Detailed Project Report

Project On
Insurance Premium Prediction
By:- Siridi Nath Peddina

Detailed Project Report

Contents

Detailed Project Report

Abstract

This project represents a machine learning-based health insurance prediction system. Recently, many attempts have been made to solve this problem, as after the Covid-19 pandemic, health insurance has become one of the most prominent areas of research. We have used the USA's medical cost personal dataset from Kaggle, having 1338 entries. Features in the dataset that are used for the prediction of insurance cost include Age, Gender, BMI, Smoking Habit, number of children etc. We used Xgboost regression and determined the relation between price and these features. We trained the system using a 80-20 split and achieved an accuracy of 85%.

Detailed Project Report

Introduction

Importance of DPR Documentation?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and written code. This also provides the detailed description on how the entire project has been designed end to end.

Key Points:

- Describes the Design flow
- Implementation
- Software requirements
- Architecture of project
- Non-functional attributes like:
    - Reusability
    - Portability
    - Resource utilization

Detailed Project Report

1. Description.

1. Problem Perspective.

The Insurance Premium Prediction is a hyper-tuned Machine Learning Regression model which helps to determine the premium for various policyholders on different parameters.
Parameters such as :- age, sex, bmi,etc.

2. Problem Statement.

Insurance Premium Prediction is a model that predicts the premium for various policyholders depending on different parameters. The HealthCare Industry is one of the prime sectors in the Insurance Industry. Also, additionally it helps the Insurance Company to revise their policy plans structure and henceforth help them to revise the premium & serve the policyholders more efficiently.
The basic idea of this system is to predict premiums on various factors & help policyholders choose a plan that is more suitable for them and importantly more economic.

3. Proposed Solution.

The solution proposed is to take the required batch file to predict the result. A pipeline has been created to get the prediction for the new dataset. A Streamlit webapp has been created to get the prediction based on certain inputs.

4. Solution Improvements.

The system can be made more futuristic by performing more hyper-tuning methods so that the prediction can be more accurately predictive. The project code has been designed in such a way that whenever new data will come, the model will go under training and if there will be an improvement in the model then the new model will be used for prediction.

2. Requirements

2.1Hardware Requirements:-

- A working computer to code with an active internet connection.

2.2 Tools / Software Requirements:-

- Python version used for this project 3.10 (This may get updated and some features might not be available in new version.)
- Python libraries such as NumPy, Pandas, Matplotlib, Seaborn and scikit-learn (Used for implementation of machine learning algorithms)
- Jupyter Notebook & Visual studio code is used as an IDE for writing the code.
- Github is used as the version control system.
- AWS is used for deployment using docker image.

3. Data Requirement.

Whenever we are working on any project the data is completely dependent on the requirement of the problem statement. For this project, the problem statement was to create a Hyper tuned Regression machine learning model which can predict the insurance premium based on various parameters.

3.1 Data Collection.

The data which is used in this project was taken from Kaggle.

Dataset link: *https://www.kaggle.com/datasets/noordeen/insurance-premium-prediction*

3.2 About the dataset.

File Contain.

- Readme.txt

insurance _dataset.csv - contains 1338 rows and 7 columns

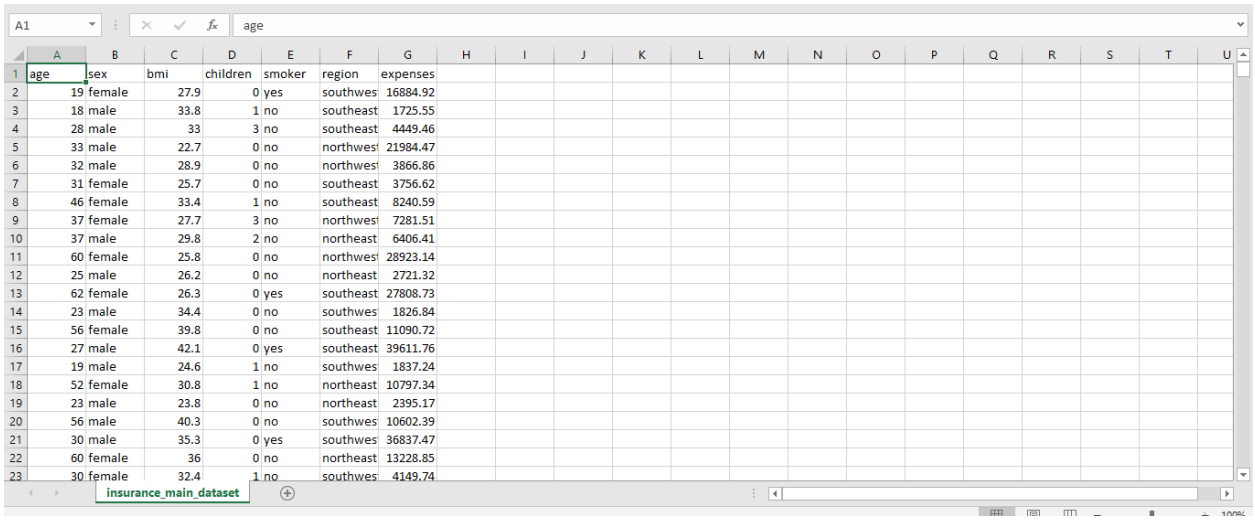
3.3 Dataset Characteristic.

insurance _dataset.csv have the following fields:

- **age**: contains the age of the policyholders from various types of policies.
- **sex** : determines the Gender of the population taking a particular policy
- **bmi**: BMI being the most important factor in the domain of Health Insurance, depending upon which the insurance company decides the premium to be paid by the policyholder.
- **children:** It is taken as no. of beneficiaries added for a particular insurance of a policyholder.
- **smoker**: Like BMI, the smoking parameter is an important factor for calculating the premium of a particular policy for a policyholder.
- **region**: Various insurance companies divide their serving in the form of regions, which in turn decides the copay or any extra premium to be given by the policyholder.
- **expenses**: this is the total premium paid by the policyholder for a given policy schedule.

3.4 License.

insurance_dataset.csv file is obtained from the Machine Learning course website (Spring 2017) from Professor Eric Suess at http://www.sci.csueastbay.edu/~esuess/stat6620/#week-6.



age	sex	bmi	children	smoker	region	expenses
19	female	27.9	0	yes	southwes	16884.92
18	male	33.8	1	no	southeast	1725.55
28	male	33	3	no	southeast	4449.46
33	male	22.7	0	no	northwest	21984.47
32	male	28.9	0	no	northwest	3866.86
31	female	25.7	0	no	southeast	3756.62
46	female	33.4	1	no	southeast	8240.59
37	female	27.7	3	no	northwest	7281.51
37	male	29.8	2	no	northeast	6406.41
60	female	25.8	0	no	northwest	28923.14
25	male	26.2	0	no	northeast	2721.32
62	female	26.3	0	yes	southeast	27808.73
23	male	34.4	0	no	southwes	1826.84
56	female	39.8	0	no	southeast	11090.72
27	male	42.1	0	yes	southeast	39611.76
19	male	24.6	1	no	southwes	1837.24
52	female	30.8	1	no	northeast	10797.34
23	male	23.8	0	no	northeast	2395.17
56	male	40.3	0	no	southwes	10602.39
30	male	35.3	0	yes	southwes	36837.47
60	female	36	0	no	northeast	13228.85
30	female	32.4	1	no	southwes	4149.74

1.1 Representation of insurance_main_dataset.csv file

4. Data Pre-processing.

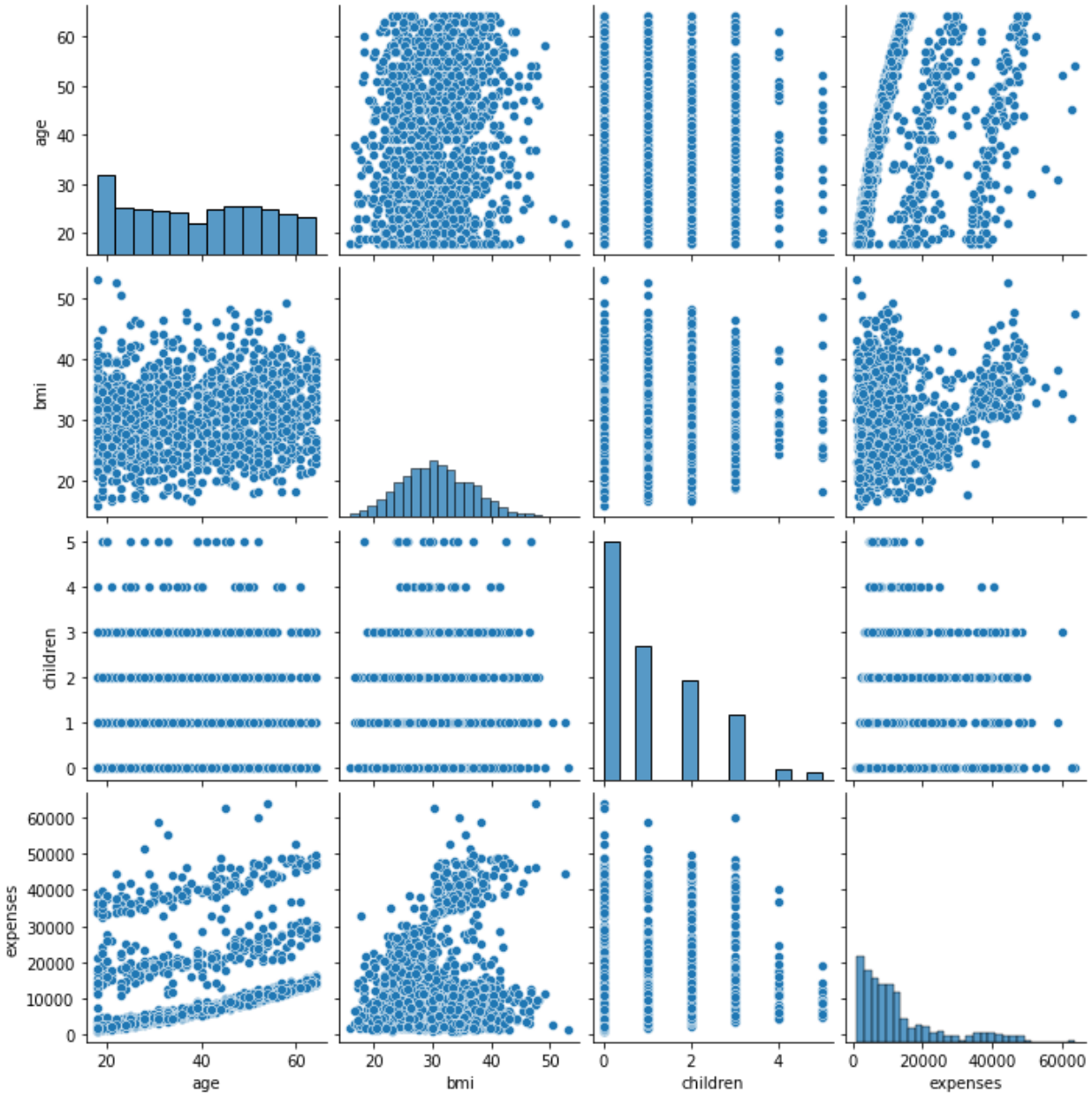
Have taken the insurance_main.csv file as my dataset

- All the necessary libraries were imported first such as Numpy, Pandas, Matplotlib, and Seaborn.
- Checking the basic profile of the dataset. To get a better understanding of the dataset.
  - Using Info method
  - Using Describe method
  - Checking for unique values of each column.
- Checking for null values, There are no null values present in our dataset.
- Used Matplotlib to plot the basic graphs which is described in the below sections separately

1.2 Graphical Distribution of all the columns.

Pair plot:

A pair plot is a type of plot that allows you to visualize relationships between multiple variables in a dataset. It is a useful tool for exploring and understanding the relationships between different features in your data.

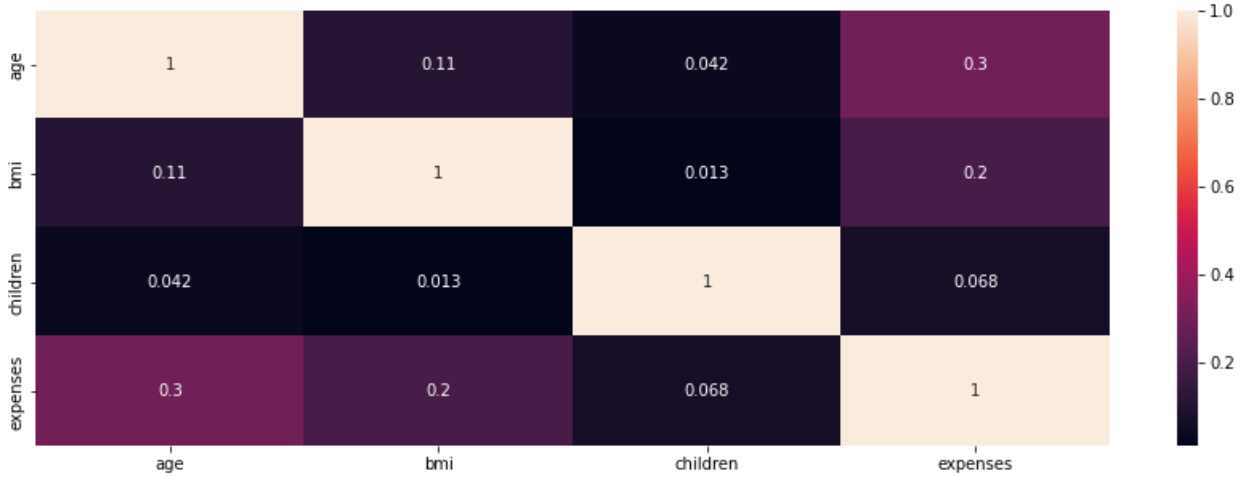


Observation:

- "expenses " is increasing wrt "bmi" and "age"
- "expenses " is decreasing wrt "children"

Heatmap:

A heatmap is a graphical representation of data where the values are represented as colours. It is a useful visualization tool for understanding the relationships between different data values, and for identifying patterns and trends in the data.

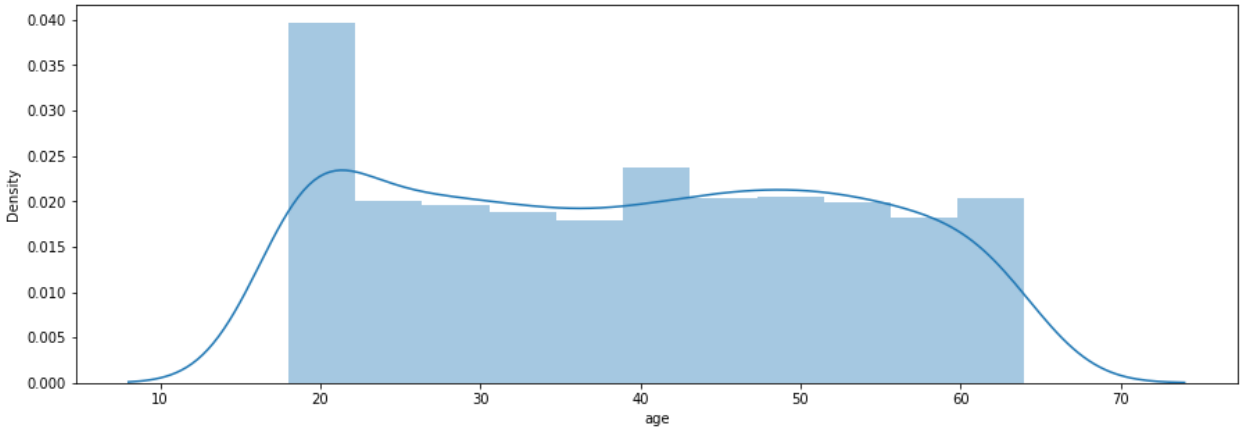


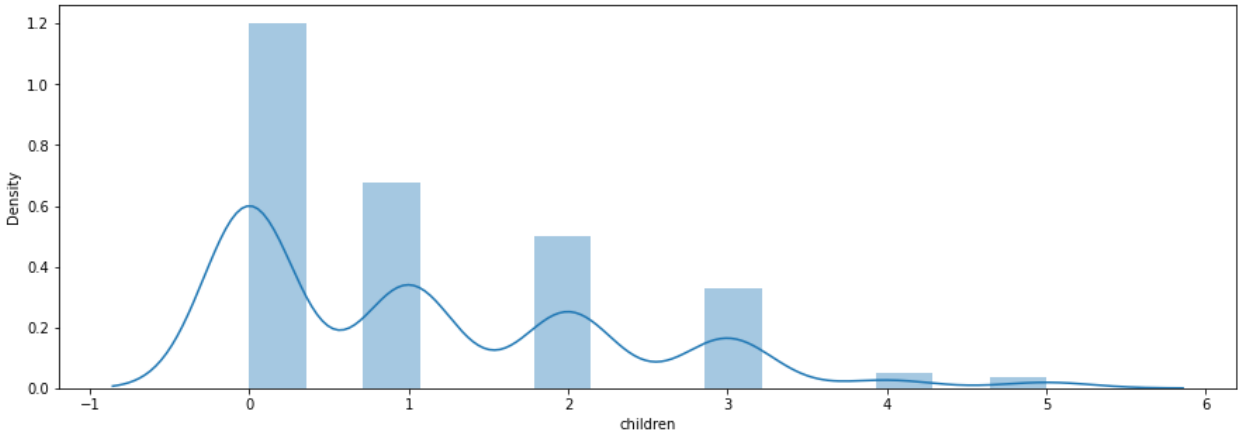
Observation:

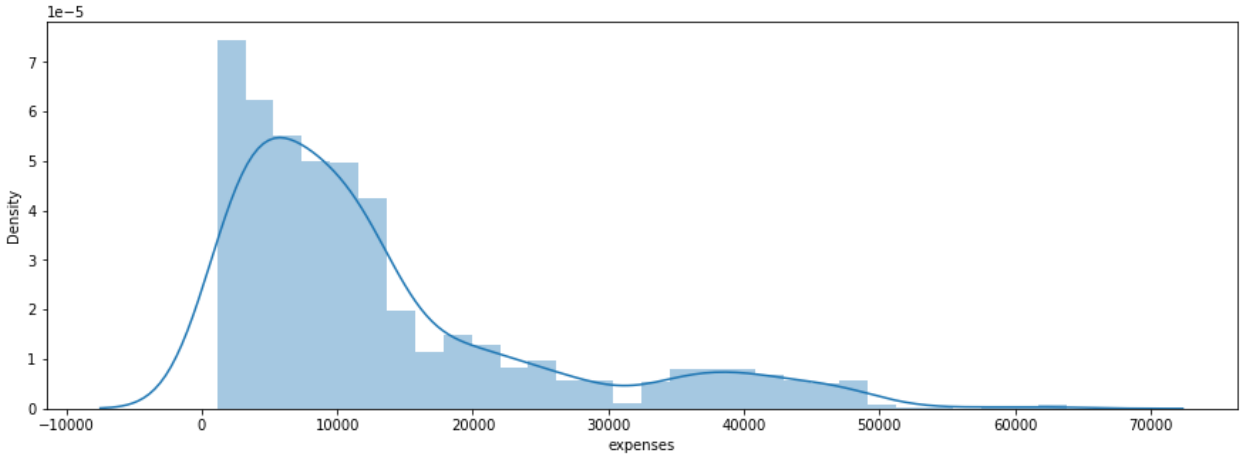
- None of the variables are highly correlated

Distplots:

Distplots are useful for visualizing the distribution of a dataset and understanding the underlying patterns and trends in the data. They can be used to identify the presence of outliers in the data, and to compare the distribution of different datasets.





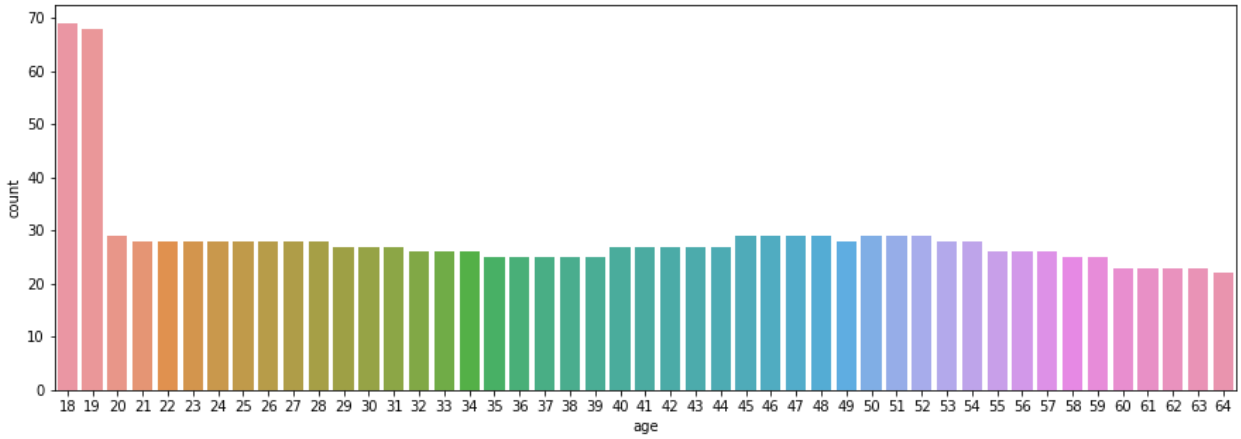


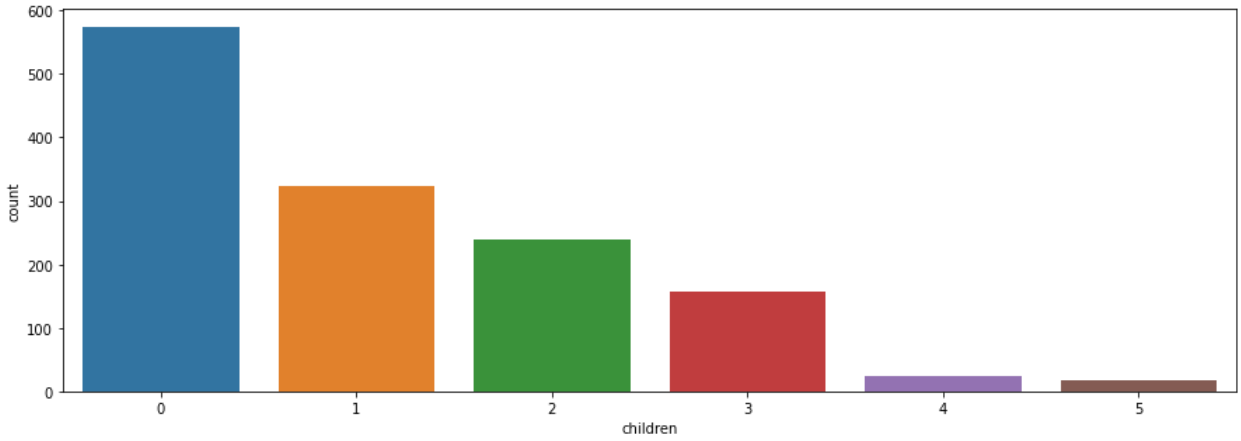
Observation:

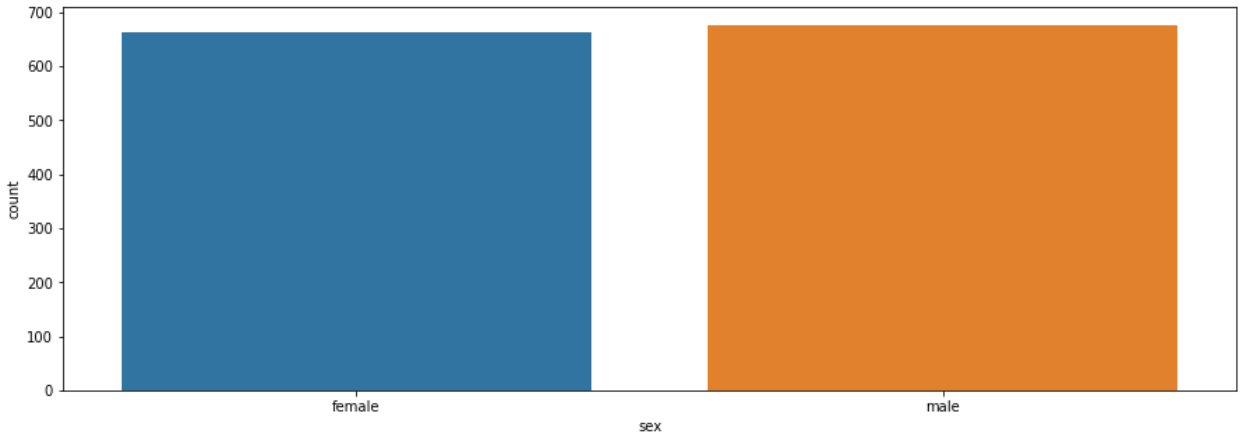
- "expenses" are slightly right skewed
- "age" is normally distributed wrt the problem statement

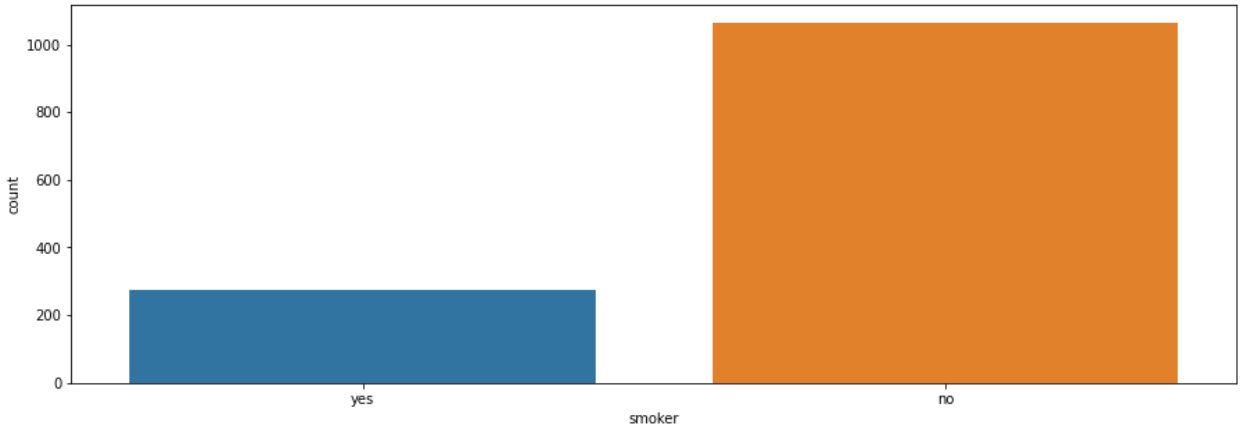
Countplots:

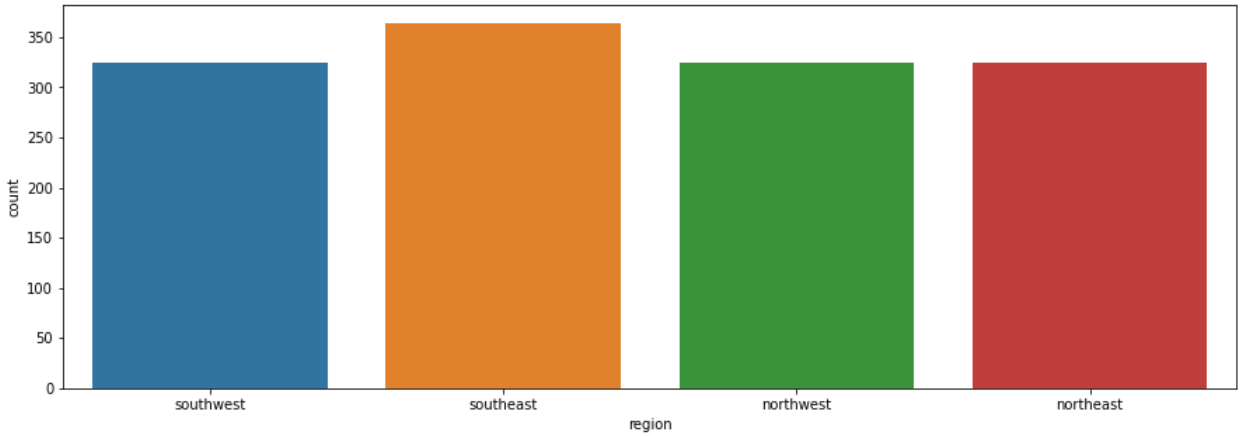
Countplots are useful for understanding the distribution of a categorical variable and identifying the most common categories in the data. They can be used to compare the counts of different categories, and to identify patterns and trends in the data.


count
age


count
children


count
female
male
sex



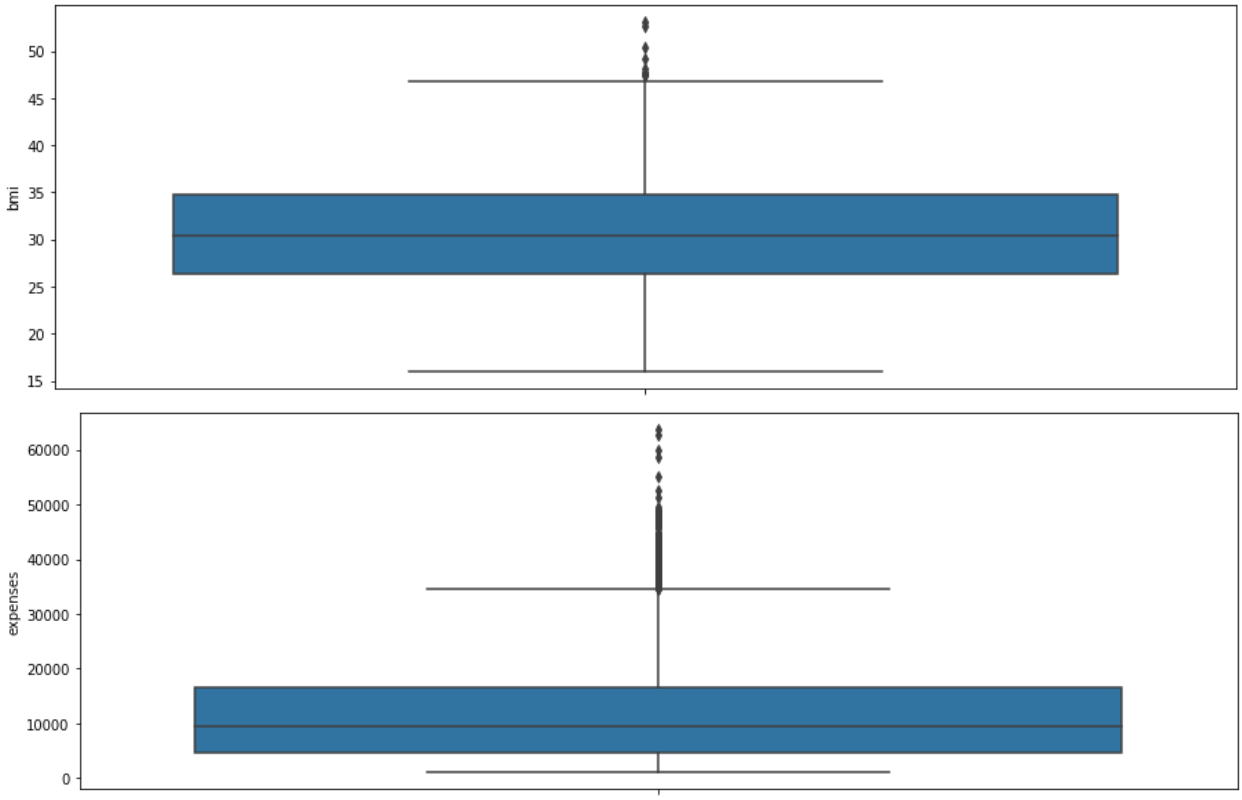


Observation:

- Count of the region is uniformly distributed
- Count of persons having no smoking habit is more
- Count of males and females is evenly distributed
- Most of the users have no children in the dataset
- Except the count of 18 and 19 years age, all the age are evenly distributed

Boxplot:

Boxplots are useful for visualizing the distribution of a dataset and identifying patterns and trends in the data. They can be used to compare the distribution of different datasets, and to identify the presence of outliers in the data.



Observation:

- There are outliers present in “bmi” but as per the problem statement, these are important outliers. Also, we are getting good training and testing accuracy, so not removing outliers.

5. Model Workflow

5.1 Model Selection

- After this the data was split into 2 sets X and y. X contains all the columns except the target column in our case (Count), y contains only the Target column.
- Using train test split we first split the dataset into X_train,X_test, y_train, y_test .
- The following libraries were imported to create Regression models.
    - sklearn.linear_model import LinearRegression, XgBoostRegressor
    - from sklearn.tree import DecisionTreeRegressor, ExtraTreeRegressor
    - from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
    - from lazypredict.Supervised import LazyRegressor

5.2 Model Accuracy Scores.

- **XgBoost Regressor has been chosen as the final algorithm to build the model**
    - ***MAE: 2531.39***
    - ***RMSE: 4579.76***
    - ***Test>R-Squared Accuracy: 0.857***
    - ***Test>Adjusted R-Squared Accuracy: 0.856***
    - ***Train> R-Squared Accuracy: 0.898***
    - ***Train>Adjusted R-Squared Accuracy: 0.898***

- The model was pickled using the Python pickle library and was ready for use into our Backend system.

6. Life cycle of a Machine learning project

A machine learning project typically goes through the following stages:

1. Problem definition: The first step is to define the problem that you want to solve using machine learning. This involves understanding the business context, determining the goals and objectives of the project, and identifying the data that you will need to train the model.

2. Data collection and preparation: The next step is to collect and prepare the data that you will use to train the model. This may involve scraping data from the web, collecting data from APIs, or using a pre-existing dataset. You will also need to clean and pre-process the data to get it into a suitable format for training.

3. Exploratory data analysis: Once you have collected and prepared the data, you will need to explore it to get a better understanding of its characteristics and any patterns that may exist. This will help you to identify any issues with the data and inform the development of the machine learning model.

4. Pre-processing:
Data pre-processing refers to the process of preparing data for use in training a machine learning model. This typically involves a number of steps, including:

1. Collecting data from various sources: This may include scraping data from websites, accessing databases, or collecting data from sensors or other devices.
2. Cleaning the data: This involves removing any invalid or missing data, correcting errors, and ensuring that the data is in a consistent format.
3. Normalizing or scaling the data: This involves transforming the data so that it is on the same scale, which can help improve the performance of some machine learning algorithms.
4. Splitting the data into training and test sets: This involves dividing the data into two sets, one for training the model and the other for evaluating its performance.
5. Extracting features: This involves selecting the relevant data and transforming it into a form that can be used by a machine learning algorithm.
6. Encoding categorical data: If the data includes categorical variables, they need to be encoded as numerical values in order to be used by most machine learning algorithms.

Data pre-processing is an important step in the machine learning process, as it helps ensure that the data is in a suitable form for training a model and can help improve the model's performance.

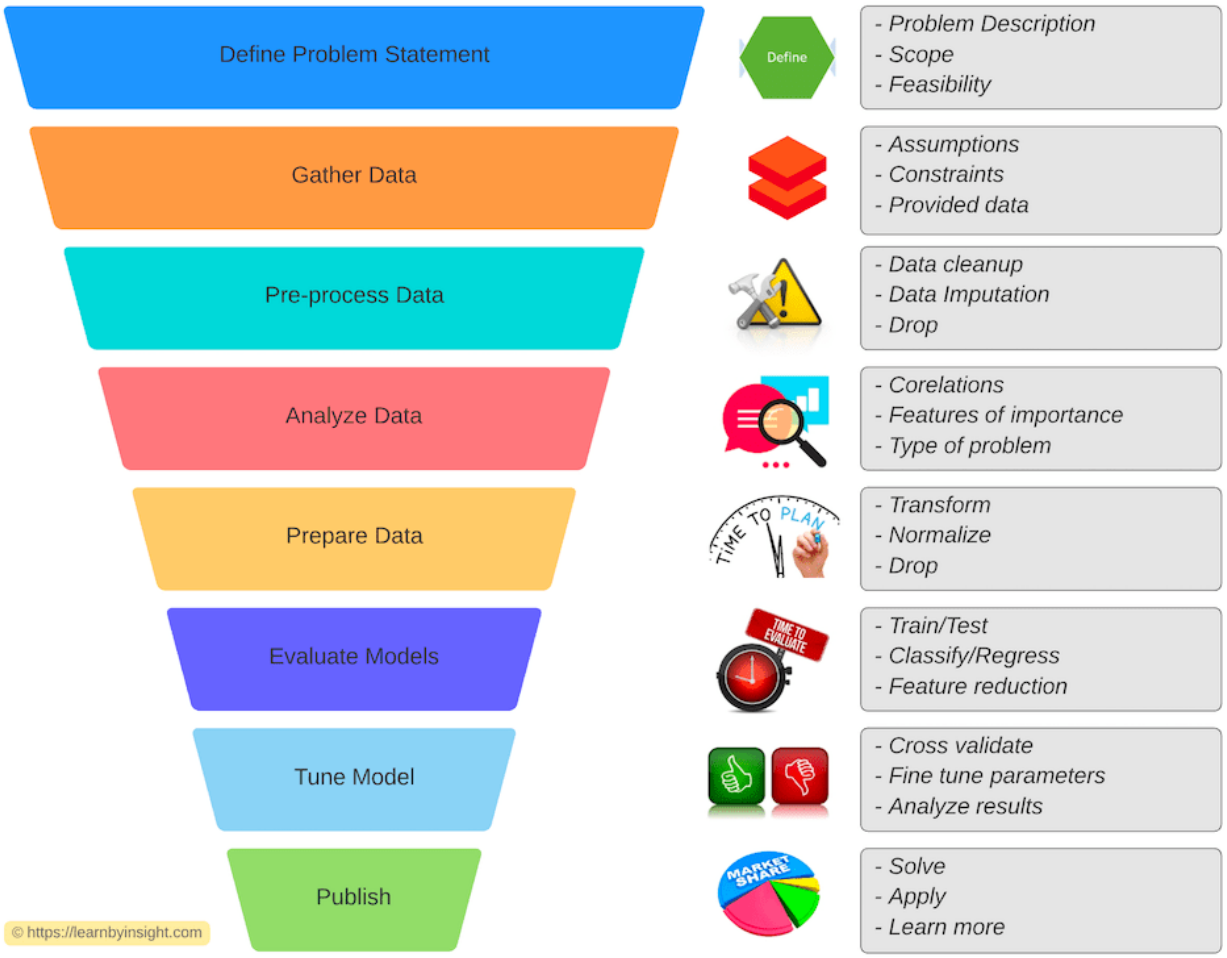
4. Model selection and training: The next step is to select an appropriate machine learning model and train it on the data. This will involve selecting a model type,

choosing the hyper parameters for the model, and training the model on the data using an optimization algorithm.

5. Model evaluation: Once the model has been trained, you will need to evaluate its performance to see how well it is able to make predictions. This will involve using a variety of evaluation metrics, such as accuracy, precision, and recall, to assess the model's performance.

6. Model deployment: If the model performs well and meets the project goals, it can be deployed to a production environment where it can be used to make predictions in real-time. This may involve integrating the model into a web or mobile application, or using it to automate a business process.

7. Model maintenance: Even after a model has been deployed, it will still need to be maintained and updated over time. This may involve retraining the model on new data, fine-tuning the hyper parameters, or implementing additional features to improve its performance.



8. Deployment

This model is deployed on Streamlit Web app using Github and Docker:

- Install Docker
- Add the runner in the GitHub
- Add all the secret keys in the GitHub
- In the GitHub actions, run the continuous delivery and deployment workflow once after starting the runner in the ec2 instance
- Start the instance & locate the Docker run.sh file for to initiate the "Runner" to pick the job.
- A web app has been created and deployed using Streamlit

Conclusion:

We have successfully built end-to-end ML projects using machine learning that can help predict the medical expenses of the users based on various conditions. This type of system can help users to get a better understanding of their medical expenses and based on it they can buy their insurance plan. Along with end to end pojects, a Streamlit webapp has been created as well to get the result based on certain inputs like age, sex, bmi etc.