ARTIFICIAL INTELLIGENCE + MACHINE LEARNING PROJECT

Medical Insurance

Made by

Rishika Garg

CS, 2nd Year

BVCOE, New Delhi

INTRODUCTION

This project has been made using a dataset named 'Insurance' which contains the details of 1338 (possibly fictitious) people who are the primary beneficiaries of a medical insurance scheme.

The task at hand is to train the machine learning model such that based on all or some of the details given in the dataset, the model is able to predict the medical costs billed by the health insurance company per person, as accurately as possible.

The datasets were processed and certain details or records had to be filtered or slightly manipulated/altered in form to make the raw data workable for the model.

5 ML regression models have been used in the process, each delivering a different level of accuracy (in terms of r^2 value and cross-value scores).

ABOUT THE DATASET

The raw dataset contained the following details:

1. Age (The age of the primary beneficiary)
2. Sex (The gender of the insurance contractor)
3. BMI (Body Mass Index, objective index of body weight in kg/m^2 using the ratio of height to weight)
4. Children (Number of children covered by health insurance / Number of dependents)
5. Smoker (Whether the beneficiary smokes or not)
6. Region (The region which beneficiary's residential area falls in)
7. Charges (The individual medical costs billed by health insurance)

These attributes or features were all retained as each had a role to play in determining the output, i.e., the charges.

DATA PREPPING

Also known as data pre-processing, it is the step wherein we alter certain aspects of the data to make it fit for the machine to understand and analyse.

The stages involved in it were:

1. **Encoding –** Changes string data into numerical data.
   Library used: Pandas
   Function/Class: get_dummies
2. **Scaling –** Changes the numerical data with a much larger or smaller size than the rest of the data to be rescaled, such that it's value is retained but it doesn't confuse the machine in terms of priority.
   Library used: sklearn.preprocessing
   Function/Class: StandardScaler()
3. **Splitting –** Changes the numerical data with a much larger or smaller size than the rest of the data to be rescaled, such that it's value is retained but it doesn't confuse the machine in terms of priority.
   Library used: sklearn.model_selection
   Function/Class : train_test_split

ALGORITHMS USED

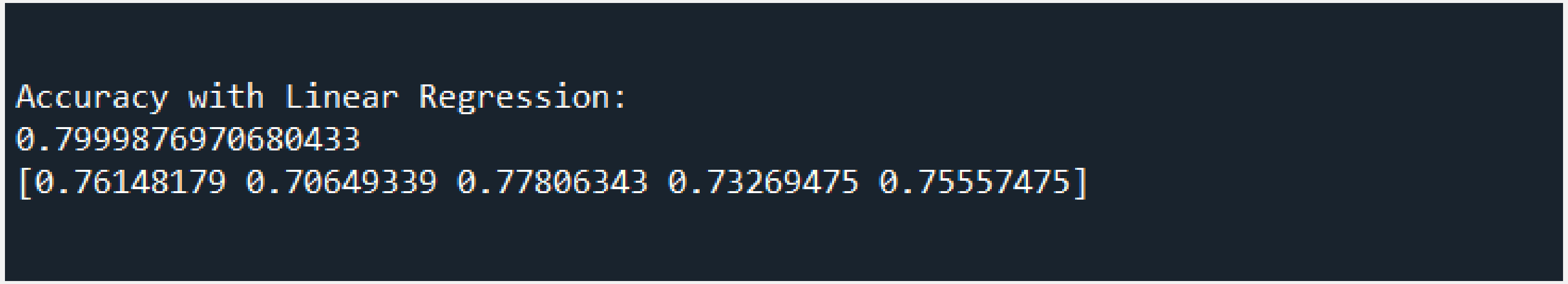
Linear Regression

A linear approach to modeling the relationship between a scalar response and one or more explanatory variables.

From the module sklearn.linear_model, the class LinearRegression was imported.

The model was trained using a fixed 80% of the dataset and tested on the remaining 20%.

The accuracy achieved in terms of r^2 score was nearly 80% and the cross-value score was an average of 74.7%.



```
Accuracy with Linear Regression:
0.7999876970680433
[0.76148179 0.70649339 0.77806343 0.73269475 0.75557475]
```

ALGORITHMS USED

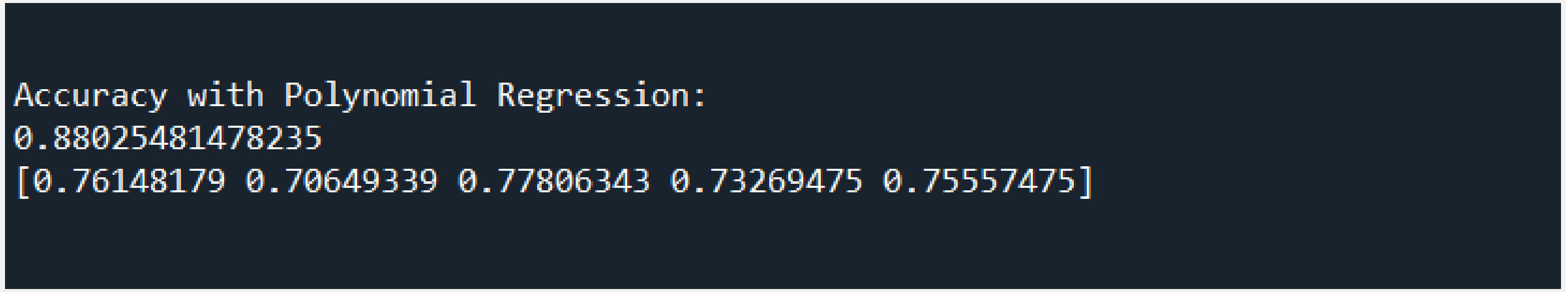
Polynomial Regression

A form of regression analysis in which the relationship between the independent variable X and the dependent variable Y is modelled as an n^{th} degree polynomial in X.

From the module sklearn.preprocessing, the class PolynomialFeatures was imported.

The model was trained using a fixed 80% of the dataset and tested on the remaining 20%.

The accuracy achieved in terms of r^2 score was nearly 88% and the cross-value score was an average of 74.7% (the same as in case of Linear Regression).



```
Accuracy with Polynomial Regression:
0.88025481478235
[0.76148179 0.70649339 0.77806343 0.73269475 0.75557475]
```

ALGORITHMS USED

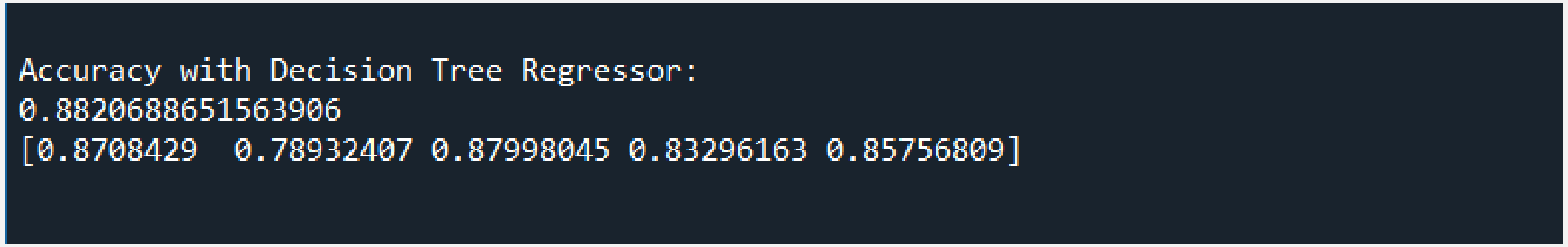
Decision Tree Regression

A regression model in the form of a tree structure wherein a dataset is broken down into smaller subsets and simultaneously, an associated decision tree is incrementally developed, giving the result as a tree with decision nodes and leaf nodes.

From the module sklearn.tree, the class DecisionTreeRegressor was imported.

The model was trained using a fixed 80% of the dataset and tested on the remaining 20%.

The accuracy achieved in terms of r^2 score was more than 88% and the cross-value score was an average of 84.6%.



```
Accuracy with Decision Tree Regressor:
0.8820688651563906
[0.8708429  0.78932407 0.87998045 0.83296163 0.85756809]
```

ALGORITHMS USED

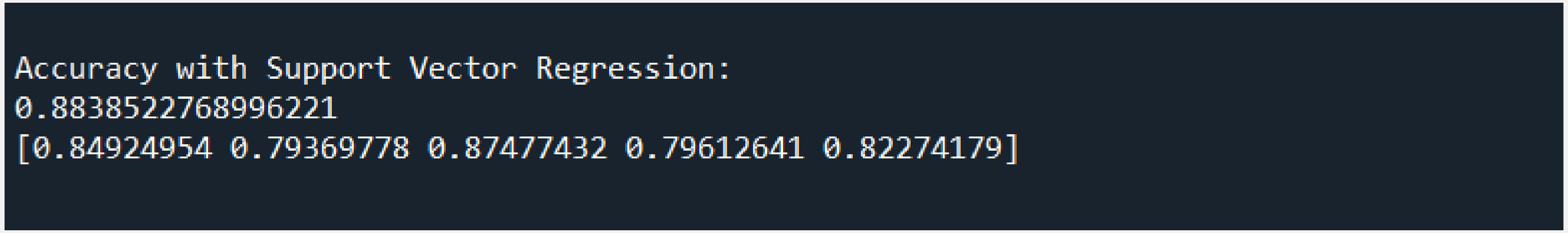
Support Vector (Machine) Regression

A supervised learning model characterized by the use of kernels, sparse solution, and VC control of the margin and the number of support vectors.

From the module sklearn.svm, the class SVR was imported.

The model was trained using a fixed 80% of the dataset and tested on the remaining 20%.

The accuracy achieved in terms of r^2 score was more than 88% and the cross-value score was an average of 82.7%.



```
Accuracy with Support Vector Regression:
0.8838522768996221
[0.84924954 0.79369778 0.87477432 0.79612641 0.82274179]
```

ALGORITHMS USED

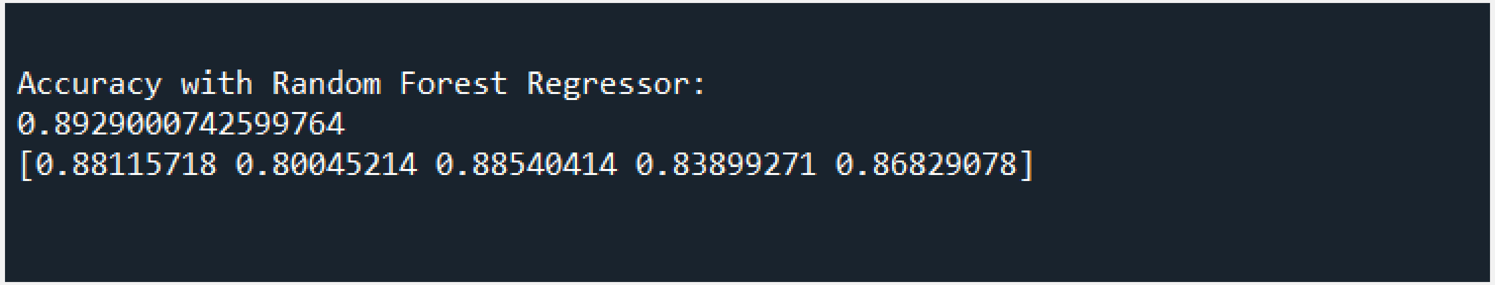
Random Forest Regression

An ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging.

From the module sklearn.ensemble, the class RandomForestRegressor was imported.

The model was trained using a fixed 80% of the dataset and tested on the remaining 20%.

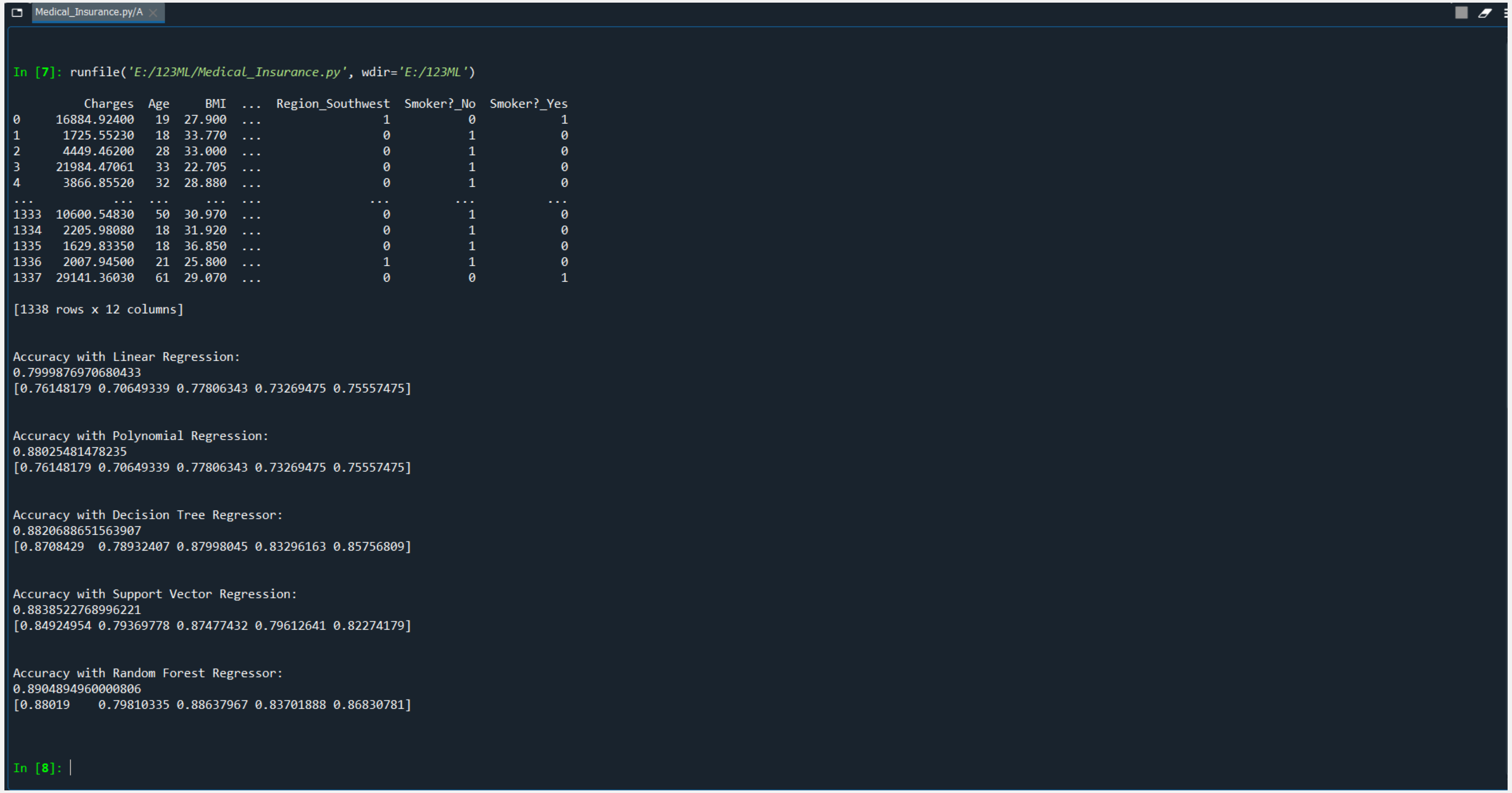
The accuracy achieved in terms of r^2 score was around 89% and the cross-value score was an average of 85.5%.



```
Accuracy with Random Forest Regressor:
0.8929000742599764
[0.88115718 0.80045214 0.88540414 0.83899271 0.86829078]
```

OUTPUT

The output contains the printed dataset (a few rows from the beginning and the end can be seen) and the accuracy achieved in terms of r^2 and cross-value scores with each of the 5 models, in increasing order of r^2 score.



```
In [7]: runfile('E:/123ML/Medical_Insurance.py', wdir='E:/123ML')

          Charges  Age     BMI  ...  Region_Southwest  Smoker?_No  Smoker?_Yes
0     16884.92400   19  27.900  ...                 1           0            1
1      1725.55230   18  33.770  ...                 0           1            0
2      4449.46200   28  33.000  ...                 0           1            0
3     21984.47061   33  22.705  ...                 0           1            0
4      3866.85520   32  28.880  ...                 0           1            0
...           ...  ...     ...  ...               ...         ...          ...
1333  10600.54830   50  30.970  ...                 0           1            0
1334   2205.98080   18  31.920  ...                 0           1            0
1335   1629.83350   18  36.850  ...                 0           1            0
1336   2007.94500   21  25.800  ...                 1           1            0
1337  29141.36030   61  29.070  ...                 0           0            1

[1338 rows x 12 columns]


Accuracy with Linear Regression:
0.7999876970680433
[0.76148179 0.70649339 0.77806343 0.73269475 0.75557475]


Accuracy with Polynomial Regression:
0.88025481478235
[0.76148179 0.70649339 0.77806343 0.73269475 0.75557475]


Accuracy with Decision Tree Regressor:
0.8820688651563907
[0.8708429  0.78932407 0.87998045 0.83296163 0.85756809]


Accuracy with Support Vector Regression:
0.8838522768996221
[0.84924954 0.79369778 0.87477432 0.79612641 0.82274179]


Accuracy with Random Forest Regressor:
0.8904894960000806
[0.88019    0.79810335 0.88637967 0.83701888 0.86830781]



In [8]: 
```

DISCUSSION

It was observed that Linear Regression had the least r^2 and average cross-value scores.

It was also seen that the (average) cross-value scores for Linear and Polynomial Regression were the same, as Polynomial Regression is Linear Regression with polynomial values of X (input).

The maximum r^2 and cross-value scores were seen in the case of Random Forest Regressor, as it creates multiple decision trees and merges them together to obtain a more stable and accurate prediction.

It was also noticed that while SVR slightly outdid Decision Tree in terms of r^2 score, the latter had a considerably higher cross-value score. This could be because while the r^2 score is calculated using only one set of training and testing data, while the cross-value score is a matrix made of a number of such distinct sets.

The future scope of this project involves using more models, using other methods to check the accuracy and giving it a visual angle with graphs, Facet Grids, Heat maps etc.