CENG 474

Data Science Project

Insurance Cost Prediction

Berkay Demir
201911022
Buğra Mutluer
201911040
Ahmet Berkay Aslan
201911009

Instructor: Nurdan Saran

Introduction

Chosen the Medical Insurance data set from kaggle. It has 6 features and one output which is the Insurance cost. Features are: age, sex, bmi, children, smoker, region and output feature is charge. There are four regions which are southwest, southeast, northwest, northeast. Goal was to predict the Medical Insurance Cost from the given 6 features. There are a total of 1328 datas. the %20 percent of the people are smokers.

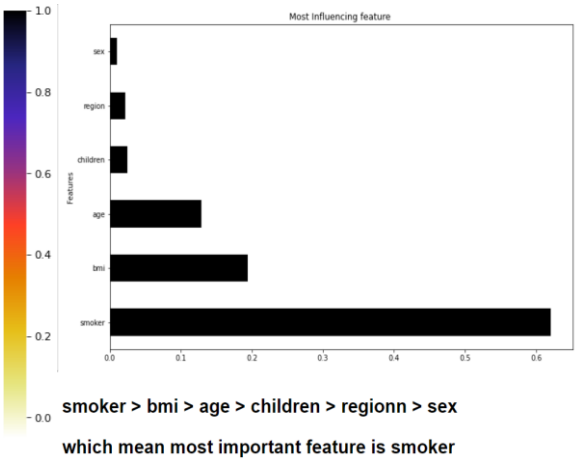
Cleaning The Data

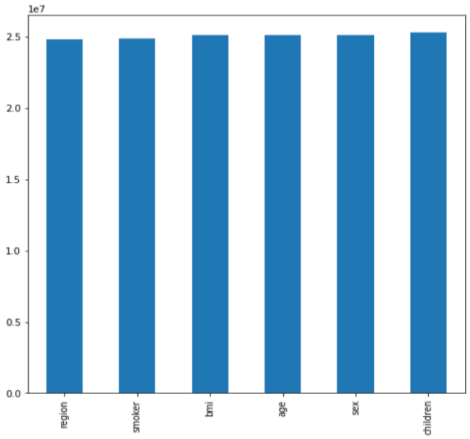
1- To clean the data, detect all of the rows that contain null values and deleted. There were a total of 9 null datas.
2- To clean the data, detected all of the duplicated rows and deleted the first occurrence of the duplicate row. There were only 1 duplicated data.
3- To clean the data, detect all of the values that lie outside of the values in a set. Detected some outliers but did not remove them since they were possible to occur. There were a total of 29 outliers.

Feature Engineering

1- Converted all the categorical variables to numeric variables which are sex, region smoker.
2- a) Using Pearson's Correlation, Measured the strength of the linear relationship between two features.
b) There were no features that had a correlation more than 0.7 between them. So did not remove a feature.
3- Found the most significant Features using **Extratreesregressor**. The most significant features are:
smoker > bmi > age > children > region > sex.
4- Using Mean Squared Error, found the average squared difference between the estimated values and the actual value. According to the Mean Squared Error the most significant features are:
smoker > bmi > age > children > sex > region.

Conclusion Of Feature Engineering: Checked the R-Square again after removing 3 features. Since the R-Square decreased. Did not remove any feature.





Regression

Before removing least significant 3 features:
The R-square of the linear regression is: **0.75066975385555**
The R-square of the polynomial regression is: **0.8148616339495293**
After removing least significant 3 features:
The R-square of the linear regression is: **0.7476294541623101**
The R-square of the polynomial regression is: **0.8389222946229109**

1- Used Linear Regression, divided the data to train and test sets in order to find a linear equation that best describes the data to make accurate predictions.

LinearRegression COST PREDICTION

```
prediction_data = (19,0,27.9,0,1,3)
prediction_data_array = np.asarray(prediction_data)
prediction_data_reshape = prediction_data_array.reshape(1,-1)
prediction = regressor.predict(prediction_data_reshape)
print('The insurance cost prediction is: ',prediction[0])
print('The real insurance  cost is: 16884.924')
```

```
The insurance cost prediction is:  24601.804100013193
The real insurance  cost is: 16884.924
```

```
prediction_data = (32,1,28.88,0,0,1)
prediction_data_array = np.asarray(prediction_data)
prediction_data_reshape = prediction_data_array.reshape(1,-1)
prediction = regressor.predict(prediction_data_reshape)
print('The insurance cost prediction is: ',prediction[0])
print('The real insurance  cost is: 3866.8552')
```

```
The insurance cost prediction is:  5714.734547769263
The real insurance  cost is: 3866.8552
```

2- Used Random Forest Regression, using multiple decision trees which are merged together for a more accurate prediction.

RandomForestRegressor COST PREDICTION

```
prediction_data = (19,0,27.9,0,1,3)
prediction_data_array = np.asarray(prediction_data)
prediction_data_reshape = prediction_data_array.reshape(1,-1)
prediction = regressor.predict(prediction_data_reshape)
print('The insurance cost prediction is: ',prediction[0])
print('The real insurance  cost is: 16884.924')
```

```
The insurance cost prediction is:  17068.6602015
The real insurance  cost is: 16884.924
```

```
prediction_data = (32,1,28.88,0,0,1)
prediction_data_array = np.asarray(prediction_data)
prediction_data_reshape = prediction_data_array.reshape(1,-1)
prediction = regressor.predict(prediction_data_reshape)
print('The insurance cost prediction is: ',prediction[0])
print('The real insurance  cost is: 3866.8552')
```

```
The insurance cost prediction is:  5085.558999000002
The real insurance  cost is: 3866.8552
```

3- Used Gradient Boosting Regression, divided the data to train and test sets. Using predictions from multiple machine learning algorithms to make a more accurate prediction.

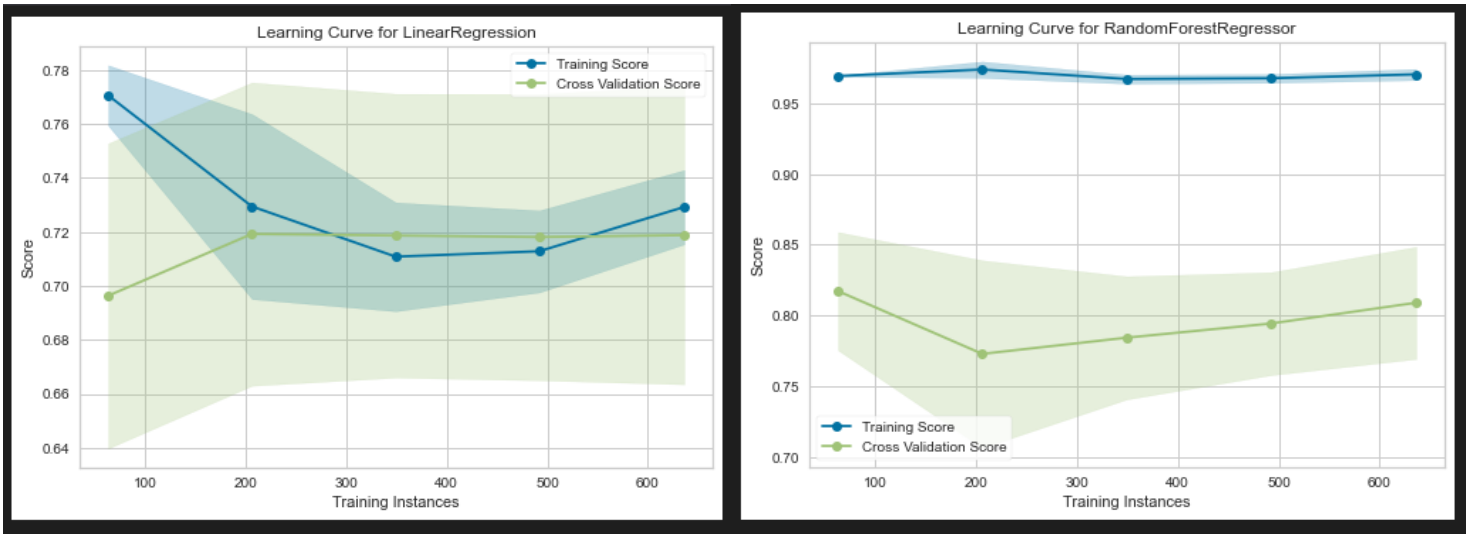
GradientBoostingRegressor COST PREDICTION

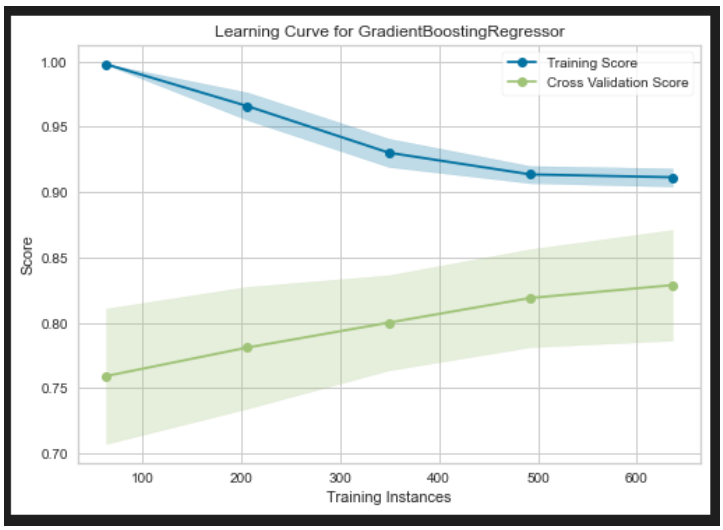
```
prediction_data = (19,0,27.9,0,1,3)
prediction_data_array = np.asarray(prediction_data)
prediction_data_reshape = prediction_data_array.reshape(1,-1)
prediction = reg.predict(prediction_data_reshape)
print('The insurance cost prediction is: ',prediction[0])
print('The real insurance  cost is: 16884.924')
```

```
The insurance cost prediction is:  17713.87363021824
The real insurance  cost is: 16884.924
```

```
prediction_data = (32,1,28.88,0,0,1)
prediction_data_array = np.asarray(prediction_data)
prediction_data_reshape = prediction_data_array.reshape(1,-1)
prediction = reg.predict(prediction_data_reshape)
print('The insurance cost prediction is: ',prediction[0])
print('The real insurance  cost is: 3866.8552')
```

```
The insurance cost prediction is:  4350.600830810343
The real insurance  cost is: 3866.8552
```

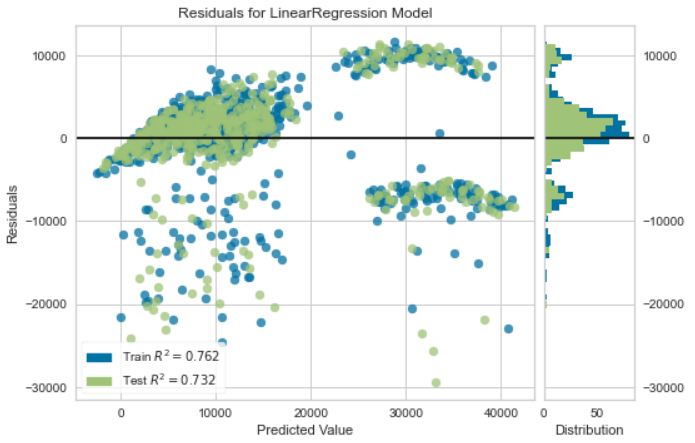


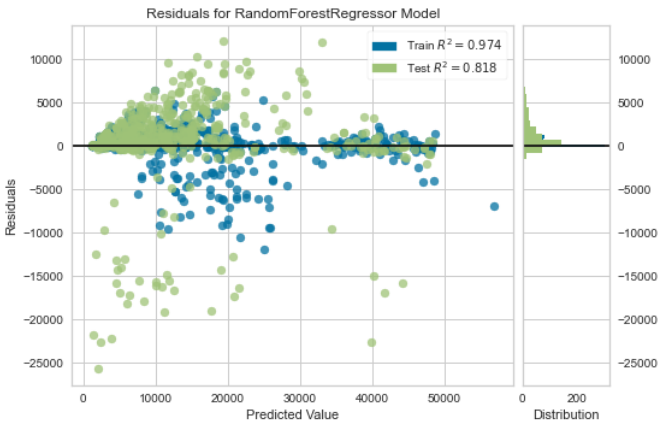


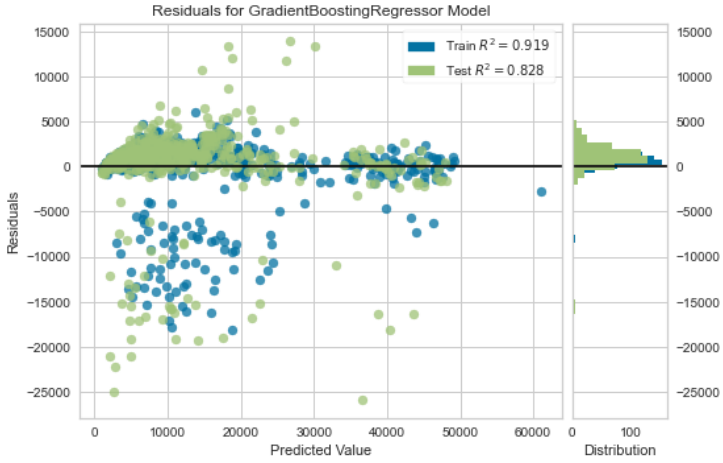
A learning curve shows the relationship of the training score versus the cross validated test score for an estimator with a varying number of training samples. This visualization is typically used to show two things:

1-How much the estimator benefits from more data (e.g. do we have “enough data” or will the estimator get better if used in an online fashion).

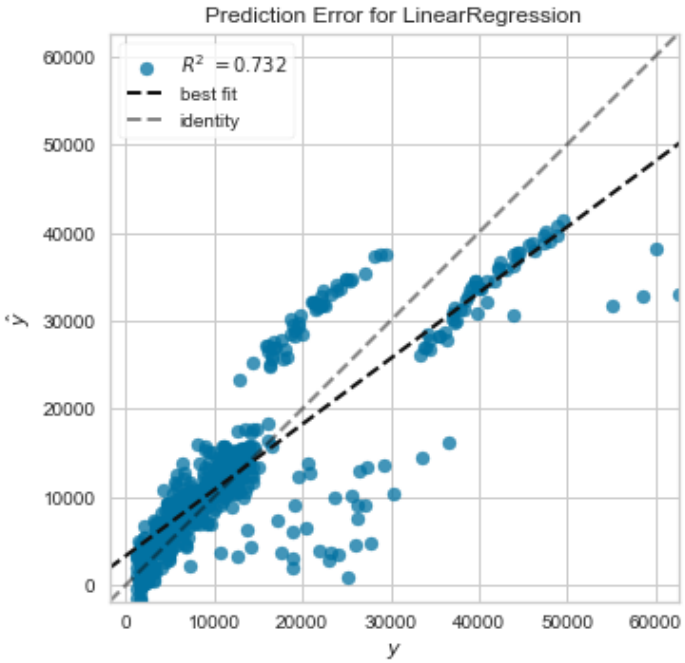
2-If the estimator is more sensitive to error due to variance vs. error due to bias.

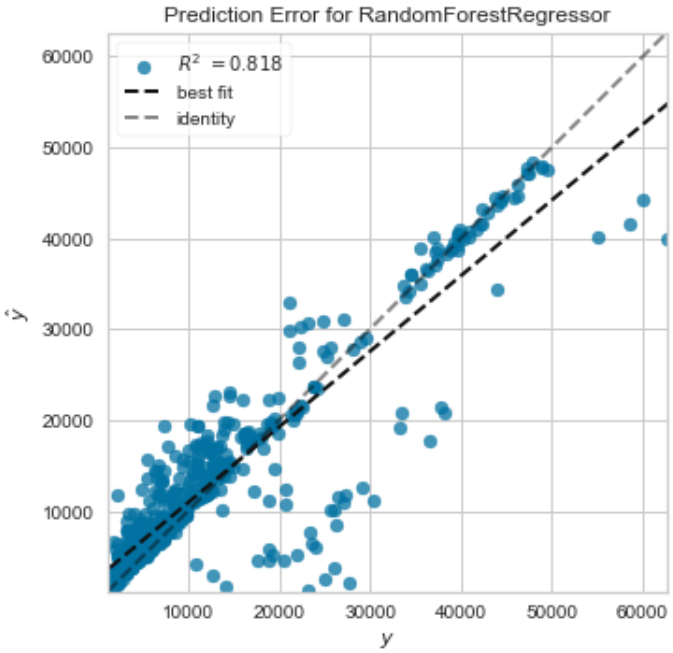


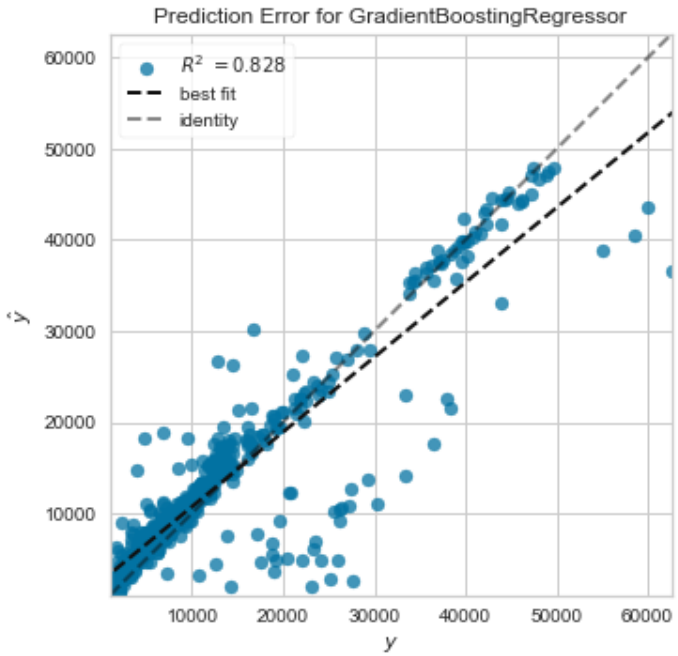




Residuals, in the context of regression models, are the difference between the observed value of the target variable (y) and the predicted value (ŷ), i.e. the error of the prediction. The residuals plot shows the difference between residuals on the vertical axis and the dependent variable on the horizontal axis, allowing you to detect regions within the target that may be susceptible to more or less error.







A prediction error plot shows the actual targets from the dataset against the predicted values generated by our model.