**CHAPTER 1**

**INTRODUCTION**

* 1. **Overview**

This report discusses the result of the work done in the development of “Insurance Policy Recommender System” using machine learning. It aims at the development of the application framework for providing a platform to showcase the recommendation for an insurance policy to a customer who’s willing to purchase an insurance policy.

Machine Learning

Machine learning is a core sub-area of artificial intelligence; it enables computers to get into a mode of self-learning without being explicitly programmed. When exposed to new data, these computer programs are enabled to learn, grow, change, and develop by themselves. The formal definition according to the scientist of US, “Machine learning is a method of data analysis that automates analytical model building.” In other words, it allows computers to find insightful information without being programmed where to look for a particular piece of information; instead, it does this by using algorithms that iteratively learn from data.

While the concept of machine learning has been around for a long time, (an early and notable example: Alan Turing’s famous WWII Enigma Machine) the ability to apply complex mathematical calculations to big data automatically—iteratively and quickly—has been gaining momentum over the last several years.

So, put simply, the iterative aspect of machine learning is the ability to adapt to new data independently. This is possible as programs learn from previous computations and use “pattern recognition” to produce reliable results.

* 1. **Objective**

The goal of the project is

• Providing an insurance policy recommendation system to a customer based on customers details.

* 1. **Methodology**

With the help of this system, a customer who is willing to buy an insurance policy is given a recommendation based on the previous train data provided to the model. The model accepts customer data and analysis the train data to return an appropriate policy suited for the customer. By doing so, this system aims at providing an easy and less time-consuming approach for insurance policy purchasing.

This model has been implemented using a decision tree algorithm which helps to find a suitable insurance policy for the customer.

**INPUT**

(Customer information)

**DECISION TREE**

Fig 1.1

Flow Diagram Insurance Policy Recommender System

**CHAPTER 2**

**TOOLS AND REQUIREMENTS**

* 1. **Software Requirements**
* Platform: Anaconda
* Python development environment: Spyder or Jupyter
* Web Browser: Chrome
* Operating System : Windows 8

**2.1.1 Anaconda**

**Anaconda** is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system *conda*.

**2.1.2 Spyder**

**Spyder** is a n open source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including Numpy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open source software. It is released under the MIT license.

**2.1.3 Jupyter**

The **Jupyter Notebook** is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

**2.1.4 Web Browser**

A web browser is a software program that allows a user to locate, access, and display web pages. In common usage, a web browser is usually shortened to "browser." Browsers are used primarily for displaying and accessing websites on the internet, as well as other content created using languages such as Hypertext Markup Language (HTML) and Extensible Markup Language (XML).

Browsers translate web pages and websites delivered using Hypertext Transfer Protocol (HTTP) into human-readable content. They also have the ability to display other protocols and prefixes, such as secure HTTP (HTTPS), File Transfer Protocol (FTP), email handling (mailto:), and files (file:). In addition, most browsers also support external plug-ins required to display active content, such as in-page video, audio and game content.

* 1. **Hardware Requirements**
* CPU : 32-bit or 64-bit.
* RAM : Minimum 4GB,
* PROCESSOR: Intel Pentium 4 or above.

**CHAPTER 3**

**IMPLEMENTATION**

* 1. **Implementation**

The project is mainly based on Machine learning. The insurance policy recommender system recommends a suitable insurance policy based on the customer information using the decision tree algorithm. A dataset with customer details is maintained in order to train the model for an accurate recommendation. The dataset consists of customer’s basic information like age and average bank balance.

Table 3.1 gives a brief about the attributes used for the project

|  |  |
| --- | --- |
| Age | Age of the customer |
| Average Bank Balance | Average Bank Balance of the customer |
| Policy | Policy chosen |

Table 3.1 Attributes in the dataset

Decision Tree Algorithm

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

The possible solutions to a given problem emerge as the leaves of a tree, each node representing a point of deliberation and decision. The decision tree in this model splits on the basis of age and average bank balance of the customer.

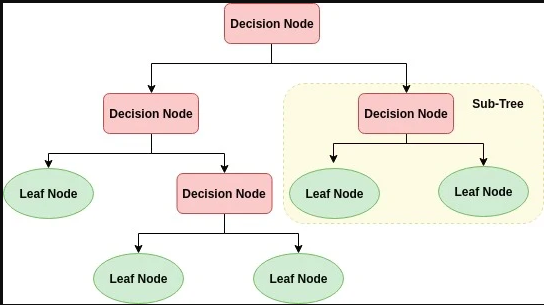


Fig 3.1 Decision Tree

* 1. **Source Code**

**Machine Learning Python Code**

#importing packages

import numpy as np

import pandas as pd

#Reading csv file

dataset=pd.read\_csv("C:\\Users\\acer-pc\\Desktop\\!Desktop\\Insurance Policy Recommender System\\policy.csv")

dataset

#seperating dependent and independent columns

X=dataset.iloc[:,[0,1]].values

X

y=dataset.iloc[:,2].values

y

#Splitting dataset into test data and train data

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1)

#Fitting DecisionTreeClassifier

from sklearn.tree import DecisionTreeClassifier

model=DecisionTreeClassifier()

model.fit(X\_train,y\_train)

#predicting for x\_test data

y\_pred = model.predict(X\_test)

# Model Accuracy, how often is the classifier correct?

from sklearn import metrics

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

#Predicting for custom inputs

predictions=model.predict([[22,35000],[26,50000]])

predictions

**HTML file Insurance.html**

<html>

<body style="background-color: #5CDB95;">

<style type="text/css">

label

{

font-family: cursive;

}

.ins-from{

max-width: 450px;

background: #FAFAFA;

padding: 30px;

margin: 50px auto;

box-shadow: 1px 1px 25px rgba(0, 0, 0, 0.35);

border-radius: 10px;

border: 6px solid #305A72;

}

.ins-from form{

padding:0;

margin:0;

list-style:none;

}

.ins-from div{

display: block;

margin-bottom: 10px;

min-height: 35px;

}

.ins-from div .field-style{

box-sizing: border-box;

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

padding: 8px;

outline: none;

border: 1px solid #B0CFE0;

-webkit-transition: all 0.30s ease-in-out;

-moz-transition: all 0.30s ease-in-out;

-ms-transition: all 0.30s ease-in-out;

-o-transition: all 0.30s ease-in-out;

}.ins-from div .field-style:focus{

box-shadow: 0 0 5px #B0CFE0;

border:1px solid #B0CFE0;

}

.ins-from div .field-split{

width: 49%;

}

.ins-from div .field-full{

width: 100%;

}

.ins-from div input.align-left{

float:left;

}

.ins-from div input.align-right{

float:right;

}

.ins-from div input[type="button"],

.ins-from div input[type="submit"] {

-moz-box-shadow: inset 0px 1px 0px 0px #3985B1;

-webkit-box-shadow: inset 0px 1px 0px 0px #3985B1;

box-shadow: inset 0px 1px 0px 0px #3985B1;

background-color: #216288;

border: 1px solid #17445E;

display: inline-block;

cursor: pointer;

color: #FFFFFF;

padding: 8px 18px;

text-decoration: none;

font: 12px Arial, Helvetica, sans-serif;

}

.ins-from div input[type="button"]:hover,

.ins-from div input[type="submit"]:hover {

background: linear-gradient(to bottom, #2D77A2 5%, #337DA8 100%);

background-color: #28739E;

}

</style>

<form class="ins-from" style="float: right;margin-top:100px; margin-right: 90px; padding: 60px ;" method="POST">{%csrf\_token%}

<div>

<h2>Insurance Policy Recommender System</h2>

</div>

<div>

<label> Enter Your Age </label>

<input type="number" name="age" class="field-style field-split align-right" min=20 max=26>

</div>

<div>

<label>Enter your average savings </label>

<input type="number" name="amount" class="field-style field-split align-right" min=20000 max=90000>

</div>

<div>

<input type="submit" value="Submit" style="float: right;margin-top:10px; margin-right: 120px; border-radius: 2em; width:100px;font-family: cursive;font-size: 20px; " >

</div>

</form>

<div style="float: left; padding-top: 90px; ">

<h1 style="font-size: 50px; color: white; padding-left: 30px; font-family:Gill Sans Ultra Bold Condensed;"> <strong> WISE </strong></h1>

<h1 style="font-size: 50px; color: white; padding-left: 30px; font-family:Gill Sans Ultra Bold Condensed;"> <strong> FINANCIAL THINKING </strong></h1>

<h1 style="font-size: 50px; color: white; padding-left: 30px; font-family:Gill Sans Ultra Bold Condensed;"> <strong> FOR A HEALTHY </strong></h1>

<h1 style="font-size: 50px; color: white; padding-left: 30px; font-family:Gill Sans Ultra Bold Condensed;"> <strong> LIFE </strong></h1>

</div>

<div>

<h1 style="font-size: 50px; color:#05386b; padding-left: 30px; font-family:Gill Sans Ultra Bold Condensed;"> <strong> Our Recommendation: {{msg}}</strong></h1>

</div>

</body>

</html>

**CHAPTER 4**

**RESULTS**

**4.1 Jupyter Notebook**

Figure 4.1 shows the implementation of machine learning on jupyter notebook.

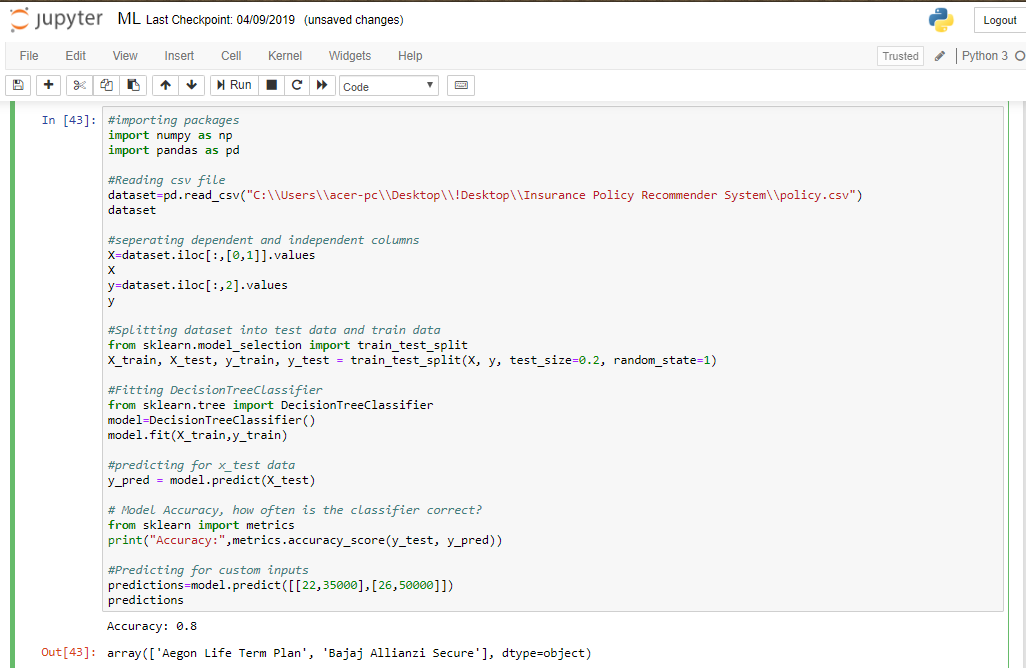


Fig 4.1 Jupyter Notebook

Figure 4.2 shows the accuracy of the algorithm and also the recommendation for the given inputs. i.e person X with age 22 having average bank balance of Rs 35,000 and person Y with age 26 having average bank balance of Rs.50,000.

Screenshot (88).png

Fig 4.2 Accuracy and Output in Jupyter Notebook

**4.2 Interface**

Figure 4.3 shows the interface for easy interaction with the model.



Fig 4.3 Interface

Figure 4.4 and 4.5 shows recommended policies for the given inputs.



Fig 4.4 Recommendation – 1



Fig 4.5 Recommendation – 2

Figure 4.6 is the custom dataset created with the help of the information in the policy bazaar website.

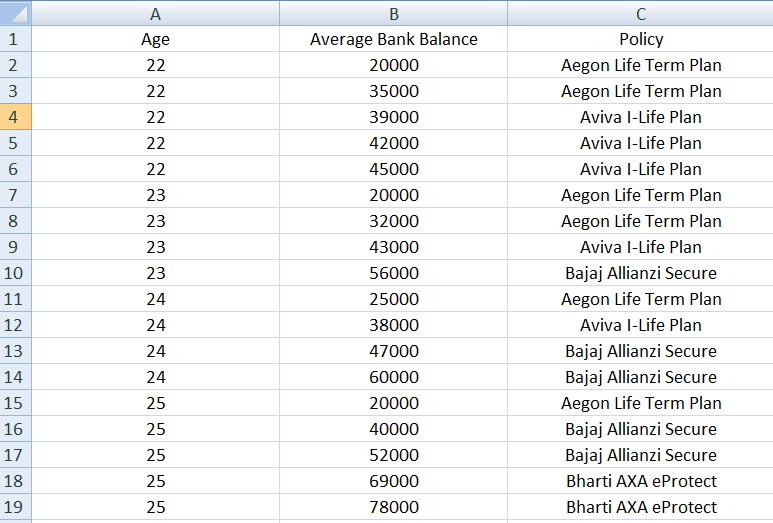


Fig 4.6 Policy.csv Dataset

**CHAPTER 5**

**CONCLUSIONS, LIMITATIONS AND FUTURE SCOPE**

* 1. **Conclusions**

The project Insurance Policy Recommender System overall simplifies the process of the recommendation of appropriate insurance policies to the customer. Customers can easily access the website and get a recommendation based on their age and their average bank balance.

The Insurance Policy Recommender System helps insurer or the customer by recommending a suitable insurance policy based on the customer’s information thereby saving extra efforts and time.

* 1. **Limitations**

Some of the limitations of this model are:

1. The model recommends insurance policy using only two attributes. Hence making it a little less reliable.
2. The model is not designed to provide multiple recommendations to the customer.
   1. **Future Scope**

The limitations of this model point towards the areas that need to be addressed in the future.

1. The model can be implemented on a large scale if the dataset is improved.
2. The model mainly finds its place in the banking system as the banks have tie-ups with insurance policy companies and also have relevant data needed for this model for a reliable prediction.
3. The model can be used in any insurance policy e-commerce website to recommend suitable insurance policy to its users.

**REFERENCES**

[1]. Sumit Das, Aritra Dey.et.al., “Applications of Artificial Intelligence in Machine Learning: Review and Prospect”, International Journal of computer applications,vol 15, issue 9, Apr2015,pp 31-39

# [2]. A. Navada, A. N. Ansari, S. Patil and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," 2011 IEEE Control and System Graduate Research Colloquium, Shah Alam, 2011, pp. 37-42.

[3]. A. Shamim, H. Hussain and Maqbool Uddin Shaikh, "A framework for generation of rules from decision tree and decision table," 2010 International Conference on Information and Emerging Technologies, Karachi, 2010, pp. 1-6.

doi: 10.1109/ICIET.2010.5625700

[4]. M. Ross, C. A. Graves, J. W. Campbell and J. H. Kim, "Using Support Vector Machines to Classify Student Attentiveness for the Development of Personalized Learning Systems," 2013 12th International Conference on Machine Learning and Applications, Miami, FL, 2013, pp. 325-328.

doi: 10.1109/ICMLA.2013.66

[5]. Yi Tan and Guo-Ji Zhang, "The application of machine learning algorithm in underwriting process," 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 2005, pp. 3523-3527 Vol. 6.

doi: 10.1109/ICMLC.2005.1527552

[6]. I. N. Gluhih, I. Y. Karyakin and L. V. Sizova, "Recommender system providing recommendations for unidentified users of a commerial website," 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), Baku, 2016, pp. 1-3.

doi: 10.1109/ICAICT.2016.7991724