

# HEALTH INSURANCE COST PREDICTION

---

Submitted By:

Tharun Aaditya

Yashika Aggarwal

MD Afroz Alam

Prathyanga

Jagatheesvaran S



# CONTENTS

---

- Introduction
- Motive of the project
- Libraries Used
- Code explanation
- Conclusion

# MOTIVE OF THE PROJECT

---

- The motive behind doing this project is that health insurance costs are to predicted so that it can give an overview to the customers that which insurance cost will be expensive for them and which off them will benefit them in true sense.
- Also the health insurance cost will depend on various factors such as age, bmi, etc. factors of the person.
- The prediction in health insurance cost will provide an overview to the insurance companies as well that which cost the consumers prefer.



# INTRODUCTION

---

- The major project is about Health Insurance Cost prediction in which there is the use of data analysis and machine learning techniques to estimate the potential medical expenses an individual or a group may incur over a specific period. This prediction is based on various factors such as age, gender, BMI, smoking habits, geographic location, and other health-related attributes.
- By leveraging data and advanced analytics, it empowers stakeholders to make informed decisions and allocate resources wisely.

# SOURCE OF THE DATA

---

- The data for the Health Insurance Cost prediction( major project) was provided by our mentor himself.
- The data includes variables such as Age, sex, BMI, Children, Smoker, Region, and Charges.
- It is a CSV file that will be imported into Python to read the file and perform various functions on it.



# LIBRARIES USED

---

- Pandas
- Numpy
- Matplotlib. pyplot
- Scikit-learn

# PANDAS

---

- Pandas is a powerful and popular Python library for data manipulation and analysis. It provides data structures and functions that make it easier to work with structured data, such as tabular data (like CSV files, Excel spreadsheets, and SQL tables) and time series data.
- Pandas allow us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science



# NUMPY

---

- NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
- NumPy arrays are stored at one continuous place in memory, unlike lists, so processes can access and manipulate them very efficiently.



# MATPLOTLIB.PYPILOT

---

- `matplotlib.pyplot` is a collection of functions that make Matplotlib work like MATLAB. Each pyplot function makes some changes to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

# Scikit-learn

---

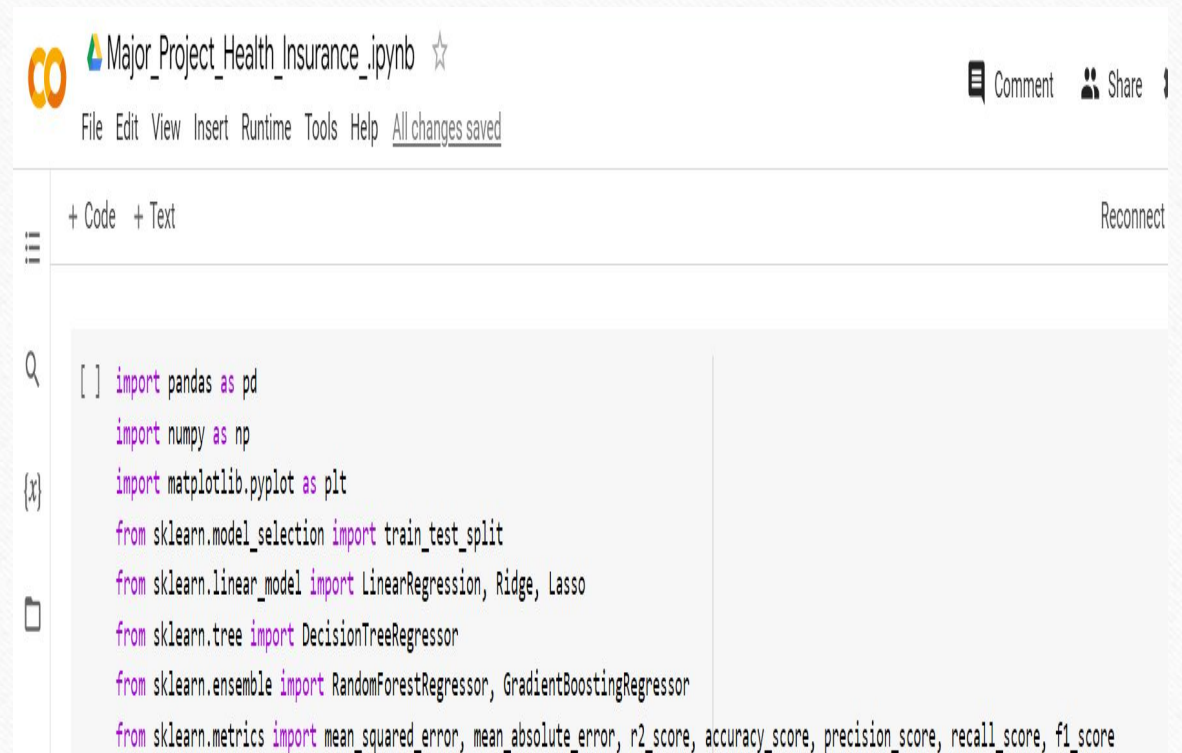
- Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistency interface in Python.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



# CODE EXPLANATION

This part of the code suggests that there are various types of libraries that need to be imported in order to implement various codes and functions in Python. Some of the libraries imported are:

- Pandas
- Numpy
- Matplotlib.pyplot
- Scikit.learn



The screenshot shows a Jupyter Notebook window titled "Major\_Project\_Health\_Insurance\_.ipynb". The interface includes a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", along with a status bar indicating "All changes saved". On the right side of the menu bar are icons for "Comment" and "Share". Below the menu bar is a toolbar with "+ Code" and "+ Text" buttons, and a "Reconnect" button on the far right. The main area of the notebook displays a code cell with the following Python code:

```
[ ] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, accuracy_score, precision_score, recall_score, f1_score
```

# CODE EXPLANATION

---

These codes explain that the dataset on which the functions are to be performed that are to be imported and then categorical variables are converted into dummy numerical variables. The predicted values are being taken into consideration upon which the values are to be predicted.

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, accuracy_score, precision_score, recall_score, f1_score
```

```
[ ] # Load the dataset
df = pd.read_csv('/content/health_insurance.csv')

# Convert categorical variables to numerical using one-hot encoding
df = pd.get_dummies(df, columns=['sex', 'smoker', 'region'])
```

```
[ ] # Split the dataset into features (X) and target variable (y)
X = df.drop('charges', axis=1)
y = df['charges']

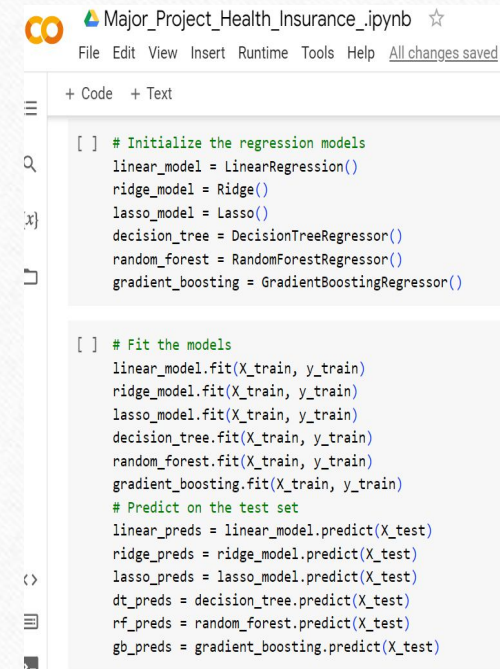
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```



# CODE EXPLANATION

Then the linear models of the predictor and the explanatory variables are formed so that there could be a linear model for the prediction of the values.

Then the linear models are fitted in such a way that the prediction of the values is easy and the error between the observed and the predicted value.



The screenshot shows a Jupyter Notebook titled "Major\_Project\_Health\_Insurance.ipynb". The code is organized into two cells. The first cell, labeled "[ ]", contains the initialization of six regression models: LinearRegression, Ridge, Lasso, DecisionTreeRegressor, RandomForestRegressor, and GradientBoostingRegressor. The second cell, also labeled "[ ]", contains the fitting of these models on training data (X\_train, y\_train) and the prediction on test data (X\_test). The predicted values are stored in variables named linear\_preds, ridge\_preds, lasso\_preds, dt\_preds, rf\_preds, and gb\_preds.

```
[ ] # Initialize the regression models
linear_model = LinearRegression()
ridge_model = Ridge()
lasso_model = Lasso()
decision_tree = DecisionTreeRegressor()
random_forest = RandomForestRegressor()
gradient_boosting = GradientBoostingRegressor()

[ ] # Fit the models
linear_model.fit(X_train, y_train)
ridge_model.fit(X_train, y_train)
lasso_model.fit(X_train, y_train)
decision_tree.fit(X_train, y_train)
random_forest.fit(X_train, y_train)
gradient_boosting.fit(X_train, y_train)

# Predict on the test set
linear_preds = linear_model.predict(X_test)
ridge_preds = ridge_model.predict(X_test)
lasso_preds = lasso_model.predict(X_test)
dt_preds = decision_tree.predict(X_test)
rf_preds = random_forest.predict(X_test)
gb_preds = gradient_boosting.predict(X_test)
```

# CODE EXPLANATION

---

- In further codes, the linear regression methods are used to predict the health insurance cost.
- Also there are various graphs and diagrams that are used for visualization of the data and the prediction of the values.
- Various machine learning methods are also used to predict the values of the health insurance costs.



# CONCLUSION

---

- Python is a great student-friendly open-source which helps to perform various programs which are also helpful in the prediction of various things.
- Python enables one to create charts and graphs for easy visualization of the data which also helps us to work with any dataset just by importing it in Python.
- Machine learning is an emerging field that helps in easy interpretation of data and it also enables one to learn various techniques of prediction as well.



A word cloud featuring the phrase "Thank You" in numerous languages and scripts. The words are arranged in a circular pattern, with "thank you" in large, bold, red letters at the center. Other prominent words include "gracias" in green, "merci" in orange, "danke" in blue, and "shukriya" in purple. Smaller words in various colors like pink, yellow, and light blue are scattered around the perimeter. The languages represented include English, Spanish, French, German, Italian, Japanese, Korean, Chinese, Hindi, Urdu, Persian, Arabic, Hebrew, Russian, Ukrainian, Polish, Czech, Slovak, Hungarian, Romanian, Bulgarian, Serbian, Croatian, Slovenian, Macedonian, Albanian, Greek, Turkish, Persian, Urdu, Hindi, Bengali, Tamil, Telugu, Malayalam, Kannada, Marathi, Gujarati, Punjabi, and many others. The background is white, and the overall shape of the word cloud is roughly circular.