**Predicting Medical Insurance Premium— Machine Learning.**

### **Predicting Medical Insurance Premium using Machine Learning**

I n this article we will apply Machine Learning various algorithm: **Linear Regression, Xgboost, Decision Tree, Random Forest Regressor** to predict Healthcare Insurance Individual premium prices.

Traditional pricing model in insurance.

Matrix pricing: — Use a small number of variables to alter the premium prices.

Some of its limitations are:

1. Leads to high uncontrollably high exposure to risk and large loss, no guarantees that are given to policy price is indeed linked to the actual risk.

2. High loss ratios due to pricing bands, issues same price regardless of whether their risks places are top of bottom of the risk brackets.

Some of the factors that are consider in pricing of individual medical pricing are.

* The age of the policy holder
* The family size of the policy holder
* The relationship ie spouse or children only
* The occupation of the policy holder.
* The geographical location.
* The age distribution of the family.
* Any existing conditions
* The sum assured — inpatient and outpatient.
* The channel of distribution

Other than the above factors that are deemed to be constant and should be considered are

* The administrative costs
* The cost of marketing and distribution ie different channel ie brokers, agency and bancassurance commission paid to them.
* Cost of capital maintain adequate reserves in case cost are higher than they except.
* Reinsurance cost in case there is such arrangements.

In matrix pricing the main factors that are used to price are age of the policy holder and the family size

X\_inpatient = Sum Assured ( age bracket) + Family Size loading.

In the above example we will use a kaggle project done on Medical Insurance Premium Prediction data. <https://www.kaggle.com/datasets/tejashvi14/medical-insurance-premium-prediction>

The objective was to calculate the annual premium paid In INR(₹) Currency And Showcases Prices For A Whole Year.

**Understanding the data :**

import pandas as pd  
**insurance** = pd.read\_csv("insurance.csv")  
insurance.head()

Table

Description automatically generated

***Note****: Regression algorithms seem to be working on features represented as numbers only…*  
By looking at our dataset we see that columns — ‘sex’, ‘smoker’ and ‘region’ are in string format, so we can work on converting them to numerical values as below

Graphical user interface, text, application

Description automatically generated

# Replacing string values to numbers  
**insurance['sex']** = insurance['sex'].apply({'male':0, 'female':1}.get) **insurance['smoker']** = insurance['smoker'].apply({'yes':1, 'no':0}.get)**insurance['region']** = insurance['region'].apply({'southwest':1, 'southeast':2, 'northwest':3, 'northeast':4}.get)

Let’s look at the dataset now,

Table

Description automatically generated

All feature variables are now in numeric format

**Exploratory Data Analysis**

Let’s create some simple plots to check out the data.

import seaborn as sns# Correlation betweeen 'charges' and 'age'   
sns.jointplot(x=insurance['age'],y=insurance['charges'])

Chart, scatter chart

Description automatically generated

*Here we see that as Age goes up Charges for health insurance also trends up*

# Correlation betweeen 'charges' and 'smoker'   
sns.jointplot(x=insurance['age'],y=insurance['charges'])

Chart

Description automatically generated

0: Non-smoker, 1: Smoker

*Here we see that charges for smokers are higher than non-smokers*

**MODELLING:**

Since the above problem is a regression we are going to use the following approach to model the data for us accurately predict the premium per annum.

* 1. Linear Regressor
  2. Cat boost Regressor
  3. Random forest regressor
  4. Boosting forest
  5. Xtreme boost forest.

**Linear Regression Algorithm :**

Simple regression equation : Y = A + B \* X

X → Input variable (training data)  
B → Coefficients of X  
A → Intercept (Constant)  
Y → Predicted Value ( Calculated from A,B and X)

Chart, line chart

Description automatically generated

Example of simple linear regression

**Let’s understand this model more clearly by doing below exercise where we apply Linear Regression to our dataset.**

Dataset ( insurance.csv) for ‘Medical Insurance’ can be downloaded from [Kaggle.](https://www.kaggle.com/mirichoi0218/insurance)

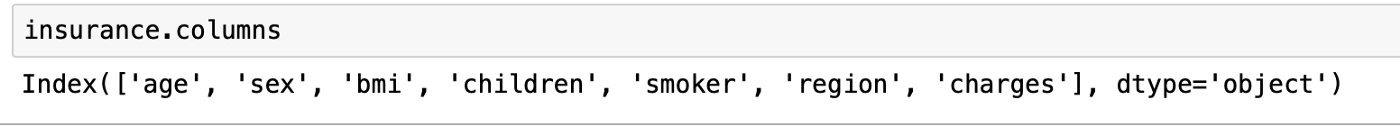
**Columns Description :**

**- Age:** Age of primary beneficiary  
**- Sex:**Primary beneficiary’s gender  
**- BMI:** Body mass index (providing an understanding of the body, weights that are relatively high or low relative to height)  
**- Children:** Number of children covered by health insurance / Number of dependents  
**- Smoker:** Smoking (yes, no)  
**- Region:** Beneficiary’s residential area in the US (northeast, southeast, southwest, northwest)  
**- Charges:** Individual medical costs billed by health insurance

**Training a Linear Regression Model**

Let’s now begin to train our regression model by following below steps

**Step 1:** First we will split our data into ‘X’ array that contains the features and a ‘y’ array with the target variable.



# features  
**X** = insurance[['age', 'sex', 'bmi', 'children','smoker','region']]# predicted variable  
**y** = insurance['charges']

Table

Description automatically generated

**Step 2:** Next we will split our dataset(insurance.csv) into a training set and a testing set. We will train our model on the training set and then use the test set to evaluate the model(Predict ‘y’ variable). *Please note that we will also compare the testing set predicted results with actual results.*

First, we will import the required libraries

# importing train\_test\_split model  
from sklearn.model\_selection import train\_test\_split# splitting train and test data  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.4)

‘test\_size’ represents that 30% of the data goes to the test data and the rest to the training set.

We can quickly check the number of records for both training dataset and testing dataset

len(X\_test) # 402  
len(X\_train) # 936  
len(insurance) # 1338

**Step 3: Train and Test the model**

Now that we have a train and test datasets, we can evaluate the model using Linear regression as below

# importing the model  
from sklearn.linear\_model import LinearRegressionmodel = LinearRegression()  
# Fit linear model by passing training dataset  
model.fit(X\_train,y\_train)

**Step 4: Predictions from our Model**

# Predicting the target variable for test datset  
predictions = model.predict(X\_test)

Here we pass ‘X\_test’ to our model to predict the ‘y\_test’

Graphical user interface, text

Description automatically generated

Predicted values

**Step 5: Comparing the results**

Let’s compare these ‘prediction’ results with actual results by plotting a graph.

import matplotlib.pyplot as pltplt.scatter(y\_test,predictions)  
plt.xlabel('Y Test')  
plt.ylabel('Predicted Y')

Chart, scatter chart

Description automatically generated

Correlation between predicted and actual results.

We can calculate ‘Root Mean Squared Error’ to check the model performance using regression evaluation metrics to see the model behavior and decide which model best fits.

Let’s just continue with our basics as of now and try to predict new customer’s insurance charges

# Predict charges for new customer : Name- Frank  
data = {'age' : 40,  
 'sex' : 1,  
 'bmi' : 45.50,  
 'children' : 4,  
 'smoker' : 1,  
 'region' : 3}  
index = [1]  
frank\_df = pd.DataFrame(data,index)  
frank\_df

Diagram

Description automatically generated

prediction\_frank = model.predict(frank\_df)  
print("Medical Insurance cost for Frank is : ",prediction\_frank)



**Conclusion**

In this article, we have explored the basics of the linear regression model and applied it to predict charges and seen the correlation between predicted and actual results. I hope you find this post useful and got some basic idea of a linear regression model.

‘

Sources.

Health Insurance Fraud Survey Report - Association of Kenya Insurers November 2013.

Pricing on Individual Health Insurance – UON Repository written by Langat Kenneth Kiprotich.