Detail Project Report

Insurance Premium Prediction

Revision Number – 1.0

Last Date of Revision: 14 – 05 -2023

Santhana Lakshmi

Document Version Control

Date	Version	Description	Author
14– 05 - 2023	1.0	First Draft	Santhana Lakshmi

Contents

Document Version Control 2

Abstract 4

1. Introduction 5

1.1 Why this DPR Document ? 5

2. General Description 5

2.1 Problem Perspective 5

2.2 Problem Statement 5

2.3 Proposed Solution 5

2.4 Further Improvements 6

3. Technical Requirements 6

3.1 Tools Used 6

4. Data Requirements 6

4.1 Data Collection 6

4.2 Data Description 7

4.3 Exploratory Data Analysis 7

5. Data Preprocessing 8

6. Design Flow 8

7. Logging 9

8. Data Validation 9

9. Rendering the Results 9

10. Deployment 9

11. Conclusion 10

Abstract:

The Insurance Premium Prediction project aims to develop a machine learning model that can predict the premium amount for an insurance policy based on various factors such as age, gender, occupation, health status, etc. This project will help insurance companies to accurately estimate the premium amount for each policy, which will ensure fair pricing and better customer satisfaction.

This detailed project report will provide an overview of the project, its scope, objectives, methodology, and implementation details. The report will also discuss the challenges faced during the project, the solutions implemented, and the results obtained.

1.Introduction

The Insurance Premium Prediction project is a machine learning-based solution to predict the premium amount for an insurance policy. The project aims to help insurance companies to accurately estimate the premium amount for each policy, which will ensure fair pricing and better customer satisfaction. The project will be developed using a supervised learning algorithm, which will be trained on a dataset of historical insurance policy data.

1.1 Why this DPR Document?

The purpose of this detailed project report (DPR) is to provide a comprehensive overview of the Insurance Premium Prediction project. This report will cover the entire project, from the problem statement to the proposed solution, and will also highlight the technical details, data analysis, and key performance indicators (KPIs). The document will serve as a guide for stakeholders to understand the project's progress and goals.

2.General Description

2.1 Problem Perspective

Insurance premium prediction is a critical task for insurance companies to ensure they have adequate reserves to cover potential claims. The traditional methods used for premium calculation are often inaccurate and lack transparency. Hence, there is a need to develop a more reliable and accurate method for insurance premium prediction.

2.2 Problem Statement

The aim of this project is to develop an accurate and reliable insurance premium prediction model that can predict the insurance premium for a policyholder based on various factors such as age, gender, location, vehicle type, and driving history.

2.3 Proposed Solution

The proposed solution for this project is to use machine learning algorithms to develop an insurance premium prediction model. The model will be trained on historical insurance data to learn the relationship between various factors and insurance premiums. Once the model is trained, it can be used to predict the insurance premium for new policyholders.

2.4 Further Improvements

There are several opportunities for improvement in this project, including:

Enhancing the accuracy of the model by incorporating more data sources.

Implementing the model in a real-time environment to facilitate on-demand premium calculation.

Developing a user interface to enable customers to receive quotes online and reduce the turnaround time for premium calculation.

3.Technical Specification

This project will use Python as the primary programming language, along with several libraries such as Pandas, NumPy, Matplotlib, and Scikit-learn. The project will also use a Jupyter notebook for data analysis and model training. The final model will be deployed using Flask, a web application framework.

4.Data Requirements:

Historical data on insurance policies including their premiums

Data on policyholders including age, gender, marital status, location, and driving record

Data on the type of vehicle insured, including make, model, and year

Data on the coverage limits and deductibles selected by the policyholder

Data on claims history of the policyholder

Data on the location of the policyholder including state and zip code

4.1Data Collection

The data has been collected from Kaggle:

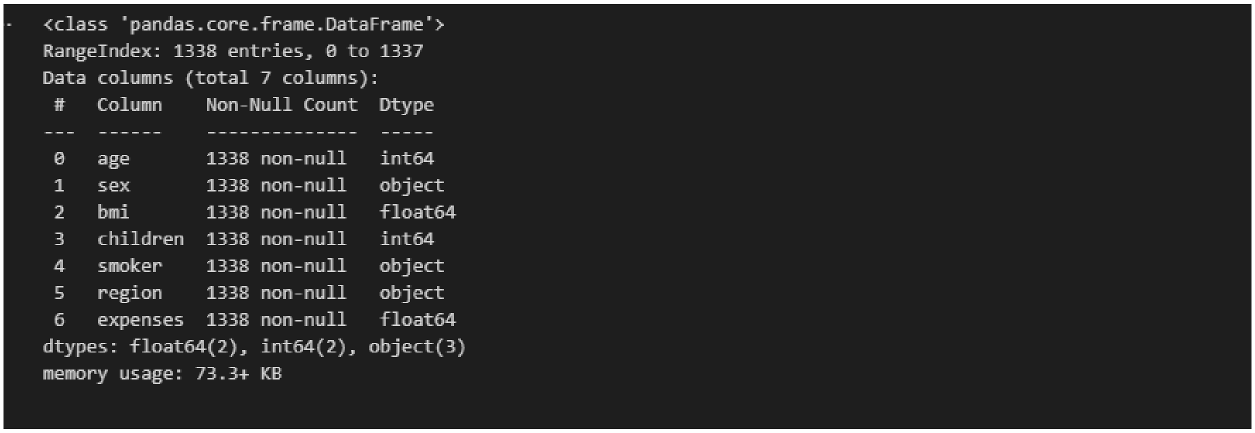
URL:https://www.kaggle.com/noordeen/insurance-premium-prediction

4.2 Data Description:

The data should be in a structured format and should include all the required fields. The data should be clean and free from any errors or inconsistencies.

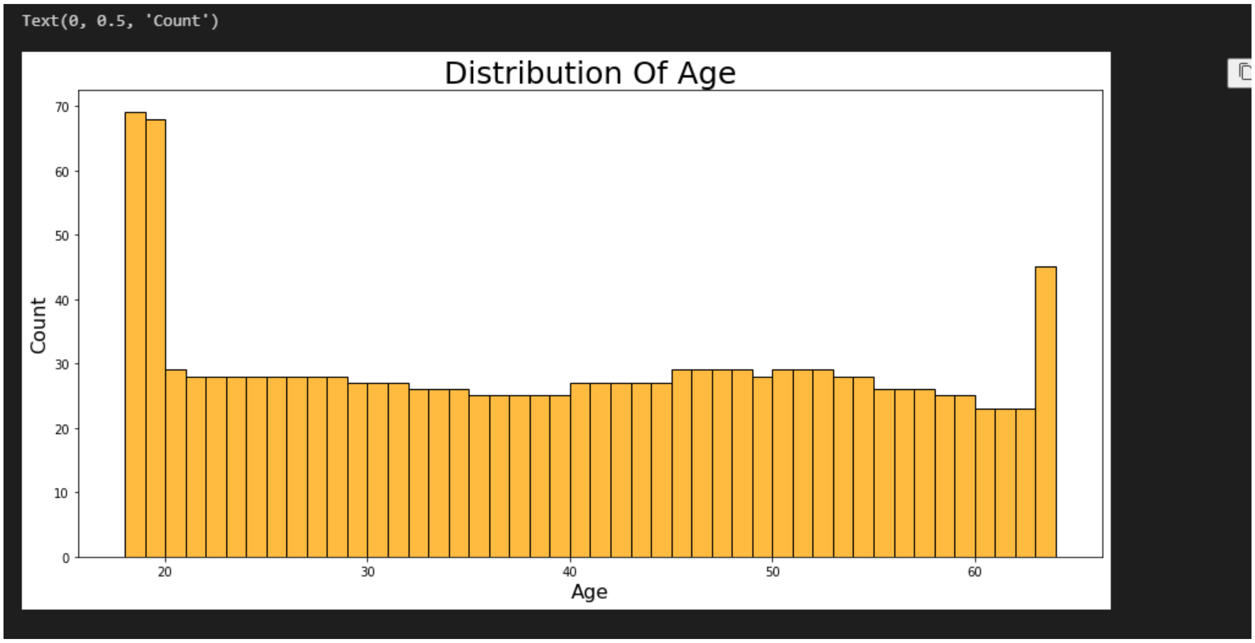
4.3 Exploratory Data Analysis:

Data cleaning: Check for missing values, outliers, and data errors that could impact the modeling process. Remove or impute missing values, remove outliers, and correct any data errors.

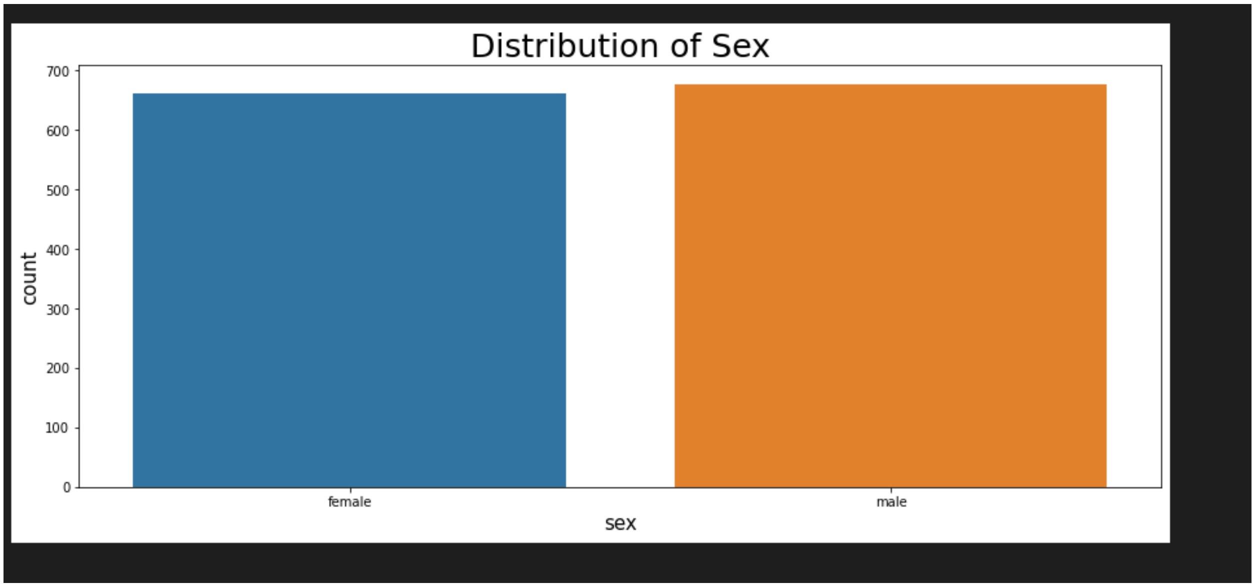


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   expenses  1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

Data visualization: Visualize the data to get insights and identify potential issues. Use histograms, scatter plots, box plots, and other visualizations to explore the relationships between features and the target variable.

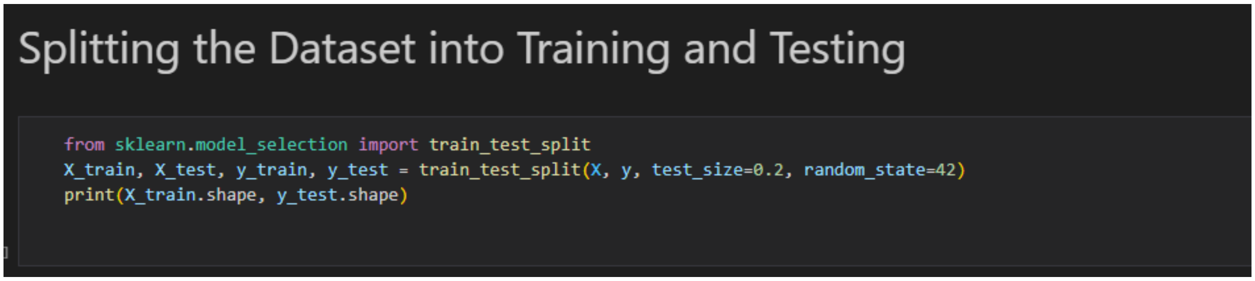


```
Text(0, 0.5, 'Count')
```



5.Data Preprocessing

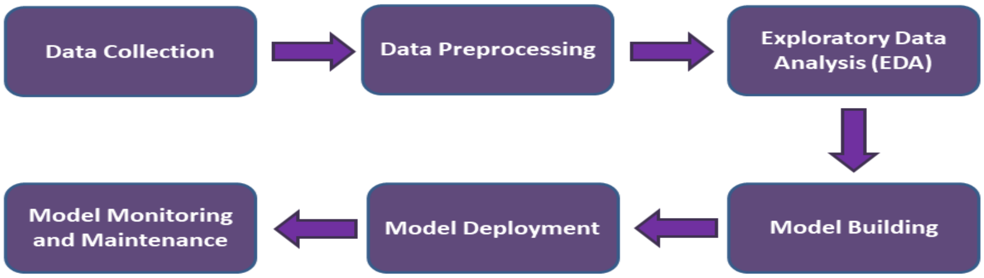
the dataset can be split into training and test sets for model building and evaluation. The data splitting can be performed using train test split.



Splitting the Dataset into Training and Testing

```
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=42)
print(X_train.shape, y_test.shape)
```

6.Design Flow

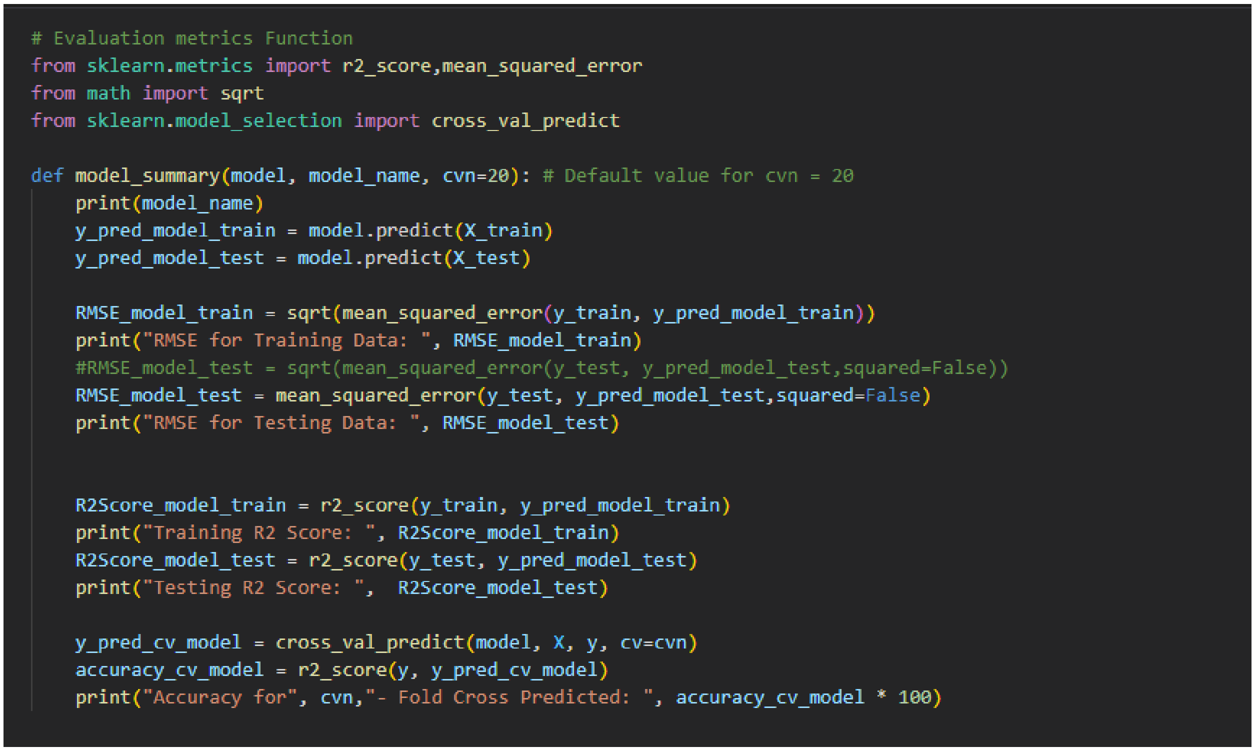

Data Collection
Data Preprocessing
Exploratory Data Analysis (EDA)
Model Building
Model Deployment
Model Monitoring and Maintenance

7. Logging

Logging will be used to capture and store relevant data for future analysis and improvement. The logging will capture key performance indicators (KPIs), such as the accuracy of the model, the time taken for the model to train, and the time taken for the model to make predictions.

8.Data Validation

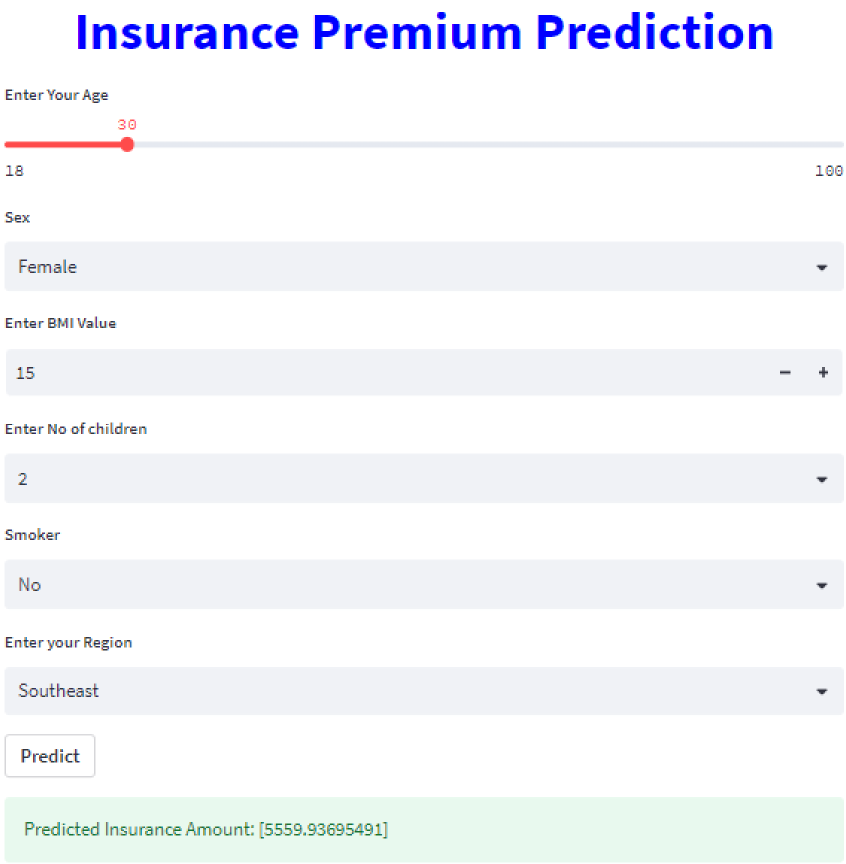
The training set is used to train the model, while the testing set is used to evaluate the model's accuracy. Ensure that the data is randomly split to avoid bias.



```
# Evaluation metrics Function
from sklearn.metrics import r2_score,mean_squared_error
from math import sqrt
from sklearn.model_selection import cross_val_predict

def model_summary(model, model_name, cvn=20): # Default value for cvn = 20
    print(model_name)
    y_pred_model_train = model.predict(X_train)
    y_pred_model_test = model.predict(X_test)

    RMSE_model_train = sqrt(mean_squared_error(y_train, y_pred_model_train))
    print("RMSE for Training Data: ", RMSE_model_train)
    #RMSE_model_test = sqrt(mean_squared_error(y_test, y_pred_model_test,squared=False))
    RMSE_model_test = mean_squared_error(y_test, y_pred_model_test,squared=False)
    print("RMSE for Testing Data: ", RMSE_model_test)


    R2Score_model_train = r2_score(y_train, y_pred_model_train)
    print("Training R2 Score: ", R2Score_model_train)
    R2Score_model_test = r2_score(y_test, y_pred_model_test)
    print("Testing R2 Score: ",  R2Score_model_test)

    y_pred_cv_model = cross_val_predict(model, X, y, cv=cvn)
    accuracy_cv_model = r2_score(y, y_pred_cv_model)
    print("Accuracy for", cvn,"- Fold Cross Predicted: ", accuracy_cv_model * 100)
```

9.Deployment

The first step in deploying an Insurance Premium Prediction model is to prepare the model for deployment. This includes ensuring that the model is properly trained and validated, and that it is compatible with the deployment environment. Once the model is prepared, the next step is to create an API (Application Programming Interface) that can be used to interface with the model. Once the API and front-end are ready, the next step is to deploy the API on Streamlit.



Insurance Premium Prediction

Enter Your Age

30

18 100

Sex

Female

Enter BMI Value

15

Enter No of children

2

Smoker

No

Enter your Region

Southeast

Predicted Insurance Amount: [5559.93695491]

11.Conclusion

In conclusion, the Insurance Premium Prediction project has the potential to revolutionize the insurance industry and improve the customer experience by providing accurate and personalized insurance quotes.