A PROJECT REPORT ON:

"ML Based Insurance Predictor"

Under subject of

"Data Science and Data Analytics"

(Microsoft Certified)

Group No: 2

Submitted by:

Mr. Patel Shubham Hitesh Bhai

(Course Participant)

(C.K.Pithawala College of Engineering and Technology, Surat)

Submitted to:

Pranav Jaipurkar

(M.E.)

(Knowledge Solutions India)

Group Members:

Dhruv Tilva

Shubham Patel

Dhwani Merai

Savidhi Mittal

Date: 12 September 2021

ACKNOWLEDGEMENT

First and foremost, we would like to express our sincere gratitude to our advisor and guide Mr. Pranav Jaipurkar for the continuous support for our Data Science and Data Analytics study and research, for his patience , motivation , enthusiasm , and immense knowledge. His guidance helped us in all the type of research and writing of this project.

Besides our guide, we would like to thank Knowledge Solutions India for providing us required resources and knowledge.

Last but not the least, we would like to thank our family, our parents for their invaluable support and help throughout the project.

TABLE OF CONTENT

ABSTRACT

Using dataset provided, we built a machine learning model using four different algorithms to predict the ‘Insurance cost’. For this project we use following algorithms ‘Multiple Linear Regressor’, ‘Random Forest Regressor’, ‘Multiple Linear Regressor with PCA’ and ‘Random Forest Regressor with PCA’.

INTRODUCTION

Prediction the insurance cost is an important problem because a health insurance company can only make money if it collects more money then it spends on the medical care of its beneficiaries. But medical costs are difficult to predict since most money comes from rare conditions of the patients. The objective of this project is to accurately predict insurance costs based on people’s data, including age, body mass index, smoking or not, etc. These estimates could be used to create actuarial tables that set the price of yearly premiums higher or lower according to the expected costs.

SOFTWARE LIBRARIES USED

We have used following libraries for this project.

- PANDAS
- SEABORN
- MATPLOTLIB.PLYPLOT
- SKLEARN

PANDAS

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open-source data analysis/manipulation tool available in any language. It is already well on its way toward this goal.

pandas is well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
- Ordered and unordered (not necessarily fixed-frequency) time series data.
- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
- Any other form of observational / statistical data sets. The data need not be labelled at all to be placed into a pandas data structure

The two primary data structures of pandas, Series (1-dimensional) and Data Frame (2- dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering. For R users, Data Frame provides everything that R's data Frame provides and much more. pandas is built on top of NumPy and is intended to integrate well within a scientific computing environment with many other 3rd party libraries.

Here are just a few of the things that pandas does well:

- Easy handling of missing data (represented as NaN) in floating point as well as non- floating-point data
- Size mutability: columns can be inserted and deleted from Data Frame and higher dimensional objects
- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series, Data Frame, etc. automatically align the data for you in computations
- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into Data Frame objects
- Intelligent label-based slicing, fancy indexing, and sub setting of large data sets
- Intuitive merging and joining data sets
- Flexible reshaping and pivoting of data sets
- Hierarchical labelling of axes (possible to have multiple labels per tick)
- Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving / loading data from the ultrafast HDF5 format
- Time series-specific functionality: date range generation and frequency conversion, moving window statistics, date shifting, and lagging.

Many of these principles are here to address the shortcomings frequently experienced using other languages / scientific research environments. For data

scientists, working with data is typically divided into multiple stages: munging and cleaning data, analysing / modelling it, then organizing the results of the analysis into a form suitable for plotting or tabular display. pandas is the ideal tool for all of these tasks.

Some other notes

- pandas is fast. Many of the low-level algorithmic bits have been extensively tweaked in Python code. However, as with anything else generalization usually sacrifices performance. So if you focus on one feature for your application you may be able to create a faster specialized tool.
- pandas is a dependency of stats models, making it an important part of the statistical computing ecosystem in Python.
- pandas has been used extensively in production in financial applications.

IN OUR PROJECT,

We use panda to create data frame

```
import pandas as pd

data = pd.read_csv('insurance.csv')
```

MATPLOTLIB.PLYPLOT

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

A Python matplotlib script is structured so that a few lines of code are all that is required in most instances to generate a visual data plot. The matplotlib scripting layer overlays two APIs:

- The pyplot API is a hierarchy of Python code objects topped by matplotlib.pyplot
- An OO (Object-Oriented) API collection of objects that can be assembled with greater flexibility than pyplot. This API provides direct access to Matplotlib's backend layers.

Matplotlib and Pyplot in Python:

The pyplot API has a convenient MATLAB-style stateful interface. In fact, matplotlib was originally written as an open-source alternative for MATLAB. The OO API and its interface is more customizable and powerful than pyplot, but considered more difficult to use. As a result, the pyplot interface is more commonly used, and is referred to by default in this article.

Understanding matplotlib's pyplot API is key to understanding how to work with plots:

- matplotlib.pyplot.figure: Figure is the top-level container. It includes everything visualized in a plot including one or more Axes.
- matplotlib.pyplot.axes: Axes contain most of the elements in a plot: Axis, Tick, Line2D, Text, etc., and sets the coordinates. It is the area in which data is plotted. Axes include the X-Axis, Y-Axis, and possibly a Z-Axis, as well.

```
from matplotlib import pyplot as plt
%matplotlib inline

variables = ['sex','smoker','region','children']

for v in variables:
    data = data.sort_values(by=[v])
    data[v].value_counts().plot(kind = 'bar')
    plt.title(v)
    plt.show()

plt.scatter(data['age'], data.charges,label='age')
plt.title('Scatter Plot of age and charges')
plt.legend()
plt.show()
plt.scatter(data['bmi'], data.charges,label="BMI")
plt.title('Scatter Plot of bmi and charges')
plt.legend()
plt.show()
```

SKLEARN

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Rather than focusing on loading, manipulating and summarising data, Scikit-learn library is focused on modelling the data. Some of the most popular groups of models provided by Sklearn are as follows −

Supervised Learning algorithms − Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit- learn.

Unsupervised Learning algorithms − On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

Clustering − This model is used for grouping unlabelled data.

Cross Validation − It is used to check the accuracy of supervised models on unseen data.

Dimensionality Reduction − It is used for reducing the number of attributes in data which can
be further used for summarisation, visualisation and feature selection.

Ensemble methods − As name suggest, it is used for combining the predictions of multiple
supervised models.

Feature extraction − It is used to extract the features from data to define the attributes in image
and text data.

Feature selection − It is used to identify useful attributes to create supervised models.

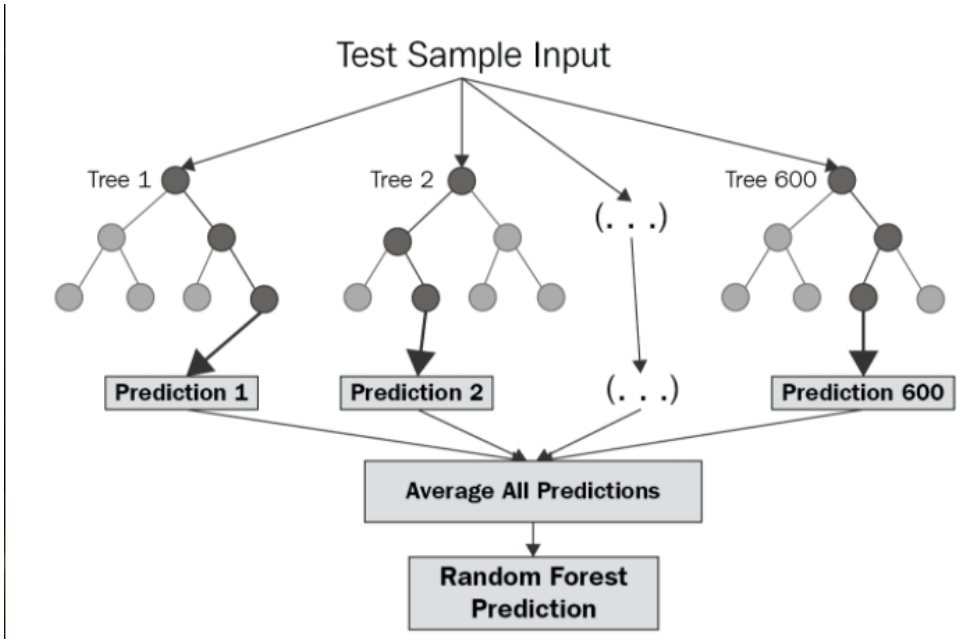
Open Source − It is open-source library and also commercially usable under BSD license.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
import numpy as np

regressor = RandomForestRegressor(n_estimators = 1000,random_state = 0)
regressor.fit(X_train, y_train)

y_pred = regressor.predict(X_test)
regressor.score(X_train,y_train)
```

ALGORITHMS

RANDOM FOREST REGRESSOR

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.



The diagram above shows the structure of a Random Forest. You can notice that the trees run in parallel with no interaction amongst them. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees. To get a better understanding of the Random Forest algorithm, let's walk through the steps:

Pick at random k data points from the training set.

Build a decision tree associated to these k data points.

Choose the number N of trees you want to build and repeat steps 1 and 2.

For a new data point, make each one of your N-tree trees predict the value of y for the data point in question and assign the new data point to the average across all of the predicted y values.

A Random Forest Regression model is powerful and accurate. It usually performs great on many problems, including features with non-linear

relationships. Disadvantages, however, include the following: there is no interpretability, overfitting may easily occur, we must choose the number of trees to include in the model.

ADVANTAGES OF RFR

1. It runs efficiently on large datasets.
2. Random forest has a high accuracy than other algorithms.
3. It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

DISADVANTAGES OF RFR

1. Random forests may result I overfitting for some datasets with noisy regression tasks.
2. For data with categorical variables having a different number of levels, random forests are found to be biased in favour of those attributes with more levels.

CONCLUSION

We have presented a simple model based on 'Multiple Linear Regressor' And 'Random Forest Regressor' for predicting the insurance cost. Different models could be used such as logistic regression, decision tree, etc. It would be a good idea to try these different approaches to see if the results are comparable to the 'Multiple Linear Regressor' And 'Random Forest Regressor' results.