Insurance Premium Prediction

Detailed Project Report

Introduction

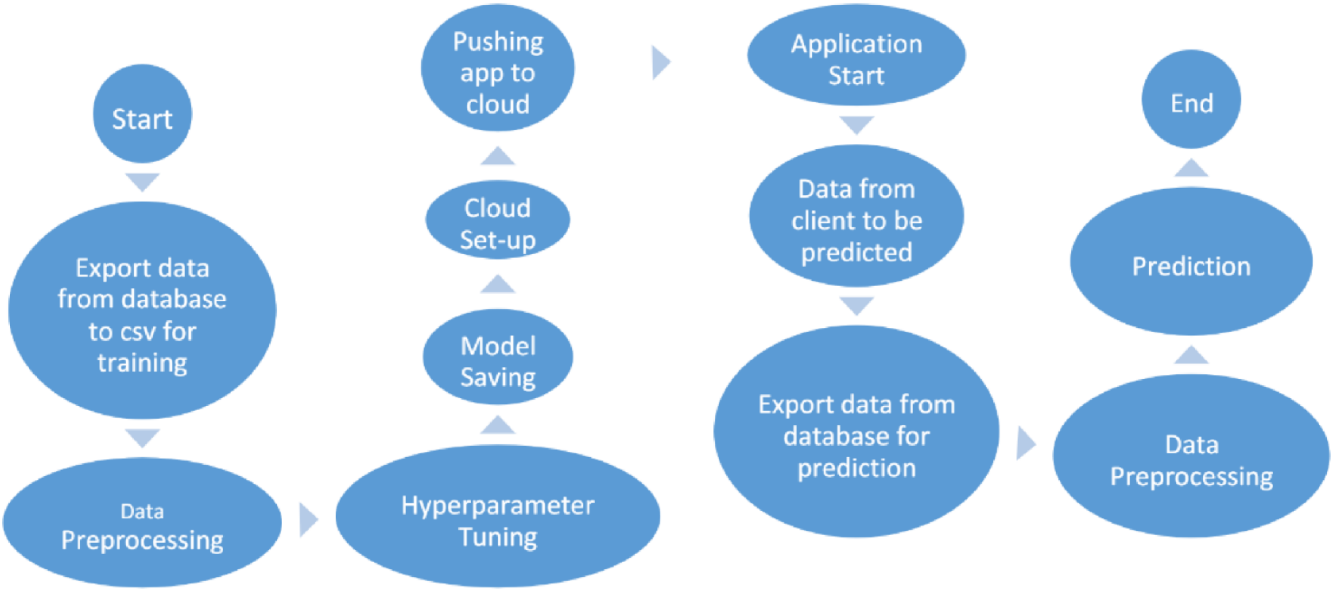
Insurance is a means of protection from financial loss in which, in exchange for a fee, a party agrees to compensate another party in the event of a certain loss, damage, or injury. It is a form of risk management, primarily used to hedge against the risk of a contingent or uncertain loss.

An entity which provides insurance is known as an insurer, insurance company, insurance carrier, or underwriter. A person or entity who buys insurance is known as a policyholder, while a person or entity covered under the policy is called an insured. The insurance transaction involves the policyholder assuming a guaranteed, known, and relatively small loss in the form of a payment to the insurer (a premium) in exchange for the insurer's promise to compensate the insured in the event of a covered loss. The loss may or may not be financial, but it must be reducible to financial terms. Furthermore, it usually involves something in which the insured has an insurable interest established by ownership, possession, or pre-existing relationship.

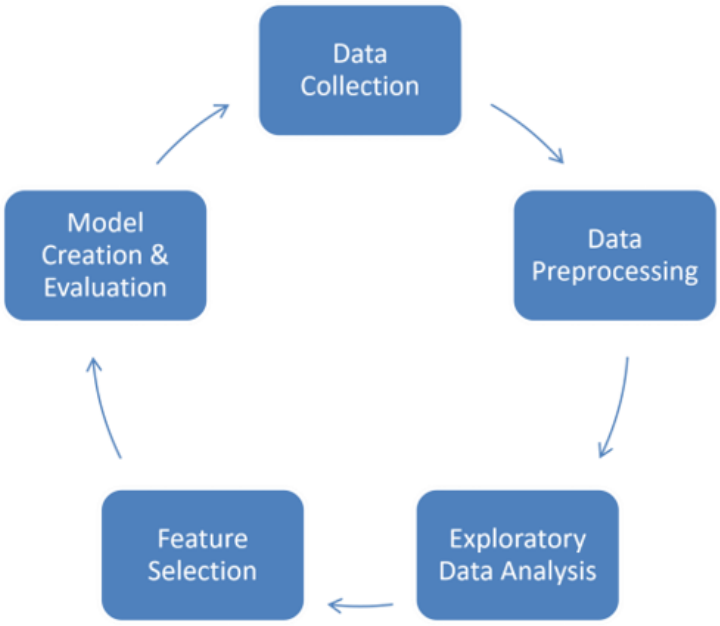
Objective

The main goal of this project is to predict the insurance premium and tell the client for their health benefits that uh can cover this much by taking insurance premium and this is good for their health and wealth.

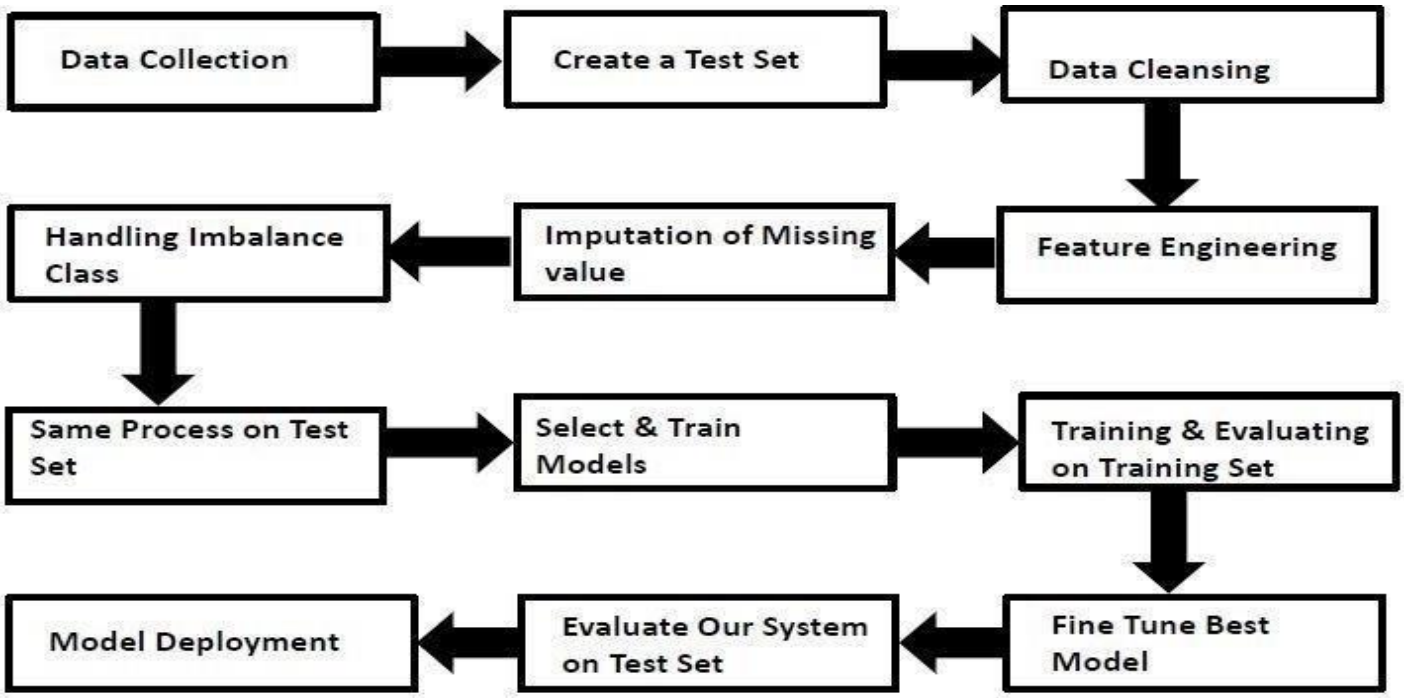
Architecture



Data Analysis Steps



Model Training and Validation Work



Flow

Data Collection

- Insurance Premium Prediction data set from Kaggle
- For Data Set: https://www.kaggle.com/datasets/noordeen/insurance-premium-prediction

Data Pre-Processing

- Missing values handling by Simple imputation (Used KNN Imputer)
- Outliers' detection and removal by boxplot and percentile methods
- Categorical features handling by ordinal encoding and label encoding
- Feature scaling done by Standard Scalar method
- Imbalanced dataset handled by SMOTE -Over sampling
- Drop unnecessary columns

Workflow

Model Creation and Evaluation

- Various classification algorithms like Random Forest, XG Boost, KNN, etc. tested.
- Random Forest, XGBoost and KNN all were given better results. Random Forest was chosen for the final model training and testing.
- Hyper parameter tuning was performed.
- Model performance evaluated based on accuracy, confusion matrix, classification report.

Random Forest Regression Model

Introduction

It is a decision-tree-based ensemble Machine Learning algorithm which combines the output of multiple decision trees to reach a single result.

The Random Forest Regressor is a supervised learning algorithm which we can use for regression and classification problems. It is among the most popular machine learning algorithms comes under bagging ensemble technique.

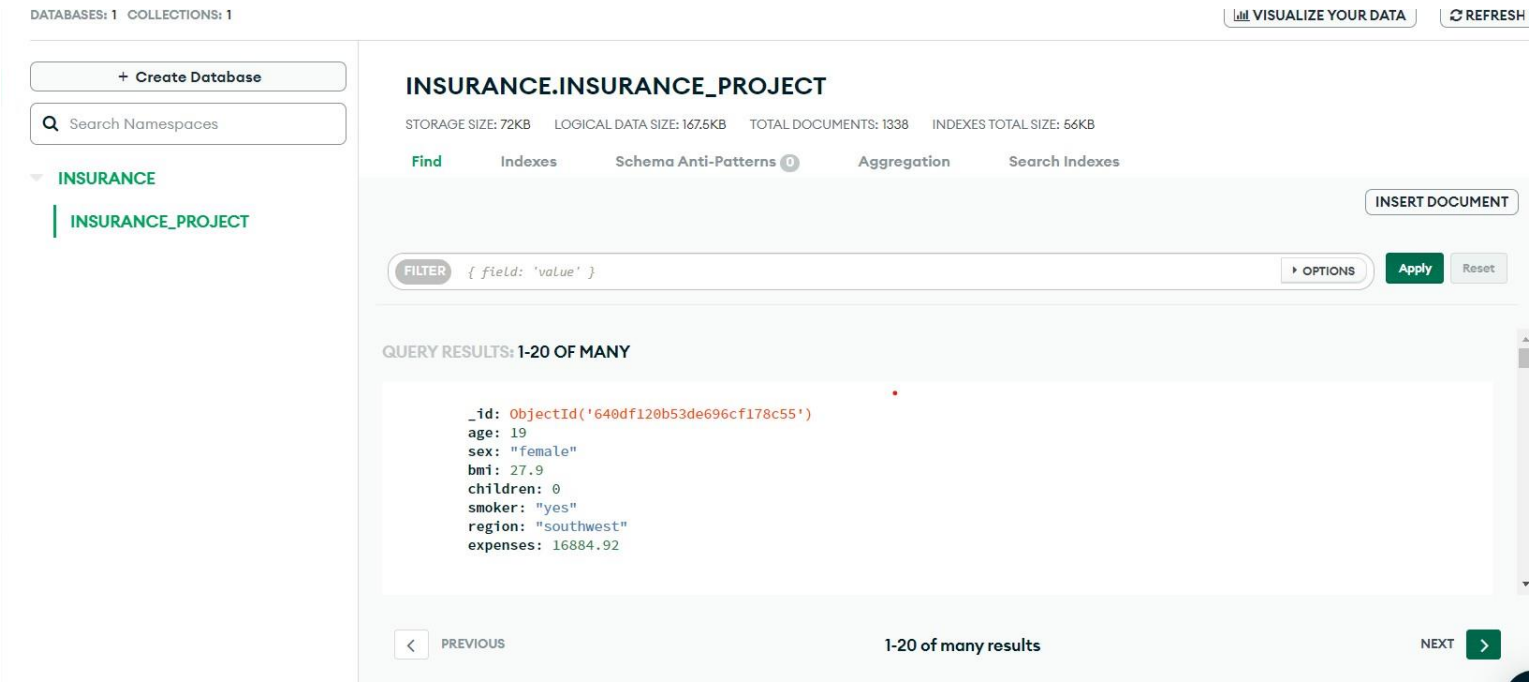
Random Forest Regressor being ensemble algorithm tends to give more accurate result. This is because it works on the principle i.e., it creates the random forest by combining N decision tree, and make predictions for each tree created in the first phase. Even if one or few decision tree are prone to noise, overall results would tend to be correct.

Reason to use Random Forest Regressor model:

- It takes less training time as compared to other algorithms.
- It gives better model performance.

Database Connection & Deployment

- MongoDB Database is used for this project.



Model Deployment

- The final model is deployed on AWS using Flask framework.

FREQUENTLY ASKED QUESTIONS

Q1) What is the source of data?

The data for training is obtained from famous Kaggle.

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

Q2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q3) What's the complete flow you followed in this Project?

Refer slide 7th, 8th and 9th for better understanding.

Q4) After the File validation what you do with incompatible file or files which didn't pass the validation?

Files like these are moved to the Archive Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q 6) What techniques were you using for data pre-processing?

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output variables
- Checking and changing Distribution of continuous values
- Removing outliers
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.

Q 7) How training was done or what models were used?

- First Data validation done on raw data and then good data insertion happen in DB.
- Then Data preprocessing done on final CSV file received from DB.
- Various model such as Decision Tree, Random Forest and XGBoost models are trained and based on performance, model is saved.

Q 8) How Prediction was done?

- The client fills the required inputs which is visible on the homepage of API.
- After filling all the required inputs, prediction was made and client sees the desired result.

Q 9) What are the different stages of deployment?

- After model training and finalizing all models. We created required files for deployment.
- Finally deployed our model over a cloud platform named as AWS.

Q 10) How is the User Interface present for this project?

- For this project we have made only one type of UI.
- It is for one user input prediction.
- It very user friendly and easy to use.

Thank You!