# **Medical Insurance Cost Prediction System**

# **A Project Report**

# ***Submitted by***

# **Priyansha Grover - N015**

**Arjun Khare - N025**

***Under the Guidance of***

**Prof. Ameyaa Biwalkar**

***in partial fulfillment for the award of the degree of***

**MBA TECH**

**COMPUTER ENGINEERING**

**At**



**MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT AND ENGINEERING**

**October 2021**

**DECLARATION**

I, Priyansha Grover, Arjun Khare, Roll Nos. N015, N025, MBA Tech (Computer Engineering), V semester understand that plagiarism is defined as anyone or combination of the following:

Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.

Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)

Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did write what. ( Source:IEEE, The institute, Dec. 2004)

I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidents of plagiarism in this body of work.

Signatures of the Students:

Names: Priyansha Grover, Arjun Khare

Roll Nos. N015, N025

Place:

Date: 26/10/2021

**CERTIFICATE**

This is to certify that the project entitled “Medical Insurance Cost Prediction” is the bonafide work carried out by Priyansha Grover and Arjun Khare of MBA Tech (Computer Engineering), MPSTME (NMIMS), Mumbai, during the V semester of the academic year 2021-22, in partial fulfillment of the requirements for the award of the Degree of MBA Tech as per the norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Ameyaa Biwalkar

Internal Mentor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examiner 1 Examiner 2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Director

**ACKNOWLEDGEMENTS**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Prof. Ameyaa Biwalkar for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

We would like to express our gratitude towards our parents and friends for their kind cooperation and encouragement which helped us in completion of this project.

We would like to express our special gratitude and thanks to our college, Mukesh Patel School of Technology Management and Engineering for providing us with resources

**Table of contents**

| 1.1 | Introduction |
| --- | --- |
| 1.2 | Hardware Specifications |
| 1.3 | Software Specifications |
| 2. | Problem Statement |
| 3. | Scope |
| 4. | Proposed Architecture |
| 5. | Literature Survey |
| 6. | Design and Implementations |
| 7. | Python Code and Outputs |
| 8. | Result And Analysis |
| 9. | Conclusion |
| 10. | References |
|  |  |
|  |  |
|  |  |

**1.1) Introduction**

This project builds a prediction system from a given dataset of Medical Insurance Cost Records, using concepts of Data Mining such as Linear Regression and Train Test Split. The language used for execution is python. Dataset was used for training the models and that training helped to come up with some predictions. Then the predicted amount was compared with the actual data to test and verify the model. Later the accuracies of these models were compared.

**1.2) Hardware Specifications**

• Mouse, Keyboard

• Colour Monitor $ Multimedia Kit

• Minimum 512 Mb of RAM (Minimum)

• Internet connection

• 600 x 800 Resolution

• HDD with 10 GB or Higher capacity

**1.3) Software Specifications:**

• ipython-notebook - Python Text Editor

• sklearn - Machine learning library (For train\_test\_split and LinearRegression)

• seaborn, matplotlib - Visualization libraries

• numpy - number python library

• pandas - data handling library

• metrics - compares and finds accuracy, r squared value

• Windows 7 and up

**2) Problem Statement**

* A Machine Learning system needs to be built for a Medical Insurance Company that can learn from the data and analyse the data to predict what the cost of Medical Insurance will be for certain customers.
* This is required for automatic, accurate and fast prediction from historical data, as calculating it individually for every customer for the company employees will be a tedious task.
* It is a very complex method people can be fooled easily about the amount of the insurance and may unnecessarily buy some expensive health insurance, hence an automatic prediction system is necessary.

**3) Scope**

Current Scope of this project involves:

* Data collection and analysis.
* Data Pre-Processing: Encoding the categorical features, Splitting the Features and Target, Splitting the data into Training data & Testing Data.
* Model Training: Linear Regression, Model Evaluation.
* Building a Predictive System.

Future Scope of this project involves:

* This project is based on a single machine learning model, hence other models like multiple linear regression, decision tree and gradient boosting algorithms can be included for better results and better accuracy.
* This project uses one particular dataset, hence it needs to be modified for other datasets if need be.

**4) Proposed Architecture**

* **Data Analysis** **:** Insurance Cost Dataset is first acquired and analysed.
* **Data Pre-processing** **:** Data Encoding is done to convert Categorical Featured to numerical.
* **Train Test Split :** We split the data into training data and testing data for comparison of accuracy.
* **Linear Regression Model :** We convert it into a trained Linear Regression Model, We can then input New Data for future prediction.

**5) LITERATURE SURVEY**

**What is Predictive modelling?**

Predictive modeling, also called predictive analytics, is a mathematical process that seeks to predict future events or outcomes by analyzing patterns that are likely to forecast future results. The goal of predictive modeling is to answer this question: "Based on known past behavior, what is most likely to happen in the future?

In many use cases, including weather predictions, multiple models are run simultaneously and results are aggregated to create one final prediction. This approach is known as ensemble modeling. As additional data becomes available, the statistical analysis will either be validated or revised.

Once data has been collected, the analyst selects and trains statistical models, using historical data. This is where Data Mining or Machine Learning algorithms come into play. The dataset is processed and then fitted into a model, which later is trained to give accurate predictions, as we further implement in this project.

**Benefits of Predictive Modeling:**

* In a nutshell, predictive analytics reduce time, effort and costs in forecasting business outcomes. Variables such as environmental factors, competitive intelligence, regulation changes and market conditions can be factored into the mathematical calculation to render more complete views at relatively low costs.
* Examples of specific types of forecasting that can benefit businesses include demand forecasting, headcount planning, churn analysis, external factors, competitive analysis, fleet and IT hardware maintenance and financial risks.

**Challenges of Predictive Modeling**

* It’s essential to keep predictive analytics focused on producing useful business insights because not everything this technology digs up is useful. Some mined information is of value only in satisfying a curious mind and has few or no business implications. Getting side-tracked is a distraction few businesses can afford.
* Also, being able to use more data in predictive modeling is an advantage only to a point. Too much data can skew the calculation and lead to a meaningless or an erroneous outcome. For example, more coats are sold as the outside temperature drops. But only to a point. People do not buy more coats when it’s -20 degrees Fahrenheit outside than they do when it’s -5 degrees below freezing. At a certain point, cold is cold enough to spur the purchase of coats and more frigid temps no longer appreciably change that pattern.
* And with the massive volumes of data involved in predictive modeling, maintaining security and privacy will also be a challenge. Further challenges rest in machine learning’s limitations.

**Limitations of Predictive Modeling:**

According to a McKinsey report, common limitations and their “best fixes” include:

* Errors in data labeling
* Shortage of massive data sets needed to train machine learning: Apossible fix is “one-shot learning,” wherein a machine learns from a small number of demonstrations rather than on a massive data set.
* The machine’s inability to explain what and why it did what it did: Machines do not “think” or “learn” like humans. Likewise, their computations can be so exceptionally complex that humans have trouble finding, let alone following, the logic. All this makes it difficult for a machine to explain its work, or for humans to do so. Yet model transparency is necessary for a number of reasons, with human safety chief among them. Promising potential fixes: local-interpretable-model-agnostic explanations (LIME) and attention techniques.

**The Future of Predictive Modeling:**

Predictive modeling, also known as predictive analytics, and machine learning are still young and developing technologies, meaning there is much more to come. As techniques, methods, tools and technologies improve, so will the benefits to businesses and societies.

However, these are not technologies that businesses can afford to adopt later, after the tech reaches maturity and all the kinks are worked out. The near-term advantages are simply too strong for a late adopter to overcome and remain competitive.

It is almost necessary for businesses to understand and deploy the technology now and then to grow the business benefits alongside subsequent advances in the technologies.

**Types of predictive analytics models:**

The top five predictive analytics models are:

**Common Predictive Algorithms:**

Predictive algorithms use one of two things: machine learning or deep learning. Both are subsets of artificial intelligence (AI). Machine learning (ML) involves structured data, such as spreadsheet or machine data. Deep learning (DL) deals with unstructured data such as video, audio, text, social media posts and images—essentially the stuff that humans communicate with that are not numbers or metric reads.

Some of the more common predictive algorithms are:

**Random Forest:** This algorithm is derived from a combination of decision trees, none of which are related, and can use both classification and regression to classify vast amounts of data.

Generalized Linear Model (GLM) for Two Values: This algorithm narrows down the list of variables to find “best fit.” It can work out tipping points and change data capture and other influences, such as categorical predictors, to determine the “best fit” outcome, thereby overcoming drawbacks in other models, such as a regular linear regression.

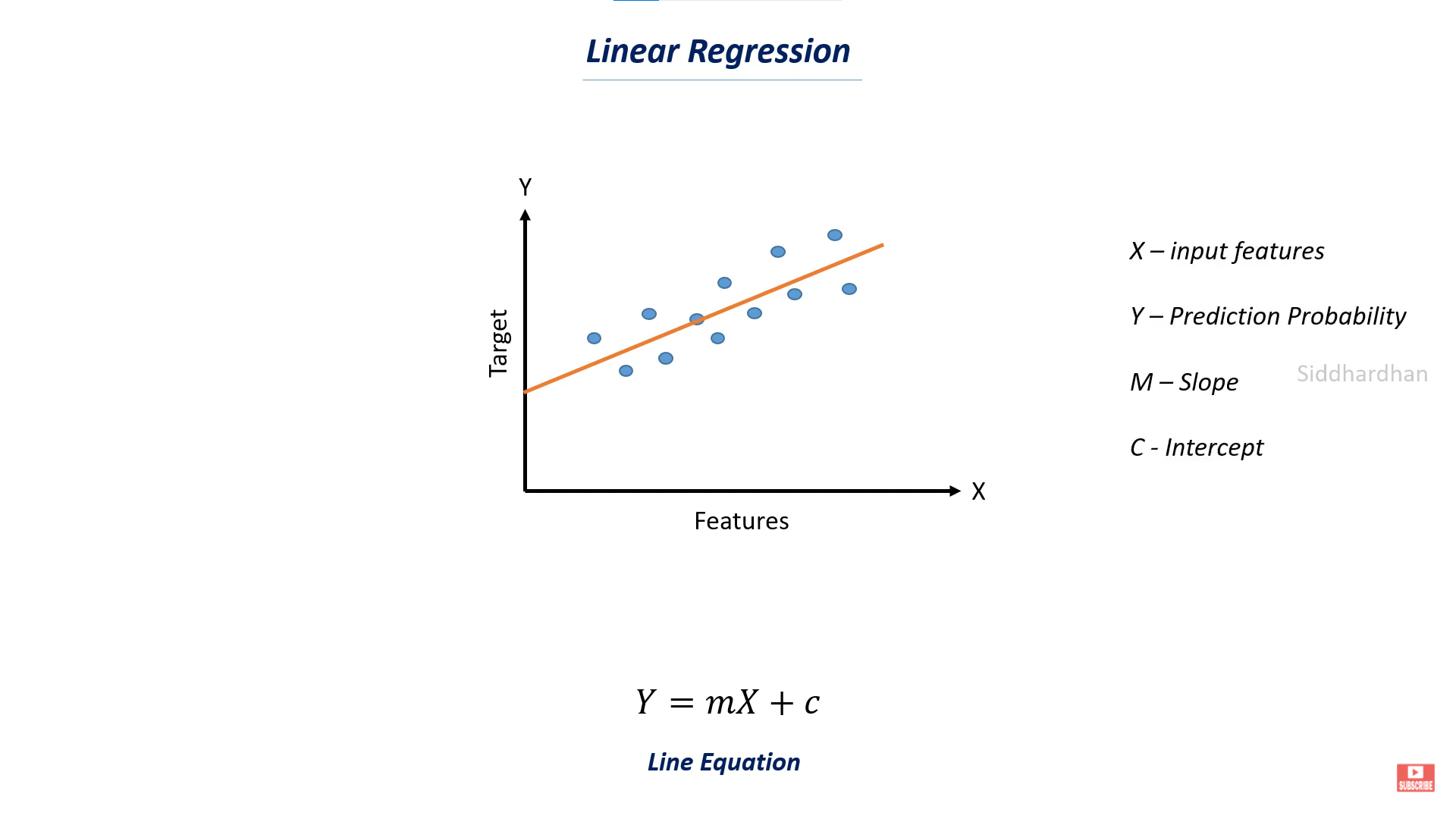
**Gradient Boosted Model:** This algorithm also uses several combined decision trees, but unlike Random Forest, the trees are related. It builds out one tree at a time, thus enabling the next tree to correct flaws in the previous tree. It’s often used in rankings, such as on search engine outputs.

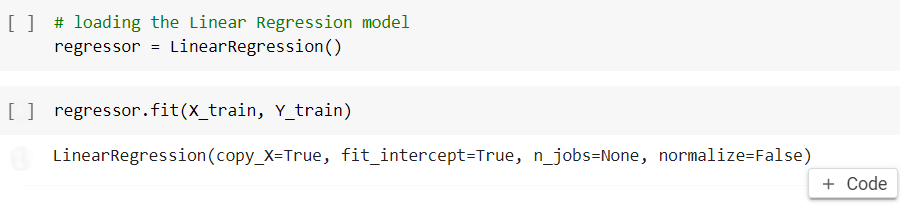
**K-Means:** A popular and fast algorithm, K-Means groups data points by similarities and so is often used for the clustering model. It can quickly render things like personalized retail offers to individuals within a huge group, such as a million or more customers with a similar liking of lined red wool coats.

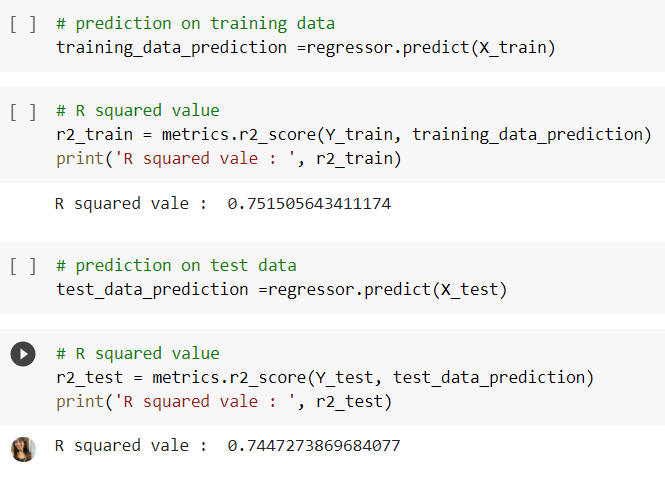
**Linear Regression Model:** This algorithm is used in this project and is explained in detail further.

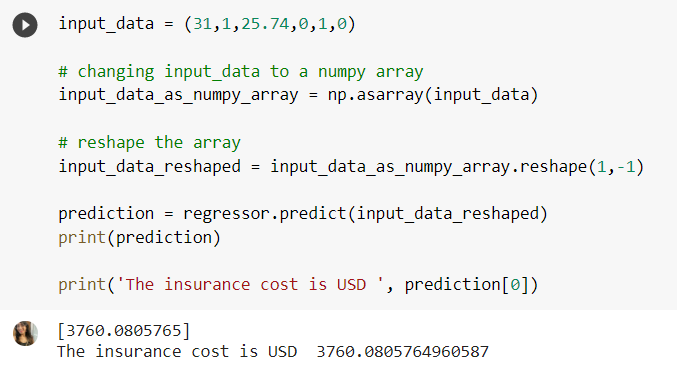
**Proposed Algorithms : Linear Regression**

* Linear regression is a basic and commonly used type of predictive analysis.
* It is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables.
* It has two variables used, one that is Dependent and the other Independent. The independent variables are fitted into the x-axis through which the dependent variables are calculated or predicted on the y-axis, based on any given information or data, or slope and intercept.
* The formula that linear regression follows is : Y = MX + C.
* Y is the dependent variable, X is the independent variable.
* M is the slope of the curve, and C is the constant or intercept.
* In this project, some new data is inputted into the program, which inputs to the X-axis of the graph.
* This value at the X-axis finds an intercept on the curve and finds the resultant Y-axis value.



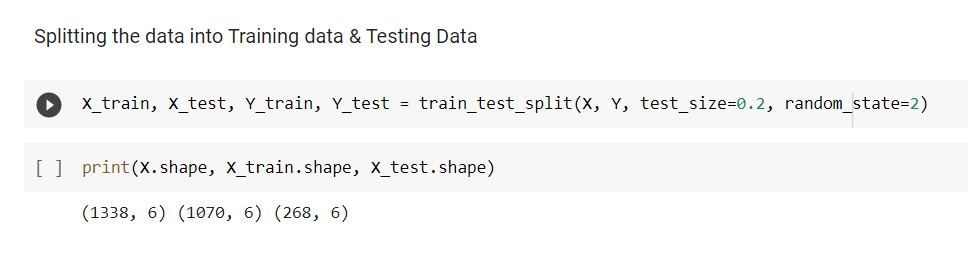


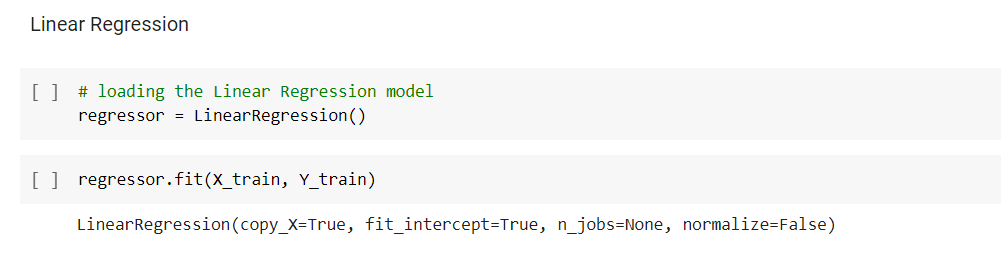


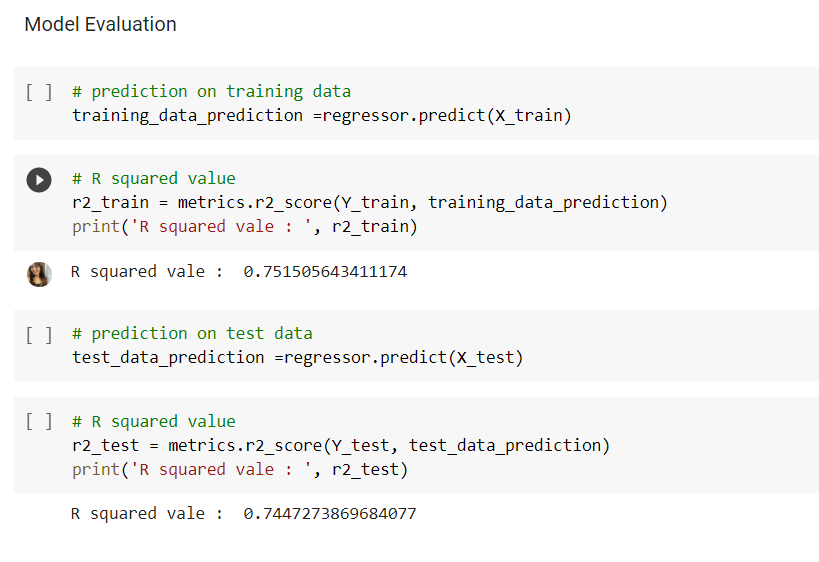


**Proposed Algorithms : Train Test Splitting**

* Train train splitting is done using a method that is imported from the sklearn library.
* It is imported so that a dataset can be split into 2 types : Train and Test data.
* Train data is used to train a predictive model by feeding it data to process and its resultant outputs as well. This gives the predictive model an idea of what the prediction should be like.
* The test data is used to test and confirm the accuracy and efficiency of the predictive model by being inputted into the predictive model. Its outputs are compared to the original outputs in the dataset and measured for accuracy or R-Squared values.
* This helps in understanding the accuracy of the predictive data in explicit detail.







**6) Design and Implementation**

**Step 1 :** Importing the dependencies for the code

* NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices.
* Pandas is just a library of python used for data manipulation and analysis. It also reads csv files as datasets, which makes it easier to do analysis.
* Matplotlib is a cross-platform, data visualization and graphical plotting library for Python.
* Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library.
* To divide a given dataset into train data and test data, we have imported train\_test\_split.
* To predict data based on a given input and data we have imported LinearRegression.
* Metrics is imported to evaluate the accuracy of the predictive model used for the given dataset by finding r squared value.

**Step 2 :** Data Collection & Analysis

* Data is loaded as a dataset from a csv file.
* Number of rows and columns in the dataset are found out.
* Information regarding the columns of the dataset are analysed.
* The dataset is checked for any null values.
* The dataset is described to see the minimum value, maximum value, standard deviation, and 25%, 50%, 75% quartiles.
* The graphs of all columns(age, sex, bmi, children, smoker, region, charges) are plotted and anaysed. The categorical columns have count graphs plotted, whereas numeric columns have distribution graphs plotted.

**Step 3 :** Data Pre-Processing

* First we encode the categorical features because the data cannot be predicted as it is not numerical, hence it needs to be encoded for further processing. The categorical features have a certain number of values hence the particular features can be encoded as certain numbers.
* Since the charges column is the target of the predictive algorithm, we split the dataset into features and target for further processing. The other columns are the features.
* Then we divide the dataset further into train and test data in order to compare predicted insurance cost from the model with actual costs to evaluate the accuracies of the model.

**Step 4 :** Model Training

* We load the Linear Regression model on a variable.
* We then fit the X and Y training data on the X and Y axis of the Regression Model respectively.

**Step 5 :** Model Evaluation

* To test or evaluate the data, we apply linear regression prediction on the X training data, so that its results can be compared to the actual Y training data.
* To compare the two, the R squared value is computed.
* R-squared is a statistical measure of how close the data are to the fitted regression line. It provides accuracy information by comparing the outputted data with the data provided.
* This same process is repeated for the Test Values.

**Step 6 :** Building a Predictive Model

* We have first selected data to input.
* The input consists of all values for all columns except for the charges as that is our target.
* We have to make sure that the input for the encoded categorical columns are also encoded, or else, the predictive model will give an error.
* Since the input selected is stored in parentheses, it is a tuple.
* We make it into a NumPy array so that it is easier to process.
* Then we need to reshape the array.
* This is done because the model does not know that we are predicting for one data point, as we have used over 1000 data points while training the model, and the model would expect the same number of values as the number of training data points used.
* That is why we use reshape(1,-1), as it tells the model to predict for one data point.
* Now we apply the prediction model to the reshaped input data.
* The predicted price of the insurance is displayed.
* We mention prediction[0], as prediction is in the form of a list which contains only one value, and 0 is the first index of the list, which we have to output.

**7) Python Code and Outputs**

**Code :**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn import metrics

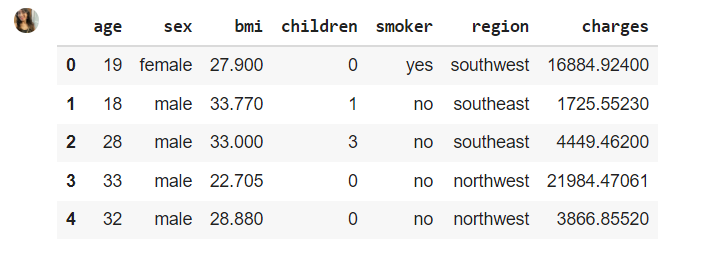
# loading the data from csv file to a Pandas DataFrame

insurance\_dataset = pd.read\_csv('/content/insurance.csv')

# first 5 rows of the dataframe

insurance\_dataset.head()

**Output:**



# number of rows and columns

Insurance\_dataset.shape

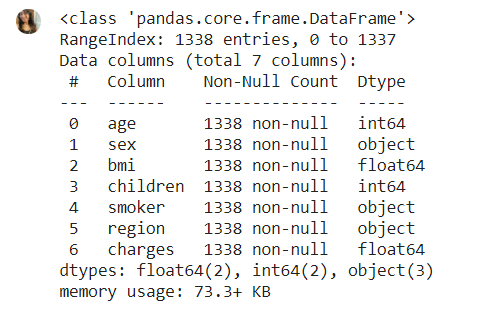
**Output:**



# getting some informations about the dataset

insurance\_dataset.info()

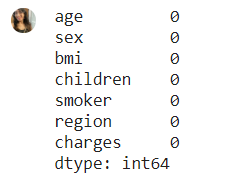
**Output:**



# checking for missing values

insurance\_dataset.isnull().sum()

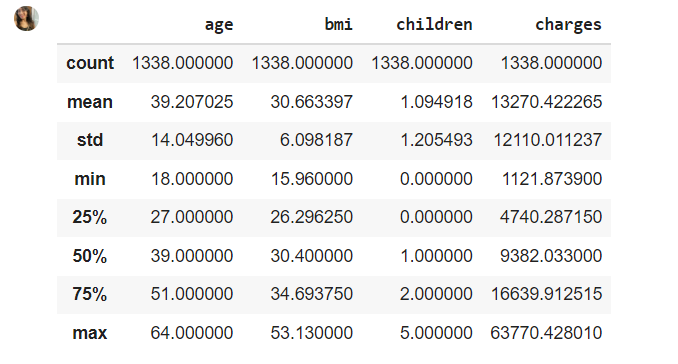
**Output:**



# statistical Measures of the dataset

insurance\_dataset.describe()

**Output:**



# distribution of age value

sns.set()

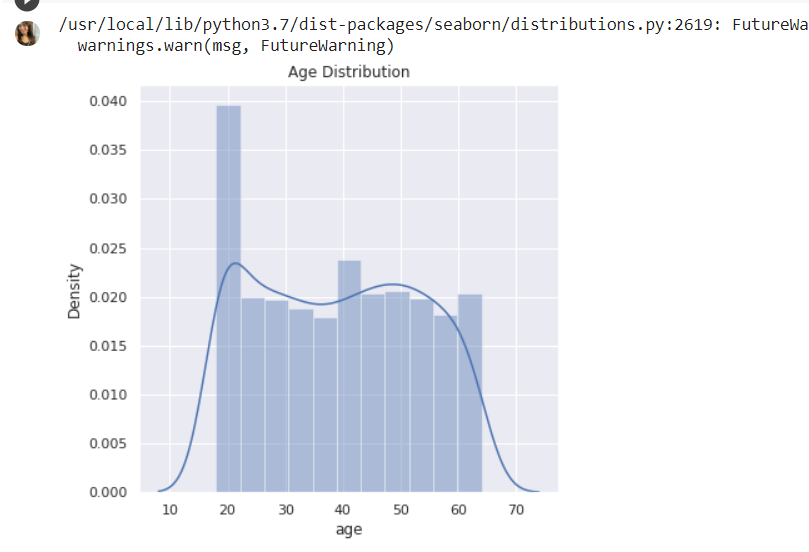
plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['age'])

plt.title('Age Distribution')

plt.show()

**Output:**



# Gender column

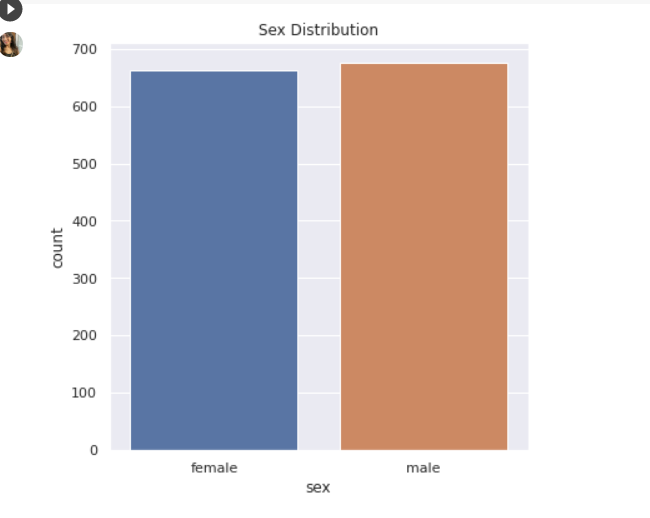
plt.figure(figsize=(6,6))

sns.countplot(x='sex', data=insurance\_dataset)

plt.title('Sex Distribution')

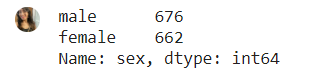
plt.show()

**Output:**



insurance\_dataset['sex'].value\_counts()

**Output:**



# bmi distribution

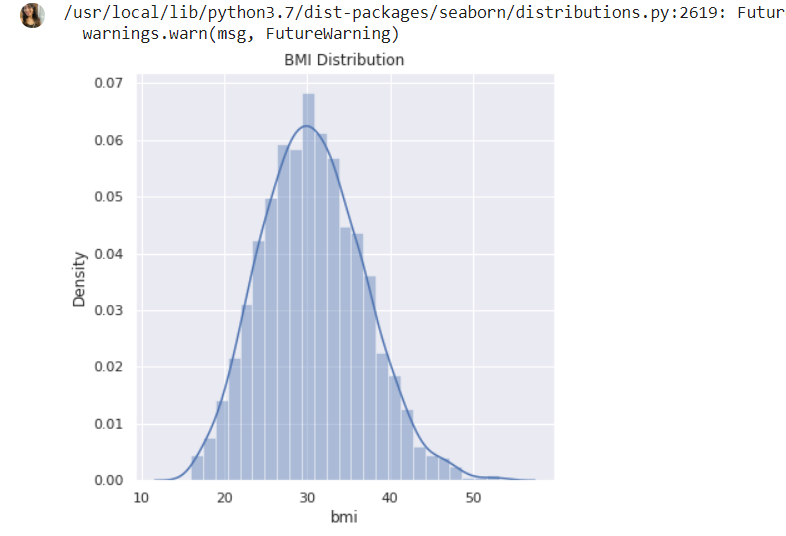
plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['bmi'])

plt.title('BMI Distribution')

plt.show()

**Output:**



# children column

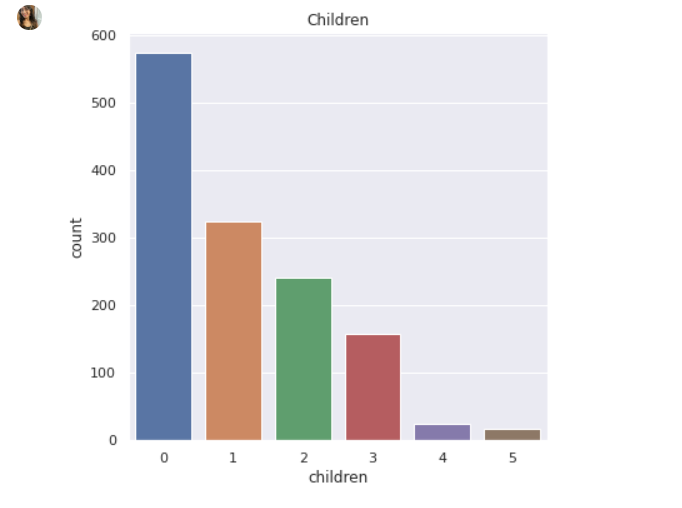
plt.figure(figsize=(6,6))

sns.countplot(x='children', data=insurance\_dataset)

plt.title('Children')

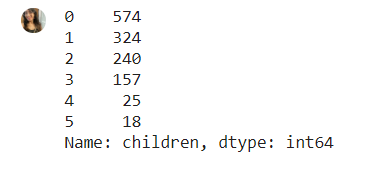
plt.show()

**Output:**



insurance\_dataset['children'].value\_counts()

**Output:**



# smoker column

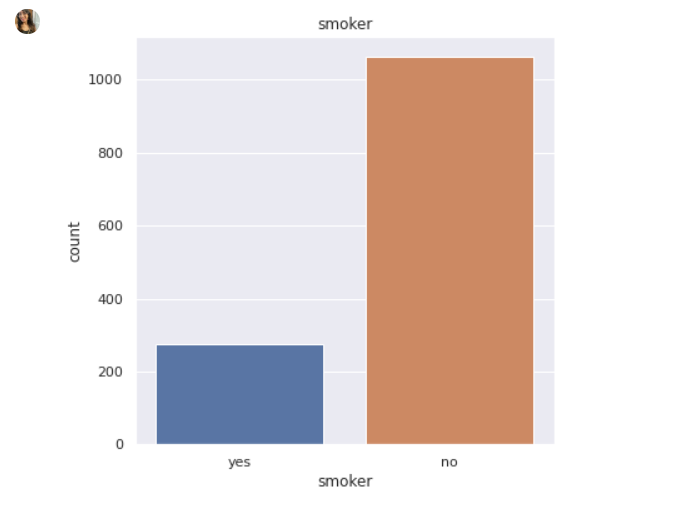
plt.figure(figsize=(6,6))

sns.countplot(x='smoker', data=insurance\_dataset)

plt.title('smoker')

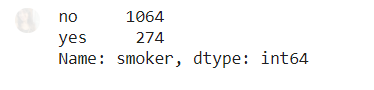
plt.show()

**Output:**



insurance\_dataset['smoker'].value\_counts()

**Output:**



# region column

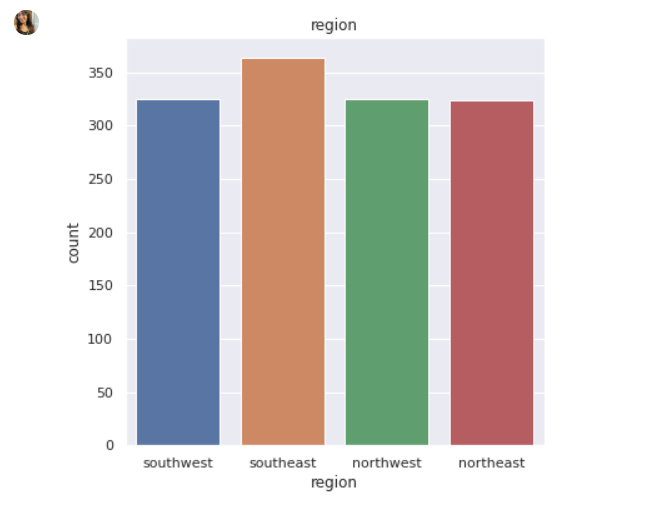
plt.figure(figsize=(6,6))

sns.countplot(x='region', data=insurance\_dataset)

plt.title('region')

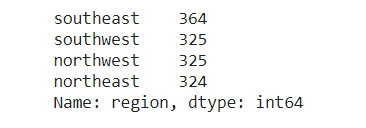
plt.show()

**Output:**



insurance\_dataset['region'].value\_counts()

**Output:**



# distribution of charges value

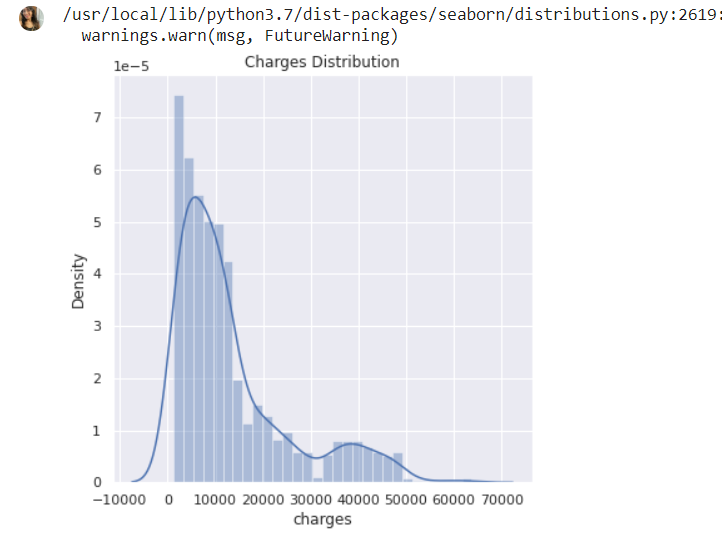
plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['charges'])

plt.title('Charges Distribution')

plt.show()

**Output:**



# encoding sex column

insurance\_dataset.replace({'sex':{'male':0,'female':1}}, inplace=True)

3 # encoding 'smoker' column

insurance\_dataset.replace({'smoker':{'yes':0,'no':1}}, inplace=True)

# encoding 'region' column

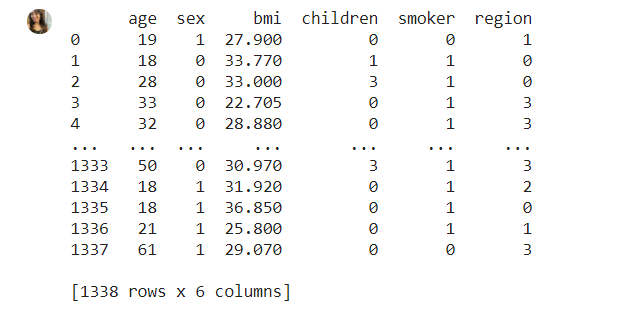
insurance\_dataset.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'northwest':3}}, inplace=True)

X = insurance\_dataset.drop(columns='charges', axis=1)

Y = insurance\_dataset['charges']

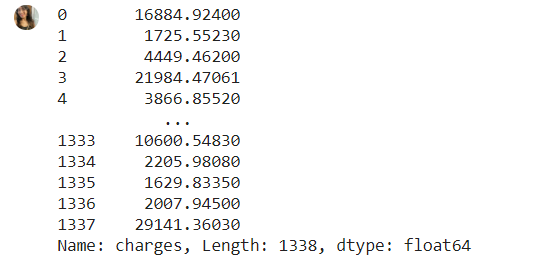
print(X)

**Output:**



print(Y)

**Output:**



X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=2)

print(X.shape, X\_train.shape, X\_test.shape)

**Output:**



# loading the Linear Regression model

regressor = LinearRegression()

regressor.fit(X\_train, Y\_train)

**Output:**



# prediction on training data

training\_data\_prediction =regressor.predict(X\_train)

# R squared value

r2\_train = metrics.r2\_score(Y\_train, training\_data\_prediction)

print('R squared vale : ', r2\_train)

**Output:**



# prediction on test data

test\_data\_prediction =regressor.predict(X\_test)

# R squared value

r2\_test = metrics.r2\_score(Y\_test, test\_data\_prediction)

print('R squared vale : ', r2\_test)

**Output:**



input\_data = (31,1,25.74,0,1,0)

# changing input\_data to a numpy array

input\_data\_as\_numpy\_array = np.asarray(input\_data)

# reshape the array

input\_data\_reshaped = input\_data\_as\_numpy\_array.reshape(1,-1)

prediction = regressor.predict(input\_data\_reshaped)

print(prediction)

print('The insurance cost is USD ', prediction[0])

**Output:**



**8) Result And Analysis**

As we will see from the code and output below

# prediction on training data

training\_data\_prediction =regressor.predict(X\_train)

# R squared value

r2\_train = metrics.r2\_score(Y\_train, training\_data\_prediction)

print('R squared vale : ', r2\_train)

**Output:**



Since, the R squared value is close to 1, it proves that the predictive model is very effective and accurate for the train dataset. R squared values are measured between 0 to 1.

# prediction on test data

test\_data\_prediction =regressor.predict(X\_test)

# R squared value

r2\_test = metrics.r2\_score(Y\_test, test\_data\_prediction)

print('R squared vale : ', r2\_test)

**Output:**



Now that we have compared the values of the train data using the train data fitted on the X-axis of the linear regression graph, and found the R squared value, we do the same process for the test value. We see the result is yet again close to 1. This is definitive proof that the algorithm is accurate and effective for the test dataset, and therefore the whole dataset.

input\_data = (31,1,25.74,0,1,0)

# changing input\_data to a numpy array

input\_data\_as\_numpy\_array = np.asarray(input\_data)

# reshape the array

input\_data\_reshaped = input\_data\_as\_numpy\_array.reshape(1,-1)

prediction = regressor.predict(input\_data\_reshaped)

print(prediction)

print('The insurance cost is USD ', prediction[0])

**Output:**



We have first selected data to input. The input consists of all values for all columns except for the charges as that is our target. We have to make sure that the input for the encoded categorical columns are also encoded, or else, the predictive model will give an error. Since the input selected is stored in parentheses, it is a tuple. We make it into a NumPy array so that it is easier to process. Then we need to reshape the array. This is done because the model does not know that we are predicting for one data point, as we have used over 1000 data points while training the model, and the model would expect the same number of values as the number of training data points used. That is why we use reshape(1,-1), as it tells the model to predict for one data point. Now we apply the prediction model to the reshaped input data. The predicted price of the insurance is displayed. We mention prediction[0], as prediction is in the form of a list which contains only one value, and 0 is the first index of the list, which we have to output.

We have achieved our goal in predicting the cost of the medical insurance based on the independent variables selected.

**9) Conclusion**

In this project, The health insurance data was used to develop the regression model, and the predicted insurance premiums from the model were compared with actual premiums to compare the accuracies of the model. It has been concluded that the algorithm is extremely accurate. For more accuracy in the future, a variety of multiple models can be used. The model can be applied to the data collected in coming years to predict the premium. This can help not only people but also insurance companies to work in tandem for better and more health centric insurance.

**10) References**

* https://www.ijert.org/health-insurance-amount-prediction
* https://www.geeksforgeeks.org/ml-linear-regression/
* https://www.w3schools.com/python/python\_ml\_linear\_regression.asp
* https://www.kaggle.com/mirichoi0218/insurance
* https://www.techscience.com/cmc/v70n2/44663/pdf