MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Insurance Premium Prediction

Submitted By:

Yash Mathur

0901CS191142

Faculty Mentor:

Mr. Mir Shahnawaz Ahmad

Assistant Professor, Computer Science and Engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Insurance Premium Prediction

A project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Yash Mathur

0901CS191142

Faculty Mentor:

Mr. Mir Shahnawaz Ahmad

Assistant Professor, Computer Science and Engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **YashMathur**(0901CS191142) has submitted the project report titled **Insurance Premium Prediction** under the mentorship of **Mr.MirShahnawaz Ahmad**, in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

07/05/22

Mr.MirShahnawaz Ahmad
Faculty Mentor
Assistant Professor
Computer Science and Engineering

09/05/22

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S.

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Mr.MirShahnawaz Ahmad, Assistant Professor,** Computer Science and Engineering.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

YashMathur
0901CS191142
3rd Year,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr.ManjareePandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Mr.MirShahnawaz Ahmad**, Assistant Professor, Computer Science and Engineering, for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

YashMathur
0901CS191142
3rd Year,
Computer Science and Engineering

ABSTRACT

Machine learning deployment is the process of deploying a machine learning model in a live environment. The model can be deployed across a range of different environments and will often be integrated with apps through an API. Deployment is a key step in an organisation gaining operational value from machine learning.

Machine learning models will usually be developed in an offline or local environment, so will need to be deployed to be used with live data. A data scientist may create many different models, some of which never make it to the deployment stage. Developing these models can be very resource intensive. Deployment is the final step for an organisation to start generating a return on investment for the organisation.

However, deployment from a local environment to a real-world application can be complex. Models may need specific infrastructure and will need to be closely monitored to ensure ongoing effectiveness. For this reason, machine learning deployment must be properly managed so it's efficient and streamlined.

सारः

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: INTRODUCTION

1.1 Dataset Link:

The insurance.csv dataset contains 1338 observations (rows) and 7 features (columns). The dataset contains 4 numerical features (age, bmi, children and expenses) and 3 nominal features (sex, smoker and region) that were converted into factors with numerical value designated for each level.

The purposes of this purpose to look into different features to observe their relationship, and plot a multiple linear regression based on several features of individual such as age, physical/family condition and location against their existing medical expense to be used for predicting future medical expenses of individuals that help medical insurance to make decision on charging the premium.

1.2 Python Libraries:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms.

Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier.

Chapter 2: Exploratory Data Analysis

In statistics, exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling and thereby contrasts traditional hypothesis testing. Exploratory data analysis has been promoted by John Tukey since 1970 to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA), which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

There are a number of tools that are useful for EDA, but EDA is characterized more by the attitude taken than by particular techniques.

Typical graphical techniques used in EDA are:

- Box plot
- Histogram
- Bar chart

The objectives of EDA are to:

- Enable unexpected discoveries in the data
- Suggest hypotheses about the causes of observed phenomena
- Assess assumptions on which statistical inference will be based
- Support the selection of appropriate statistical tools and techniques
- Provide a basis for further data collection through surveys or experiment

In statistics, exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling and thereby contrasts traditional hypothesis testing. Exploratory data analysis has been promoted by John Tukey since 1970 to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA), which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

2.1 Categorical Variables:

Categorical variables are qualitative data in which the values are assigned to a set of distinct groups or categories. These groups may consist of alphabetic (e.g., male, female) or numeric labels (e.g., male = 0, female = 1) that do not contain mathematical information beyond the frequency counts related to group membership. Instead, categorical variables often provide valuable social-oriented information that is not quantitative by nature (e.g., hair color, religion, ethnic group).

In the hierarchy of measurement levels, categorical variables are associated with the two lowest variable classification orders, nominal or ordinal scales, depending on whether the variable groups exhibit an intrinsic ranking. A nominal measurement level consists purely of categorical variables that have no ordered structure for intergroup comparison.

	age	sex	bmi	children	smoker	region	expenses
0	19	female	27.9	0	yes	southwest	16884.92
1	18	male	33.8	1	no	southeast	1725.55
2	28	male	33.0	3	no	southeast	4449.46
3	33	male	22.7	0	no	northwest	21984.47
4	32	male	28.9	0	no	northwest	3866.86

Fig 2.1.1 Dataframe

Categorical variables:

Sex

Smoker

Region

Sex –

It is basically a gender of the person who want to predict his insurance premium.

To analyse the columns, we can draw countplot or pie chart.

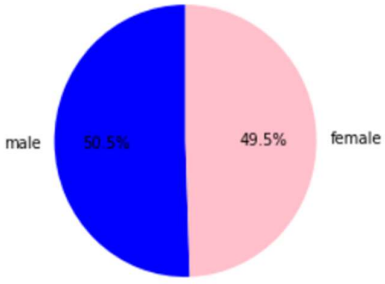

Fig 2.1.2: Distribution of sex

Smoker-

It tells whether the person smokes or not.

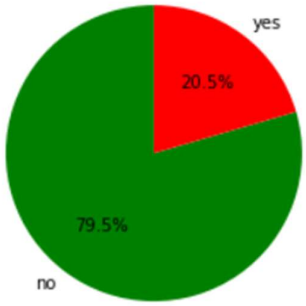

Fig 2.1.3: Distribution of Smoker

Region-

It tells the region where the person belongs to.

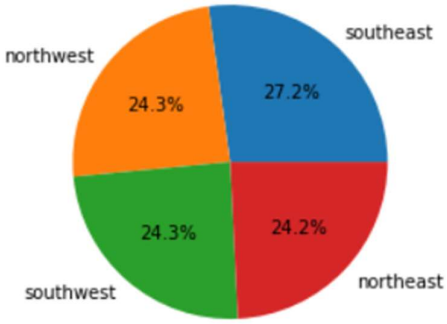

Fig 2.1.4: Distribution of Region

2.2 Numerical variables:

Numerical variables have values that describe a measurable quantity as a number, like 'how many' or 'how much'. Numeric variables are also called quantitative variables; the data collected containing numeric variables are called quantitative data. Numeric variables may be further described as either continuous or discrete:

- Continuous numeric variable: Observations can take any value between a certain set of real numbers, i.e. numbers represented with decimals. This set is typically either "every possible number" (e.g. the change in population density can be positive or negative, and very large or very small) or "all the positive numbers" (e.g. biomass may be very large or very small, but it is strictly positive). Examples of continuous variables include body mass, age, time, and temperature. Though in theory continuous variables may admit any number in the set of possible numbers, in practice the values given to an observation may be bounded and can only include values as small as the measurement protocol allows. Elephants are very large, but they never get as big as a passenger jet, and trying to measure the mass of an elephant at a precision of a few grams is probably not practical.
- Discrete numeric variable: Observations can take a value based on a count from a set of whole values; e.g. 1, 2, 3, 4, 5, and so on. A discrete variable cannot take the value of a fraction between one value and the next closest value. Examples of discrete variables include the number of individuals in a population, number of offspring produced ('reproductive fitness'), and number of infected individuals in an experiment. All of these are measured as whole units. Discrete variables are very common in the biological and environmental sciences.

Numerical Variables:

Age

bmi

children

expenses

To analyse the distribution of the numerical values, we use distplot, countplot.

Age: It tells the age distribution in the dataset.

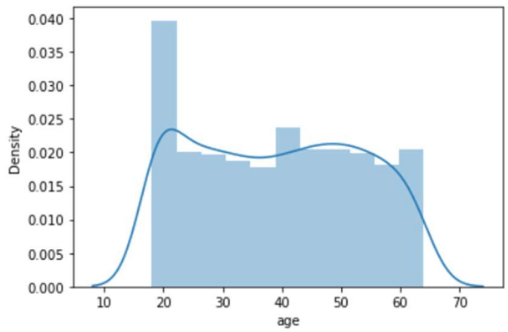

Fig 2.2.1:Distribution of age

Bmi: It tells the distribution of bmi in the dataset.

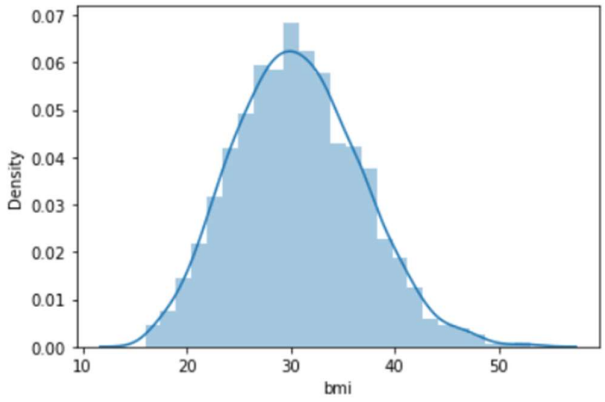

Fig 2.2.2: Distribution of bmi

Children: It tells the distribution of applicant having children.

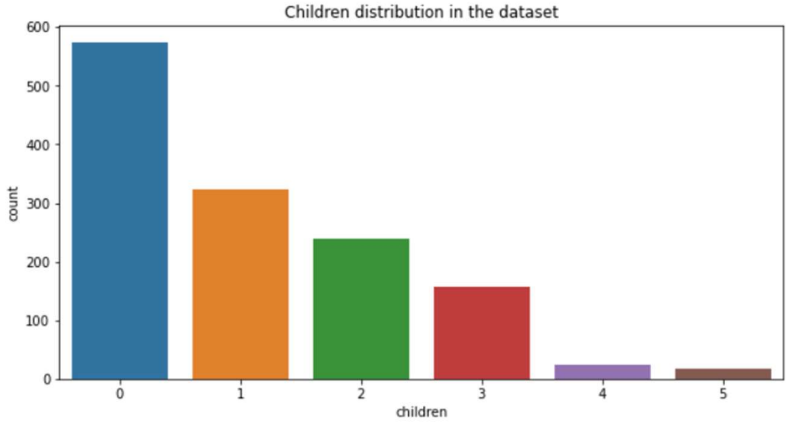

Fig 2.2.3: Distribution of children

Expenses: It tells the distribution of expenses.

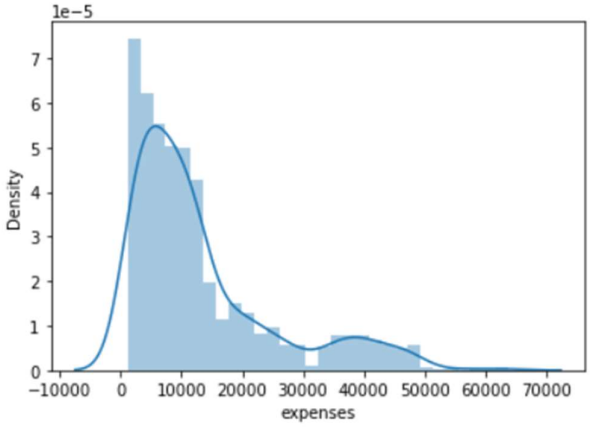

Fig 2.2.4: Distribution of expenses

Chapter 3: Machine Learning Deployment

The simplest way to deploy a machine learning model is to create a web service for prediction. In this project, we use the Flask web framework to wrap a simple random forest classifier built with scikit-learn.To create a machine learning web service, you need at least three steps.The first step is to create a machine learning model, train it and validate its performance. In the next step, we need to persist the model. The environment where we deploy the application is often different from where we train them. Training usually requires a different set of resources. Thus this separation helps organizations optimize their budget and efforts.

Scikit-learn offers python specific serialization that makes model persistence and restoration effortless.

Finally, we can serve the persisted model using a web framework.

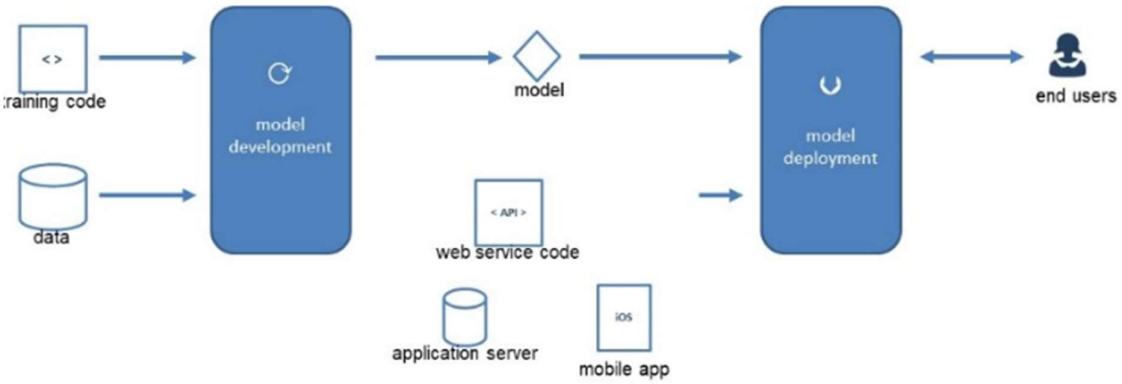

Fig 3.1: Machine Learning Deloyment

Insurance Premium Prediction

Age: 19
Sex: Female
bmi: 27.9
Children: 0
Smoker: No
Region: Northeast
Predict

The predicted medical expenses is [2827.66895].

3.1. Deployment on Heroku:

One of the most prevalent misunderstandings and mistakes for a failed ML project is spending a significant amount of time optimizing the ML model. Instead, teams that have completed a successful machine learning project devote time to gathering data, developing efficient data pipelines to reduce training-serving skew, and constructing dependable model serving infrastructure. The diagram below depicts the stages of machine learning development.

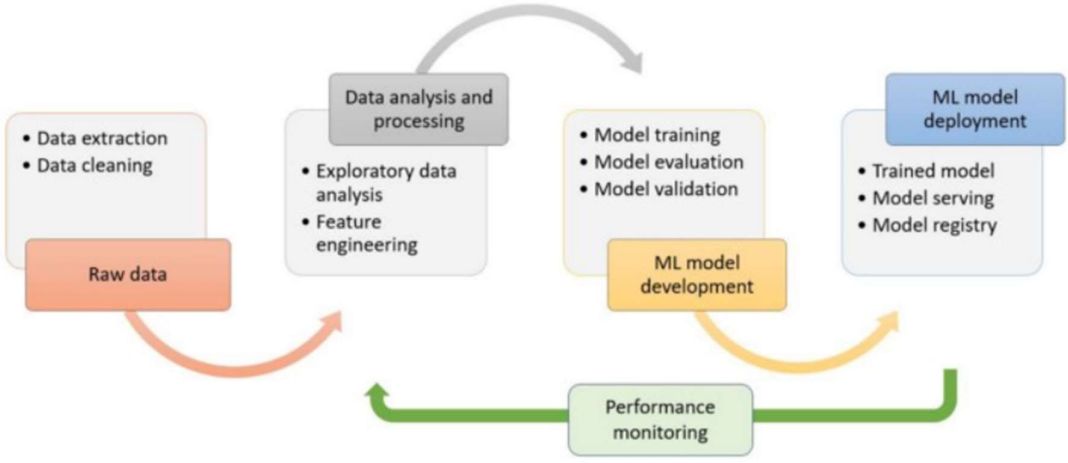

Fig 3.1.1: Machine learning project lifecycle

We want the user to interact with our webpage, as opposed to a static website where the user merely reads the content. Web applications are websites that have functionality and an interactive element.

In our scenario, a user will have six input fields, and we provide a prediction to the user by collecting sample data. Before that, we must first comprehend how a browser interacts with other online pages by making requests. A standard browser will submit a GET request using the HTTP protocol and a POST request to send data through the web page. Hopefully, you will figure it out along the way.

To handle user requests, we have to create a lot of code as Python programmers. Instead of developing repeated code to handle requests, we can use flask.Flask is a web server framework that requires us to organize our code in a specific way. Begin by creating two folders in your MyApp directory called Models and Templates.

templates directory. The fact that the app.py and index.html files are empty is unimportant; instead, concentrate on the framework.

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market. Heroku is fully maintained, allowing developers to focus on their core product rather than having to worry about servers, hardware, or infrastructure. It provides tools, services, workflows, and support for polyglot—all intended to improve developer productivity.

We're ready to start our Heroku deployment now that our model has been trained, the machine learning pipeline has been set up, and the application has been tested locally. There are a few ways to upload your application source code onto Heroku. The easiest way is to link a GitHub repository to your Heroku account.

Although flask is fantastic, they help in local development. It does not handle the kind of queries that a typical web server does. We'll need to install the gunicorn python library to take care of a large number of requests.

We need to tell Heroku to use the gunicorn now that we've installed it. We accomplish this by generating a file called procfile that has no extension (for instance, Procfile.txt is not valid.). The file consists of commands to execute on the startup.

It's just one line of code that tells a web server which files to run first when someone logs into the application. In this case, the name of our application file is app.py, and the name of the application is also app.

requirements.txt -*It is a text file containing the python packages required to execute the application. If these packages do not found in the environment application is running, it will fail. I recommend you mention the particular versions of all your libraries.*

Heroku link-https:/insurancepremiumbyyash.herokuapp.com/

Code- https://github.com/yashmathur0310/Insurance-Premium-prediction

Chapter 4: Conclusion

We have performed the exploratory data analysis on the dataset and learned how to deal with categorical variables, understood the distribution of the variables. Learned how to make HTML form and connect it with flask. Learned to implement machine learning algorithm and deploy the model on Heroku.

References –

https://www.analyticsvidhya.com/blog/2021/10/a-complete-guide-on-machine-learning-model-deployment-using-heroku/

https://www.youtube.com/watch?v=mrExsjcvF4o

https://www.dataminingapps.com/2019/10/some-thoughts-on-deploying-machine-learning-models/