Project Report: Health Insurance Cost prediction

By :

V.Aadhav Prasannaa

Vellore Institute of Technology, Chennai

17BCE1021

Career Basic plan (September)

INTRODUCTION

The project “Health Insurance Cost prediction” predicts the cost of health insurance based on some details of person which helps to improve performance of Insurance companies and to provide better services to the customer.

An important informatics tool for controlling healthcare costs is accurately predicting the likely future healthcare costs of individuals. To address this important need, we conducted a systematic literature review and identified five methods for predicting healthcare costs. To enable a direct comparison of these different approaches, we empirically evaluated the predictive performance of each reported approach, as well as other state-of-the-art supervised learning methods, using data from University of Utah Health Plans for October 2013 to October 2016. The data set consisted of approximately 90,000 individuals, 6.3 million medical claims and 1.2 million pharmacy claims. In this comparative analysis, gradient boosting had the best predictive performance overall and for low to medium cost individuals. For high cost individuals, Artificial Neural Network (ANN) and the Ridge regression model, which have not been previously reported for use in healthcare cost prediction, had the highest performance.

LITERATURE SURVEY

Existing problem

The growing healthcare industry is generating a large volume of useful data on patient demographics, treatment plans, payment, and insurance coverage attracting the attention of clinicians and scientists alike. In recent years, a number of peer-reviewed articles have addressed different dimensions of data mining application in healthcare. However, the lack of a comprehensive and systematic narrative motivated us to construct a literature review on this topic.

With the wide usage of computers and internet, there has recently been a huge increase in availability of data that can be analyzed. Be it online sales information, website traffic, or user habits, data is generated every day. Such a large amount of data presents both a problem and an opportunity. The problem is that it is difficult for humans to analyze such large data. The opportunity is that this type of data is ideal and easy for computers to handle and process, because it is stored digitally in a well-formatted way and orderly manner, also computers can process data much faster than humans.

Proposed solution

The main aim of this project is to create a model based on statistically significant factors (independent variable) which will affect premiums charges (dependent variable) by an insurance company. In this project we are using Machine Learning algorithms for the accurate prediction.An application is also build which can be interlinked with the model, so as to view the result on UI (User Interface) based on input parameters.

THEORITICAL ANALYSIS

Block diagram

EXPERIMENTAL INVESTIGATIONS

There are six steps in experimental investigation on of a general project:
A machine learning model can be a mathematical representation of a real world processi steps to build a machine learning model:

1) Gathering the data
2) Data preprocessing
3) Researching the model that will be best for type of data
4) Training and testing the model
evaluation

RESULT

The project was to design a machine learning model to take in a ccount all the aspects of the given parameters and predict the health insurance costs of the person. The factors invovled are age , gender , BMI , number of children, whether you're a smoker and the region you're from

ADVANTAGES & DISADVANTAGES

The advantages are easy to implement, accessibility is fast, continuous Improvement, wide application, available 24x7, no human intervention needed. We can handle multi-dimensional and multi-variety data.
Whereas the disadvantages are lack of security, loss of control on data, high error susceptibility, dependence of network/providers.

APPLICATIONS

Using The Auto AI Experiment, one can build and deploy a machine learning model with sophisticated training features.This particular model can be used to predict any kind of price with a decent amount of accuracy like predicting the cost of groceries , predicting the the cost of new vehicles ,etc.

CONCLUSION

In this project the following things have been used: Python,Python Web Frame Works,Python For Data Analysis,Python For Data Visualization,Data Preprocessing Techniques,Machine Learning,Regression Algorithms

FUTURE SCOPE

As we are developing day to day there is a continuous growth of Auto AI and Machine Learning. The web application can be used to predict the cost of the health insurance accurately, precisely and efficiently instead of n number of people being involved directly or indirectly.

APPENDIX

A. Source code

Python code:

```
import numpy as np
from flask import Flask, request, jsonify, render_template
from joblib import load
app = Flask(__name__)
model= load('model.save')
trans=load('transf')
@app.route('/')
def home():
    return render_template('Frontend1.html')
@app.route('/y_predict',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
```

```
    x_test = [[x for x in request.form.values()]]
    print(x_test)
    test=trans.transform(x_test)
    test=test[:,1:]
    print(test)
    prediction = model.predict(test)
    print(prediction)
    output=prediction[0]
    return render_template('Frontend1.html', prediction_text='Charges
{}'.format(output))
if __name__ == "__main__":
    app.run(debug=True)
```

HTML frontend:

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>HEALTH INSURANCE POLICY</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
```

```
href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:
300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css')
}}">

<style>
.login{
top: 20%;
text-align: center;
font-family: "Arial Black", Gadget, sans-serif;
}
* {
  box-sizing: border-box;
}
body {
  background: linear-gradient(110deg, #fd3b3b 60%, #b4b4a5 60%);
  display: flex;
  justify-content: center;
  align-items: center;
}
input::placeholder {
  color:black;

}
label, input {
  font-family: sans-serif;
}
```

```
input[type=number], select {
  border: #000;
  padding: 12px 20px;
  padding: 10px;
  font-size: 1.2em;
  width: 100%;
}
</style>
</head>

<body>
 <div class="login">
   <br>
  <h1>Health Insurance Prediction</h1>
     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('y_predict')}}"method="post">
      <input type="number" name="age"  placeholder="How old are you
?" required="required" /><br><br>
       <select name="sex" required="required" />
      <option value="">Select Gender</option>
      <option value=0>Female</option>
        <option value=1>Male</option>
       </select><br><br>
    <input type="number" name="bmi" placeholder="Enter BMI"
required="required" /><br><br>
    <input type="number" name="children" placeholder="Number of
Children" required="required" /><br><br>
```

```
    <select name="smoker" required="required">
      <option value="">Smoker</option>
      <option value=1>Yes</option>
         <option value=0>No</option>
       </select><br><br>
    <select name="region" required="required">
      <option value="">Select Region</option>
      <option value="southwest">Southwest</option>
      <option value="southeast">Southeast</option>
      <option value="northwest">Northwest</option>
      <option value="northeast">Northeast</option>
    </select><br><br>
        <button type="submit" class="btn btn-primary btn-block
btn-large">Predict</button>
    </form>
   <br>
   <br>
   {{ prediction_text }}
 </div>
</body>
</html>
```

UI output Screenshot

The output window from the frontend HTML page looks like this: