Linear Regression

What is Linear Regression?

Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things:

1. Does a set of predictor variables do a good job in predicting an outcome (dependent) variable?
2. Which variables in particular are significant predictors of the outcome variable, and in what way do they–indicated by the magnitude and sign of the beta estimates–impact the outcome variable?

These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables. The simplest form of the regression equation with one dependent and one independent variable is defined by the formula:

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

and by extension

$$y_i = \beta_0 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

Notations

- **y** is a vector of observed values **yi (i=1,.....,n)** of the variable called the regressand, endogenous variable, response variable, measured variable, criterion variable, or dependent variable. This variable is also sometimes known as the predicted variable, but this should not be confused with predicted values, which are denoted $\hat{y}$**.**
- **X** may be seen as a matrix of row-vectors **Xi** or of n-dimensional column-vectors **Xj** which are known as regressors, exogenous variables, explanatory variables, covariates, input variables, predictor variables, or independent variables (not to be confused with the concept of independent random variables). The matrix **X** is sometimes called the design matrix.
- **β** is a **(p+1)** dimensional parameter vector where **β0** is the intercept term if one is included in the model—otherwise **β** is *p*-dimensional).
- **ε is a vector of values**

Naming the Variables

There are many names for a regression's dependent variable. It may be called an outcome variable, criterion variable, endogenous variable, or regressand. The independent variables can be called exogenous variables, predictor variables, or regressors.

Advantages

Three major uses for regression analysis are:

1. determining the strength of predictors,
2. forecasting an effect
3. trend forecasting

First, the regression might be used to identify the strength of the effect that the independent variable(s) have on a dependent variable. Typical questions are what is the strength of relationship between dose and effect, sales and marketing spending, or age and income.

Second, it can be used to forecast effects or impact of changes. That is, the regression analysis helps us to understand how much the dependent variable changes with a change in one or more independent variables. A typical question is, "how much additional sales income do I get for each additional $1000 spent on marketing?"

Third, regression analysis predicts trends and future values. The regression analysis can be used to get point estimates. A typical question is, "what will the price of gold be in 6 months?"

Decision Trees

What is a decision tree?

A **decision tree** is a supervised machine learning model used to predict a target by learning decision rules from features. As the name suggests, we can think

of this model as breaking down our data by making a decision based on asking a series of questions.

Based on the features in our training set, the decision tree model learns a series of questions to infer the class labels of the samples. As we can see, decision trees are attractive models if we care about interpretability.Although the preceding figure illustrates the concept of a decision tree based on categorical targets (**classification**), the same concept applies if our targets are real numbers (**regression**).

Decision Trees Regression

A decision tree is constructed by **recursive partitioning** — starting from the root node (known as the first **parent**), each node can be split into left and right **child** nodes. These nodes can then be further split and they themselves become parent nodes of their resulting children nodes. Starting from the root, the data is split on the feature that results in the largest **Information Gain** (**IG**). In an iterative process, we then repeat this splitting procedure at each **child node** until the leaves are pure — i.e. samples at each node all belong to the same class.

In order to split the nodes at the most informative features, we need to define an objective function that we want to optimize via the tree learning algorithm. Here, our objective function is to maximize the information gain at each split, which we define as follows:

$$IG(D_p, f) = I(D_p) - \left(\frac{N_{left}}{N_p} I(D_{left}) + \frac{N_{right}}{N_p} I(D_{right})\right)$$

Here, *f* is the feature to perform the split, *Dp*, *Dleft*, and *Dright* are the datasets of the parent and child nodes, *I* is the **impurity measure**, *Np* is the total number of samples at the parent node, and *Nleft* and *Nright* are the number of samples in the child nodes.

To use a decision tree for regression, we need an impurity metric that is suitable for continuous variables, so we define the impurity measure using the **weighted mean squared error** (**MSE**) or **mean absolute error (MAE)** of the children nodes:

$$\mathrm{MSE}(t) = \frac{1}{N_t} \sum_{i \in D_t} (y^{(i)} - \hat{y}_t)^2 \qquad \mathbf{MAE} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{n}} |\boldsymbol{y_j} - \hat{\boldsymbol{y}}_{\boldsymbol{j}}|$$

Advantages and Disadvantages

- Easy to use and understand.
- Can handle both categorical and numerical data.
- Resistant to outliers, hence require little data preprocessing.
- New features can be easily added. Can be used to build larger classifiers by using ensemble methods.

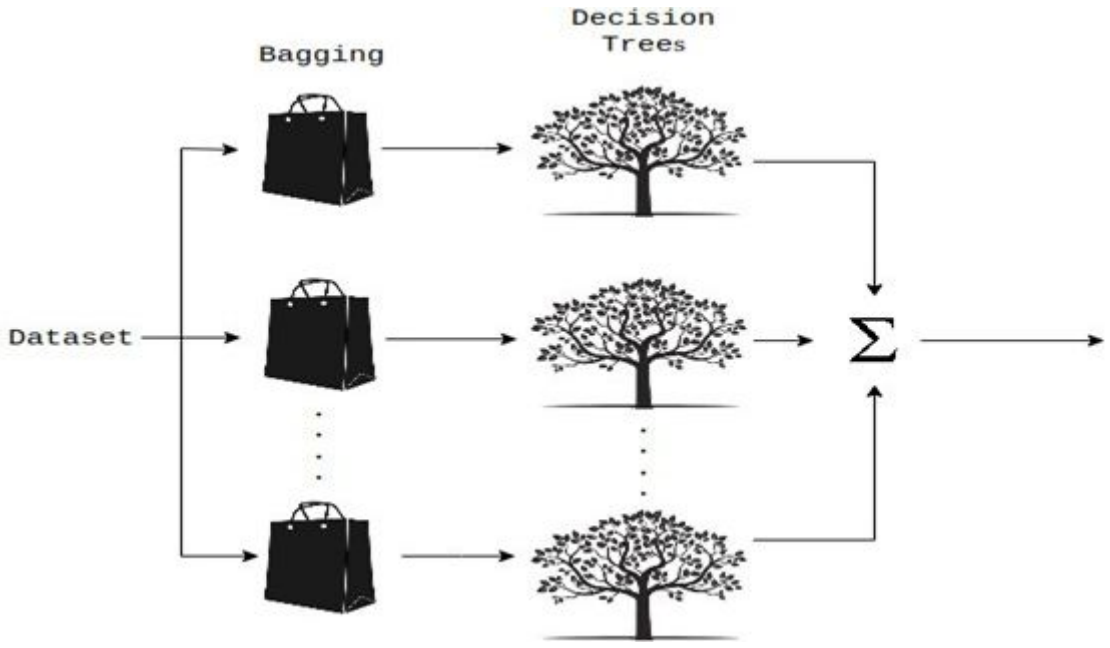
Disadvantages of decision trees:

- Prone to overfitting.
- Require some kind of measurement as to how well they are doing.
- Need to be careful with parameter tuning.
- Can create biased learned trees if some classes dominate.

Random forest

What is a Random forest?

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging. Bagging, in the Random Forest method, involves training each decision tree on a different data sample where sampling is done with replacement.



The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.
In the random forest, we grow multiple trees in a model. To classify a new object based on new attributes each tree gives a classification and we say that tree votes for that class. The forest chooses the classifications having the most votes of all the other trees in the forest and takes the average difference from the output of different trees. In general, Random Forest builds multiple trees and combines them together to get a more accurate result.

Why choose random forests?

Different kinds of models have different advantages. The random forest model is very good at handling tabular data with numerical features, or categorical features with fewer than hundreds of categories. Unlike linear models, random forests are able to capture non-linear interaction between the features and the target.

One important note is that tree based models are not designed to work with very sparse features. When dealing with sparse input data (e.g. categorical features with large dimension), we can either pre-process the sparse features to generate numerical statistics, or switch to a linear model, which is better suited for such scenarios.

Advantages:

1. Can be used for both classification and regression problems: Random Forest works well when you have both categorical and numerical features.
2. Reduction in overfitting: by averaging several trees, there is a significantly lower risk of overfitting.
3. Make a wrong prediction only when more than half of the base classifiers are wrong: Random Forest is very stable - even if a new data point is introduced in the dataset, the overall algorithm is not affected much as new data may impact one tree, but it is very hard for it to impact all the trees.

Disadvantages:

1. Random forests have been observed to overfit for some datasets with noisy classification/regression tasks.
2. More complex and computationally expensive than decision tree algorithm.

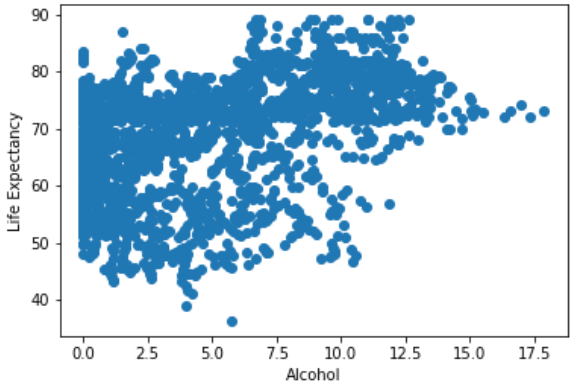
3. Due to their complexity, they require much more time to train than other comparable algorithms.

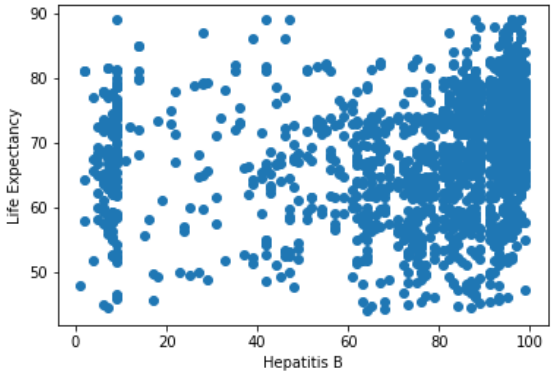
The Dataset

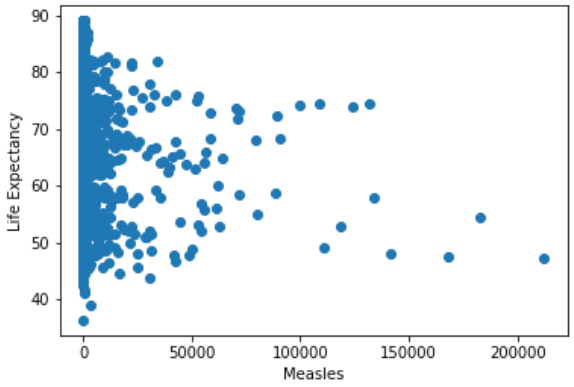
The dataset chosen from Kaggle is the Life Expectancy Data that was used by applying the models on it. The dataset contains columns “Country”, “Year”, “Status”, “Life Expectancy”, “Adult Mortality”, “Infant deaths”, “Alcohol”, “Percentage Expenditure”, "Hepatitis B", "Measles ", " BMI ", “Under-five deaths”, “Polio”, “Total expenditure”, “Diphtheria”, “HIV/AIDS”, “GDP”, “Population”, “Thinness 1-19 years”, “Thinness 5-9 years”, “Income”, “Schooling” from which the columns that were used are "Status", "Life expectancy ", "Adult Mortality", "Alcohol", "Hepatitis B", "Measles ", " BMI ", "Polio", "Total expenditure", "Diphtheria ", "GDP". After reading the data, the “null” (“NaN”) values were searched and replaced with the mean of the others, so they would not affect in a negative way when the prediction will be made.

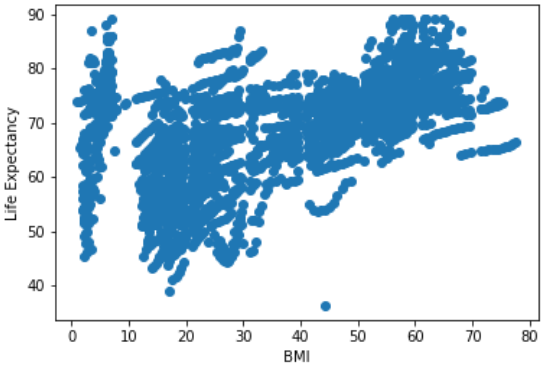
The data was split into the features and labels and then the process of normalization of the data was made by handling the categorical columns, which meant that the values were replaced by numerical values (e.g 0, 1) and the old columns were replaced by these ones. Also, in the process of normalization, the numbers were divided by 100 (in case of percentage), 1000 (in case of per 1000 population) or min-max normalization for the other values, so the numbers that were to work with were smaller. The polynomial transformer was used for the better capture of the patterns in the data. After that, the data was split between test and train and used.

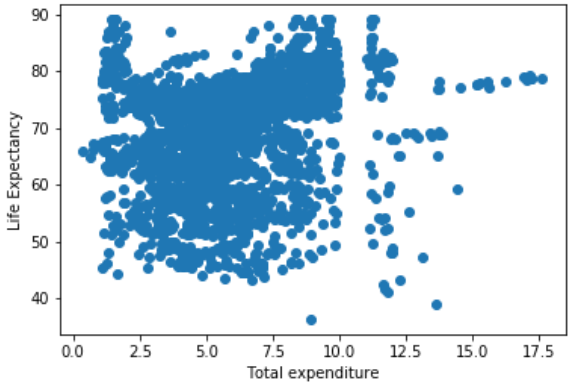
Dataset Plots

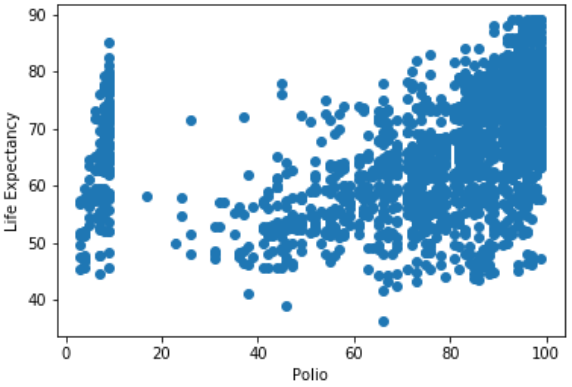


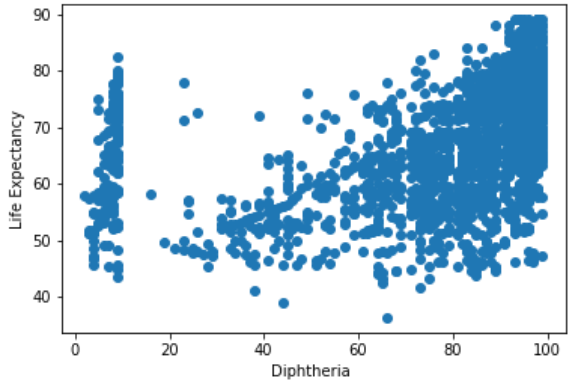


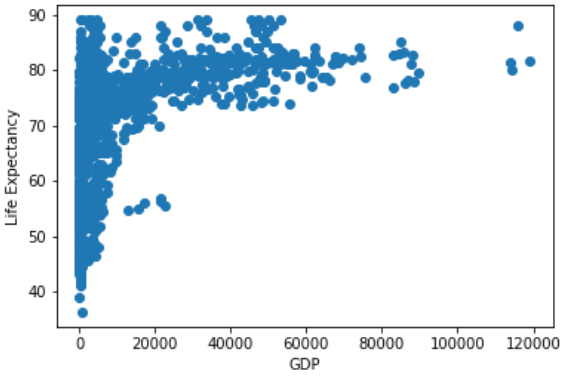












Models

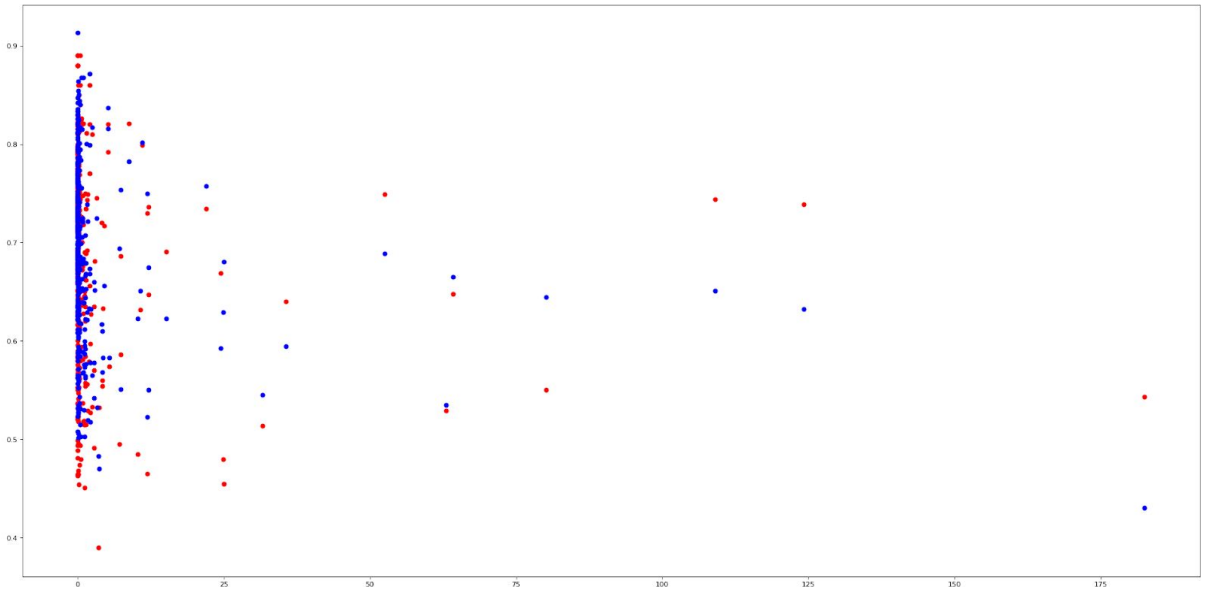
Linear Regression

Model

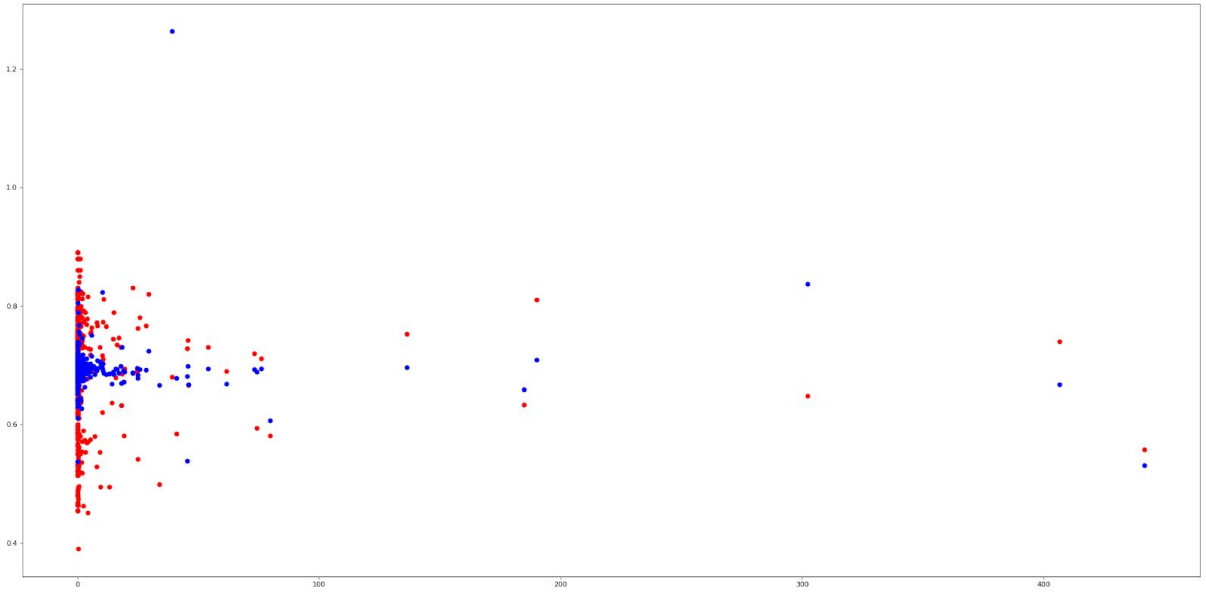
The model that was used for linear regression is the model from “sklearn” with the default parameters.

Results

1. Normal Data
    a. Mean Squared Error (MSE): 0.002789
    b. Variance: 0.710817



2. Polynomial Data
    a. Mean Squared Error (MSE): 0.010913
    b. Variance: -0.131255



Decision Trees

Model

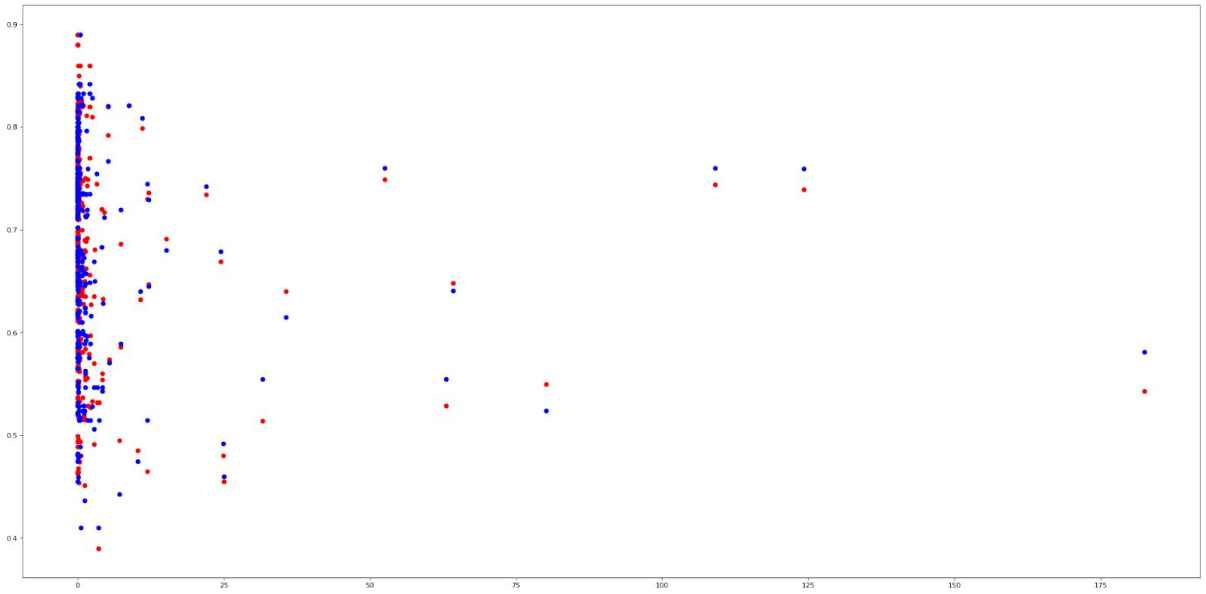
For the Decision Tree model was chosen by running the training and testing on multiple variations of Decision Tree parameters and looking at MSE, Score (Variance) and plots of data (RED - Expected Data, BLUE - Predicted Data).

The list of parameters that was used is the following:

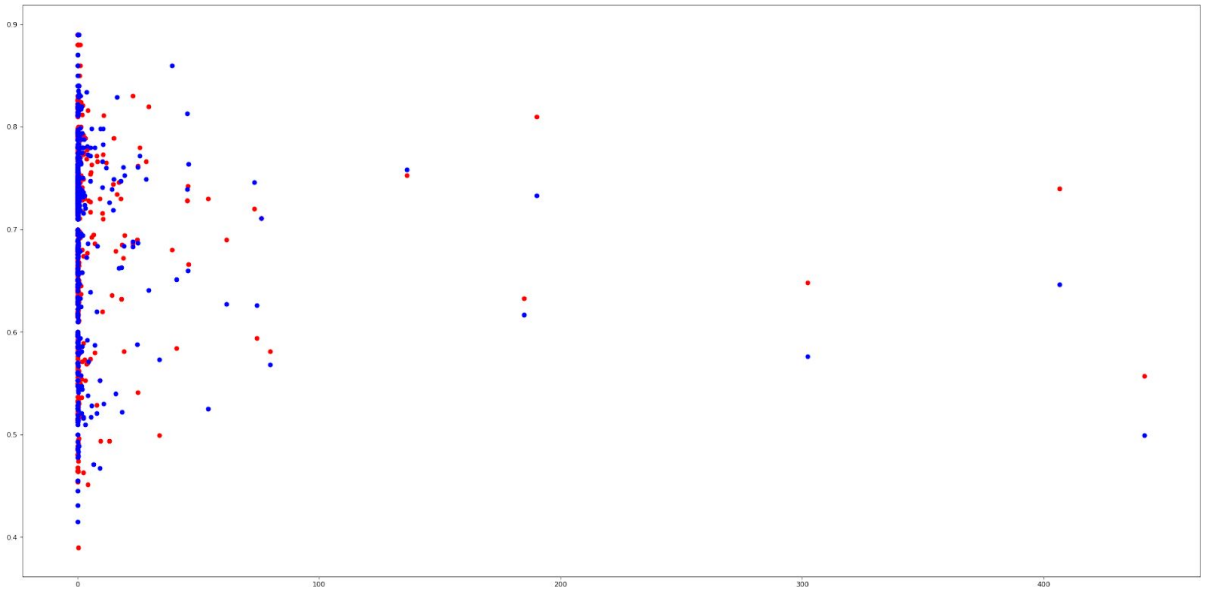
1. {}
2. {"max_depth" : 2}
3. {"max_depth" : 5}
4. {"max_depth" : 7}
5. {"max_depth" : 10}
6. {'criterion': "friedman_mse"}
7. {'criterion': "friedman_mse", "max_depth" : 2}
8. {'criterion': "friedman_mse", "max_depth" : 5}
9. {'criterion': "friedman_mse", "max_depth" : 7}
10. {'criterion': "friedman_mse", "max_depth" : 10}
11. {'criterion': "mae"}
12. {'criterion': "mae", "max_depth" : 2}
13. {'criterion': "mae", "max_depth" : 5}
14. {'criterion': "mae", "max_depth" : 7}
15. {'criterion': "mae", "max_depth" : 10}

Results

1. Normal Data - Best Model: {"max_depth" : 10} (5)
    a. Mean Squared Error (MSE): 0.000770
    b. Variance: 0.920171



2. Polynomial Data - Best Mode: {'criterion': "mae"} (11)
    a. Mean Squared Error (MSE): 0.017250
    b. Variance: -0.788134



Random Forest

Model

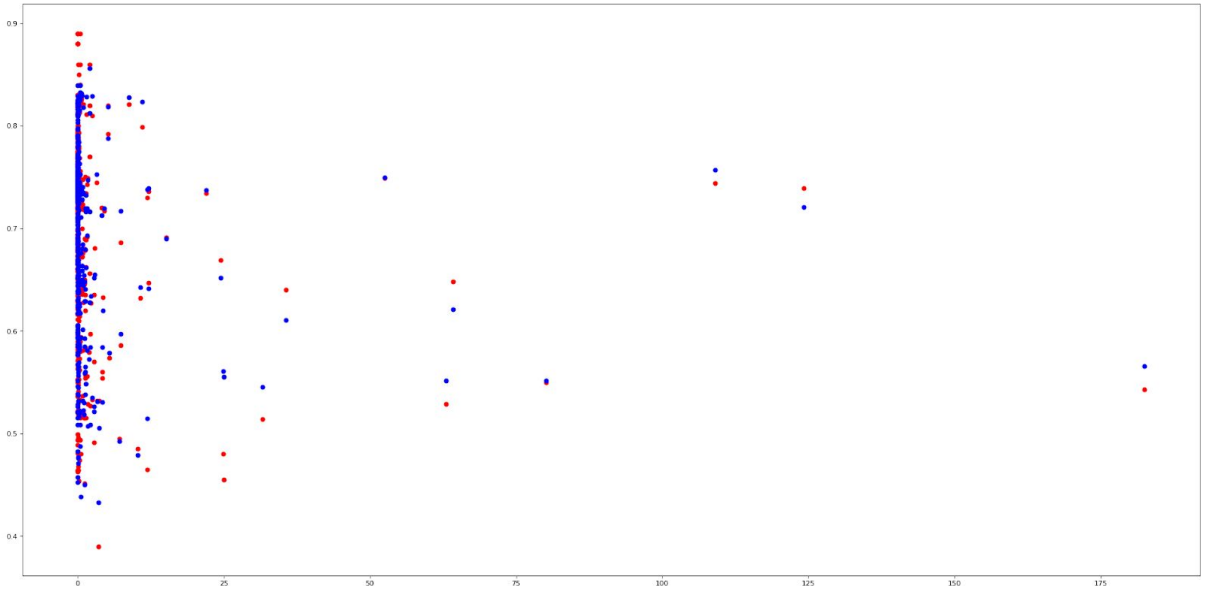
The Random Forest model was chosen by running the training and testing on multiple variations of parameters and looking at MSE, Score (Variance) and plots of data (RED - Expected Data, BLUE - Predicted Data).

The list of parameters that was used is the following:

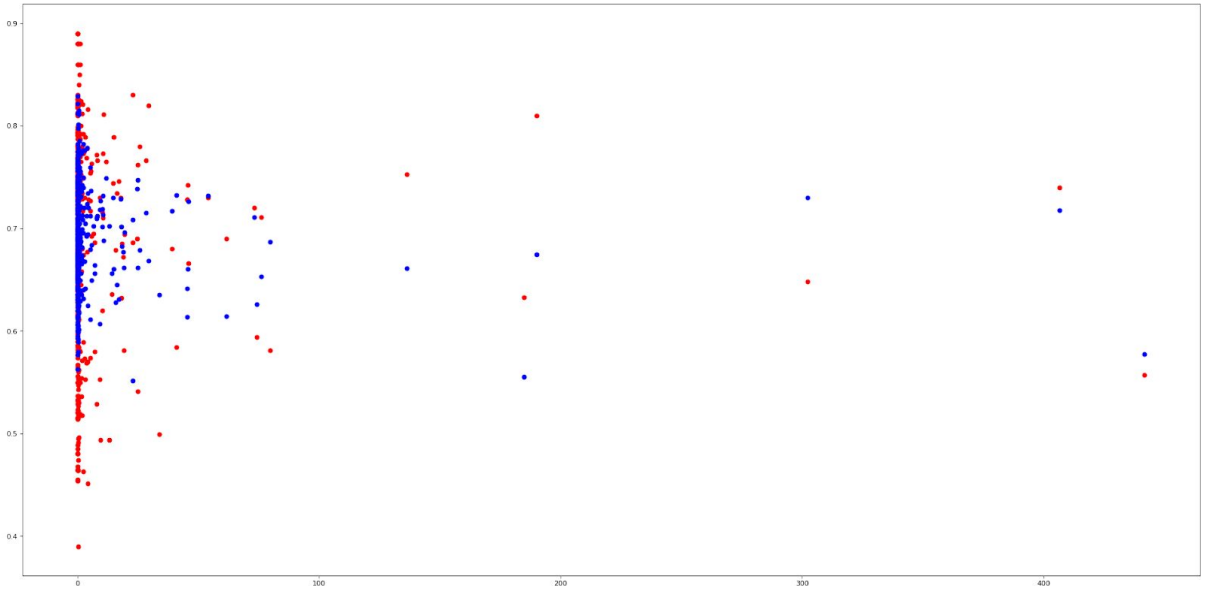
1. {}
2. {"n_estimators" : 5}
3. {"n_estimators" : 20}
4. {"n_estimators" : 50}
5. {"n_estimators" : 100}
6. {"n_estimators" : 150}
7. {"n_estimators" : 200}
8. {"criterion": "mae"}
9. {"criterion": "mae", "n_estimators" : 5}
10. {"criterion": "mae", "n_estimators" : 20}
11. {"criterion": "mae", "n_estimators" : 50}
12. {"criterion": "mae", "n_estimators" : 100}
13. {"criterion": "mae", "n_estimators" : 150}
14. {"criterion": "mae", "n_estimators" : 200}

Results

1. Normal Data - Best Model - {"n_estimators" : 200} (7)
    a. Mean Squared Error (MSE): 0.000628
    b. Variance: 0.934872



2. Polynomial Data - Best Model - {"n_estimators" : 5} (2)
    a. Mean Squared Error (MSE): 0.012011
    b. Variance: -0.245126



Conclusions

After analyzing the results we can notice that we can predict pretty well the life expectancy of humans according to different parameters.

According to the results, we can conclude that it is better not to use polynomial transformers as it seems to restrict the predictions in a smaller area on Y axis.

The best results were obtained from using Decision Tree and Random Forest algorithms as there seems to be not a big difference between their results.