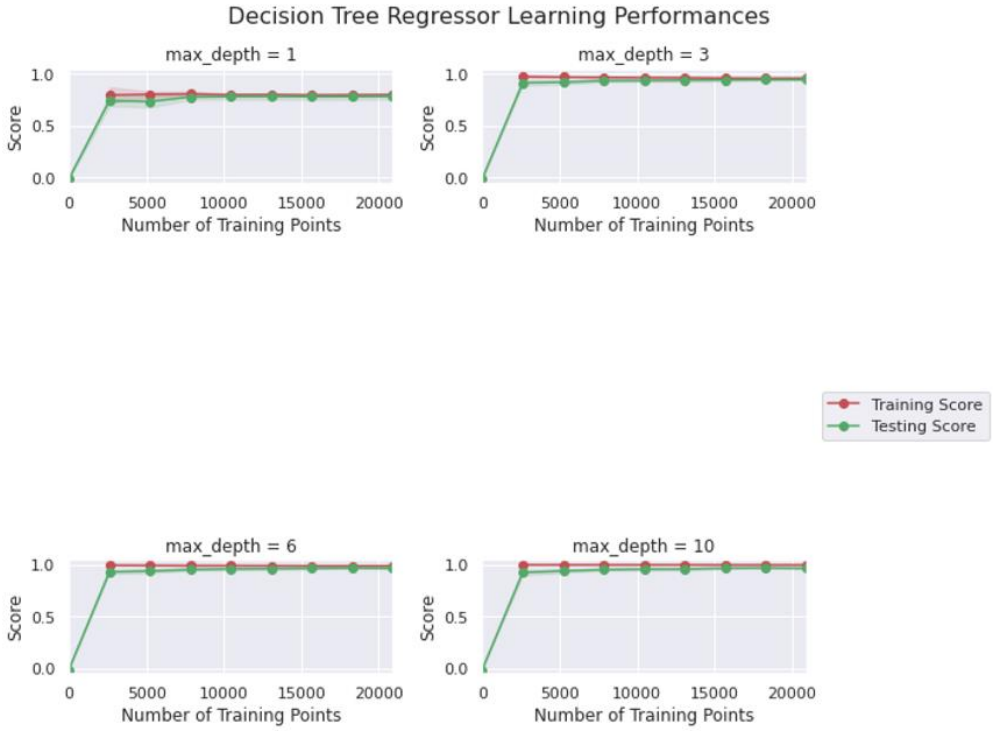

Figure 11

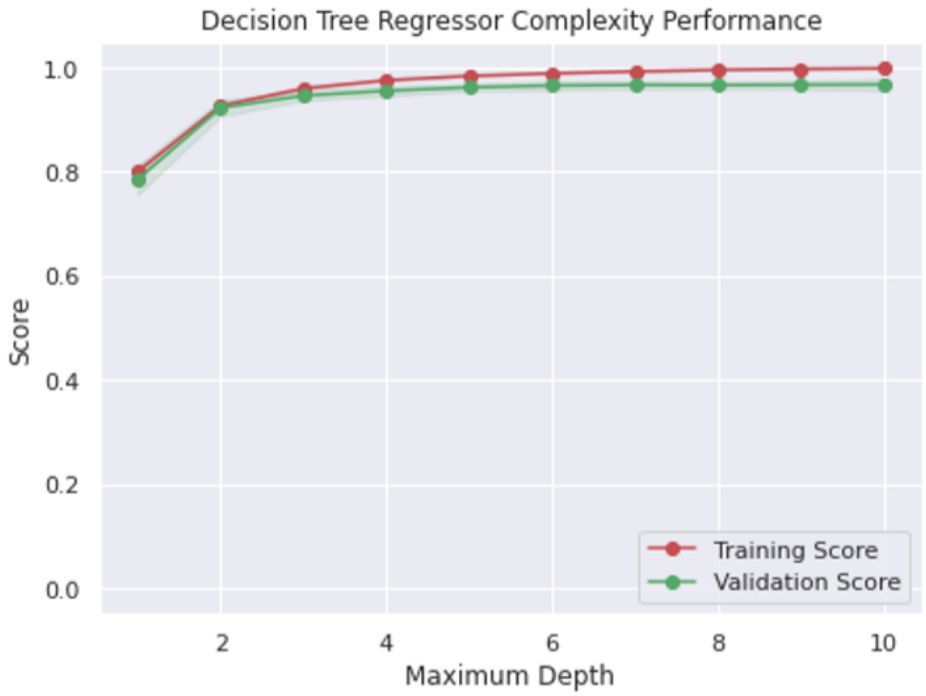
Decision Tree Regressor Complexity Performance
1.0
0.8
0.6
0.4
0.2
0.0
Score
2
4
6
8
10
Maximum Depth
Training Score
Validation Score

Figure 12