# **Predicting the Insurance Charge in Machine Learning**

## Abstract

In this project, a charge amount for insurance of a person is predicted using Machine Learning algorithms. All the algorithms which supports multiple inputs are taken to sort out the best model and the best model will be saved and handed over th the end user.

### Introduction

This project comes under the supervised Machine Learning learning with the training and test set. The results are in numerical values which falls under the methodology called Regression. A wide array of supervised machine learning algorithms are available for prediction. Here we have chosen Multiple Linear, Support Vector Machine, Decision Tree and Random Forest.

##### Dataset Pre-processing Module

The dataset that was used for this project contains thousand three hundred and thirty nine rows and six columns. The sex and smoker parameters holds nominal data (fig.1) which is converted into numerical data (fig.2) using a pre-processing technique called one hot encoding.

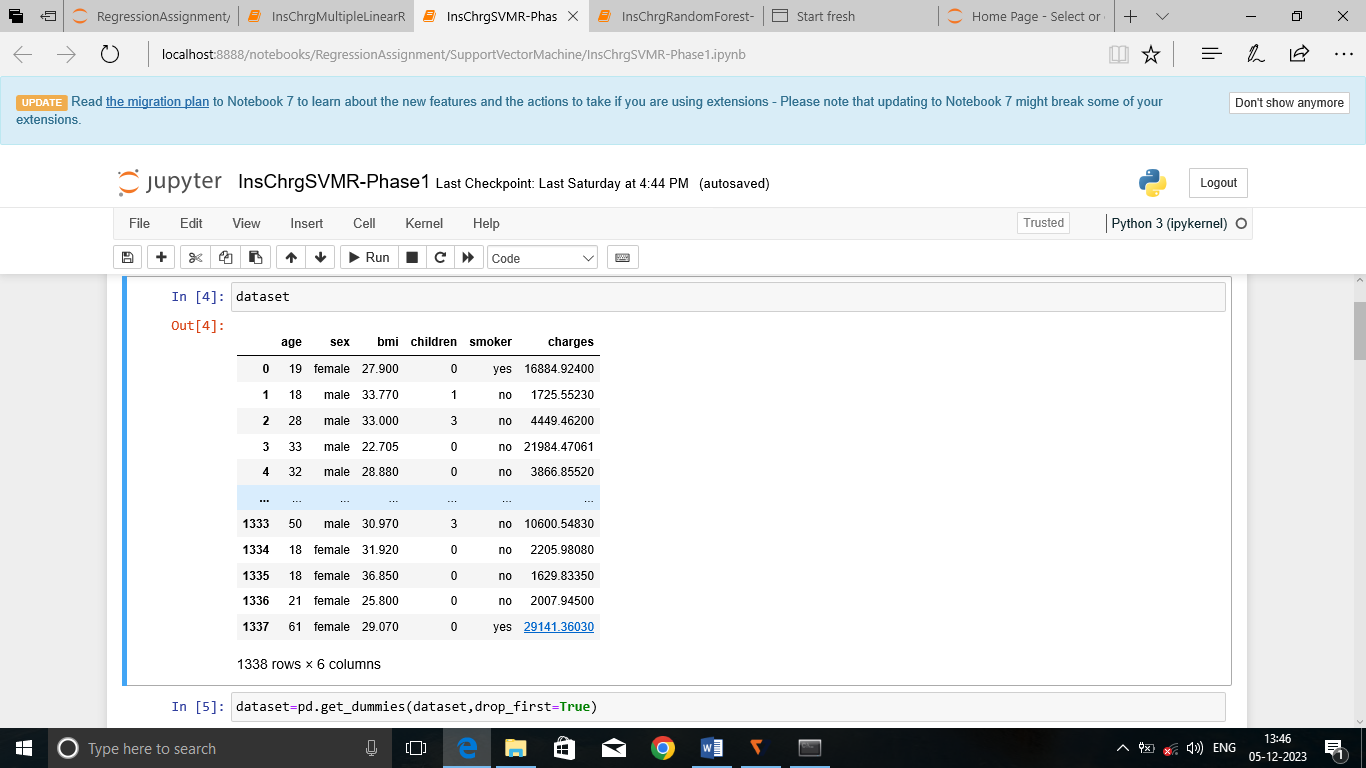


Fig.1 Nominal data

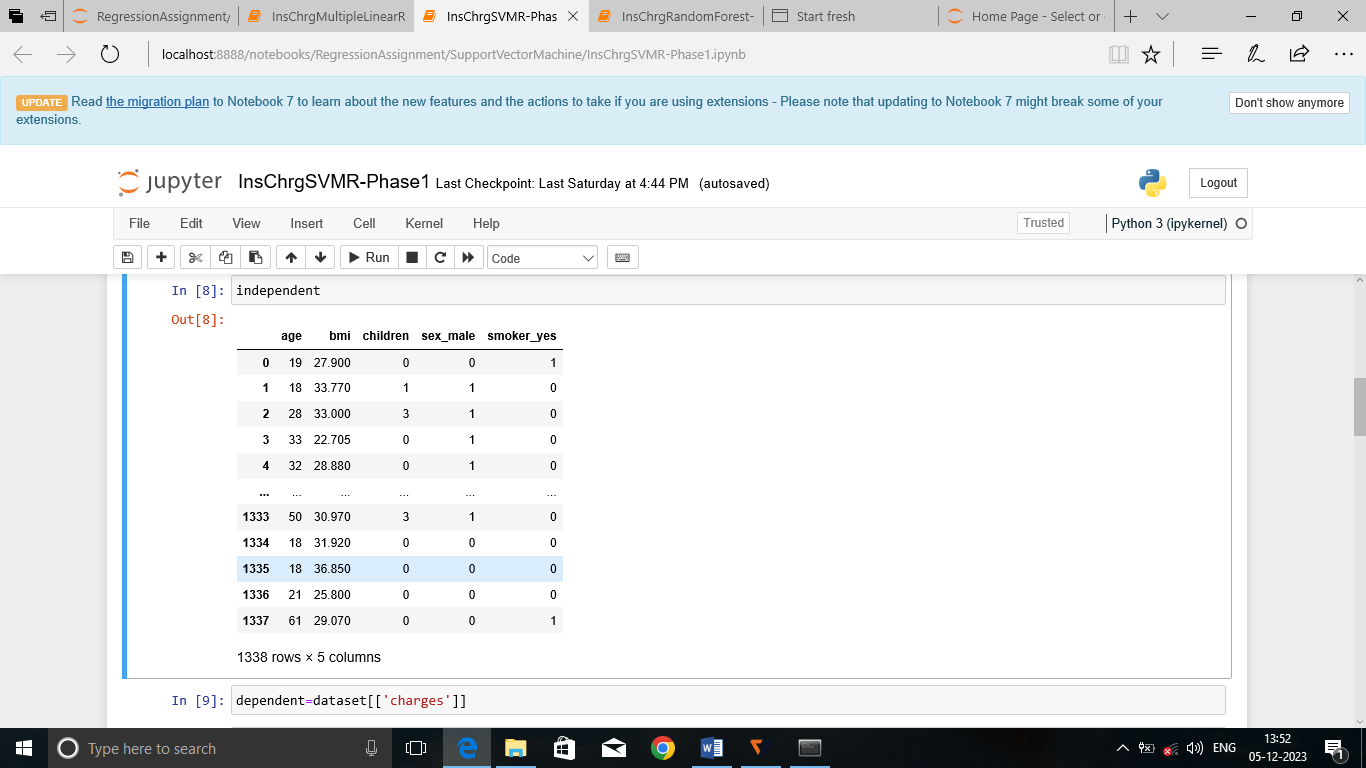


Fig.2 One Hot Encoding Technique

Algorithms :

1.Multiple Linear Regression:

Regression models with one dependant variable and more than one independent variable is called multiple linear regression. In this dataset, insurance charges is the dependant variable and the other parameters are independent variables.

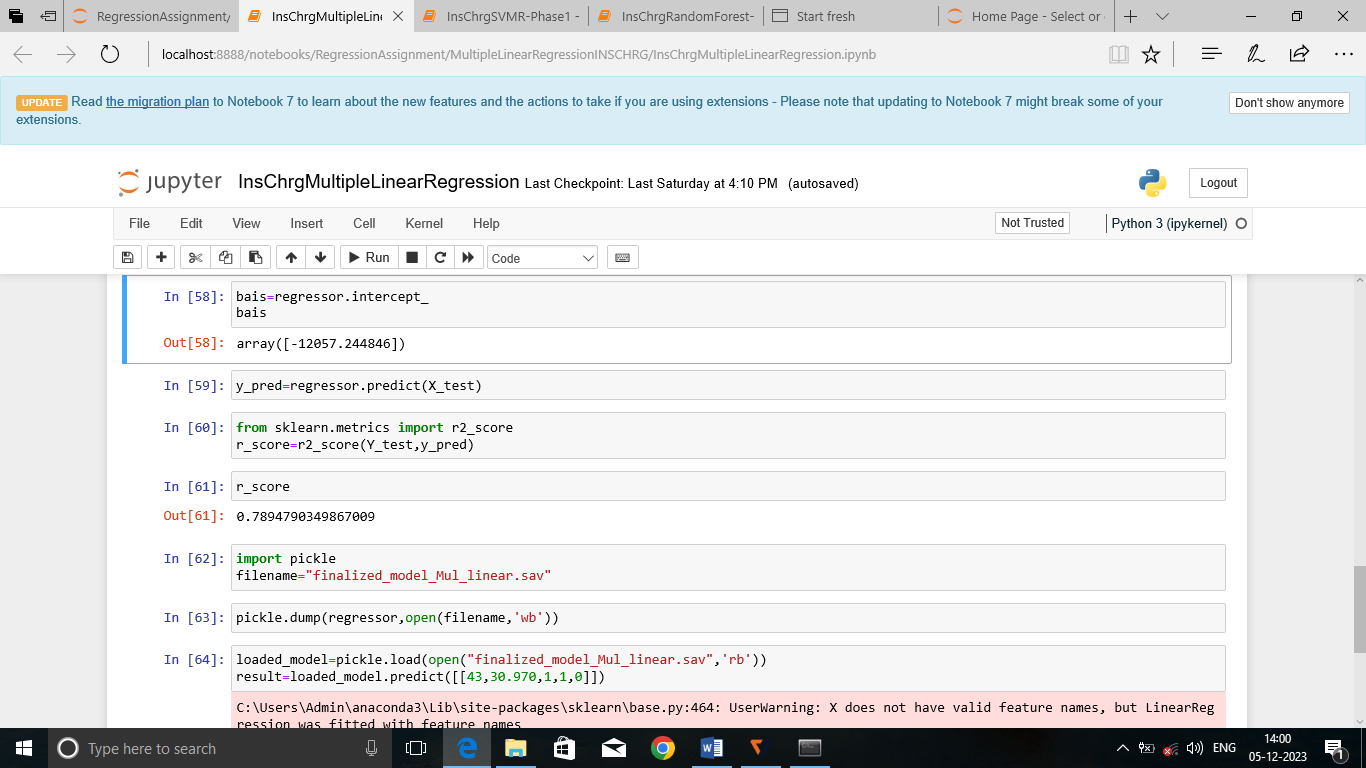


Fig.3.R\_squared value of Multiple Linear Regression

2. Support Vector Machine- Regression

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear and nonlinear data. The main objective of the SVM algorithm is to find the optimal hyper-plane in an N-dimensional space that can separate data points in different classes as maximum as possible.

R \_score with different parameter values:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Kernel | Gamma | R \_score |
| 1. | Linear | Scale | -0.11166128719608448 |
| 2. | Linear | Auto | -0.11166128719608448 |
| 3. | Poly | Auto | -0.05710387514922144 |
| 4. | Poly | scale | -0.06429258402105531 |
| 5. | Sigmoid | scale | -0.0899412170256757 |
| 6. | Sigmoid | Auto | -0.0897090046144422 |
| 7. | Rbf | Auto | -0.08947360164580087 |
| 8. | Rbf | scale | -0.08842732776913875 |

|  |  |  |  |
| --- | --- | --- | --- |
| S .No. | Kernel | C \_value | R \_score |
| 1 | Linear | 10.0 | -0.0016176324886472138 |
| 2 | Linear | 100.0 | 0.5432818196692804 |
| 3 | **Linear** | **1000.0** | **0.634036931263208** |
| 4 | rbf | 10.0 | -0.08196910396420853 |
| 5 | rbf | 100.0 | -0.12480367775039669 |
| 6 | rbf | 1000.0 | -0.11749092439183229 |

3. Decision Forest

This algorithm also supports both classification and regression. It builds a tree like structure from the training dataset given. During training it finds the best attribute that maximizes the information gain.

R\_score values with different parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Criterion** | **Splitter** | **r\_score** |
| 1. | Squared \_error | Best | 0.681381918422653 |
| **2.** | **Squared \_error** | **Random** | **0.7309137972478617** |
| 3. | Friedman \_mse | Best | 0.6859082249139272 |
| 4. | Friedman \_mse | Random | 0.709396536639725 |
| 5. | Absolute \_error | Best | 0.6766916091906545 |
| 6. | Absolute \_error | Random | 0.7695978473389806 |
| 7. | Poisson | Best | 0.6843126265705682 |
| 8. | Poisson | Random | 0.6678142788896531 |

#### 4.Random Forest

#### This algorithm contains several decision trees derived from each column and takes the average to improve the predictive accuracy of the dataset.

#### R \_score values with different parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Criterion** | **n\_estimators** | **r\_score** |
| 1. | **Squared \_error** | **100** | **0.8538307913484513** |
| 2. | Squared \_error | 50 | 0.8498329315421834 |
| 3. | Friedman \_mse | 100 | 0.8500716139332296 |
| 4. | Friedman \_mse | 50 | 0.8500716139332296 |
| 5. | Absolute \_error | 100 | 0.8520093621081837 |
| 6. | Absolute \_error | 50 | 0.8526655993519747 |
| 7. | Poisson | 100 | 0.8526334258892607 |
| 8. | Poisson | 50 | 0.8491075958392151 |

# Result

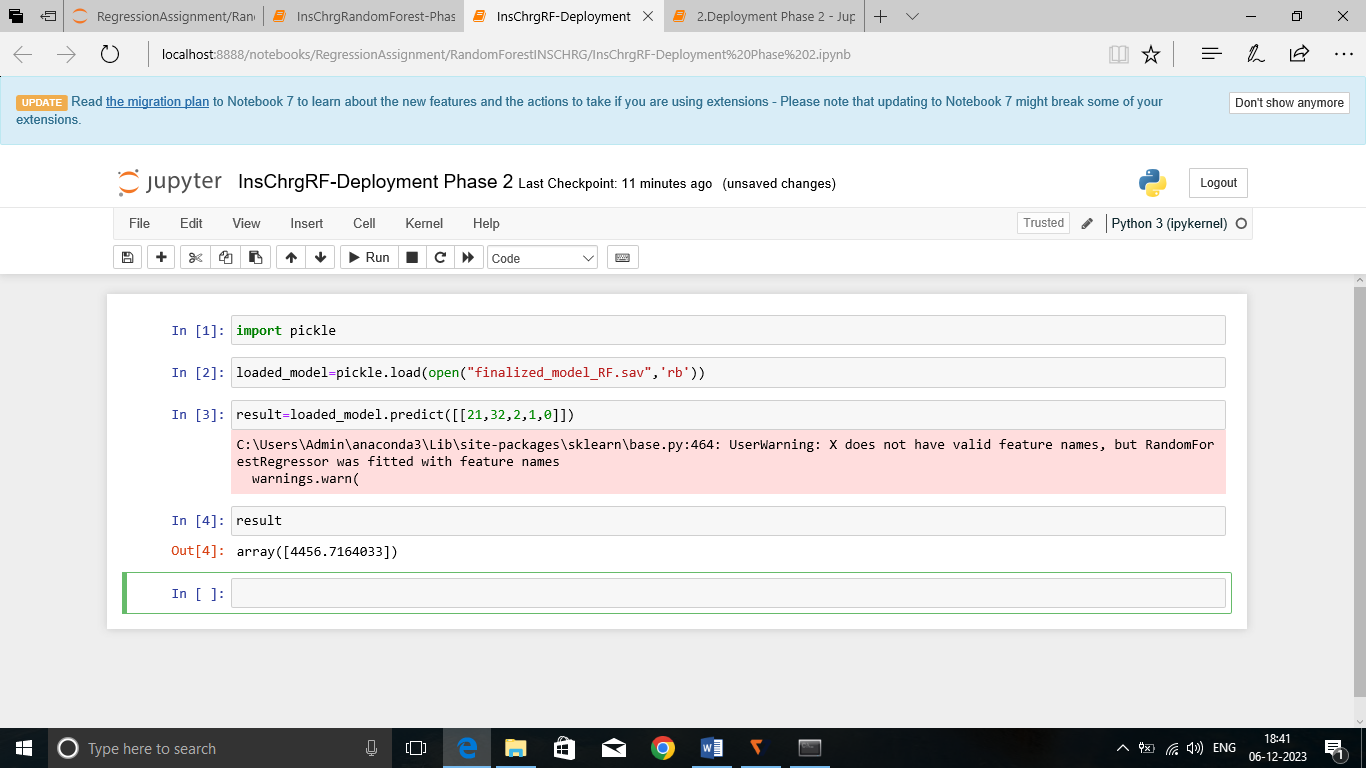


Fig.4.Deployment Phase of RF algorithm

###### Conclusion

As a result the accuracy of the model created from Random Forest algorithm is better compared to the other algorithm’s accuracy level (r\_squared). Hence, the model created from learning the training set through random forest is sorted out and saved to be deployed to the end users.