Architecture Design

Insurance Premium Prediction

Document Version Control

Date Issued	Version	Description	Author
26.8.2024	V1.0	Initial LLD- V1.0	Sarthak Bansal
26.8.2024	V1.0	Initial LLD- V1.0	Jayant Bisht
26.8.2024	V1.0	Initial LLD- V1.0	Hardik Mudgal
26.8.2024	V1.0	Initial LLD- V1.0	Kartik
26.8.2024	V1.0	Initial LLD- V1.0	Mayank Singla

Index

Abstract

This design introduces an advanced architecture for predicting insurance premiums using machine learning. It integrates data ingestion, preprocessing, feature extraction, and predictive modeling to enhance accuracy. The process starts with consolidating and cleaning diverse datasets, followed by extracting key features. Machine learning algorithms, including ensemble methods and deep learning, then generate precise premium forecasts. A feedback loop ensures continuous improvement by adapting to new trends. This architecture aims to improve prediction accuracy and provide valuable insights for insurers.

1. Introduction

1.1 What is Architecture Design?

The goal of Architecture Design is to give the internal design of the actual program code for the **Insurance Premium Prediction**. Architecture Design describes the class diagrams with the methods and relation between classes and program specification. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Architecture Design is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. And the complete workflow.

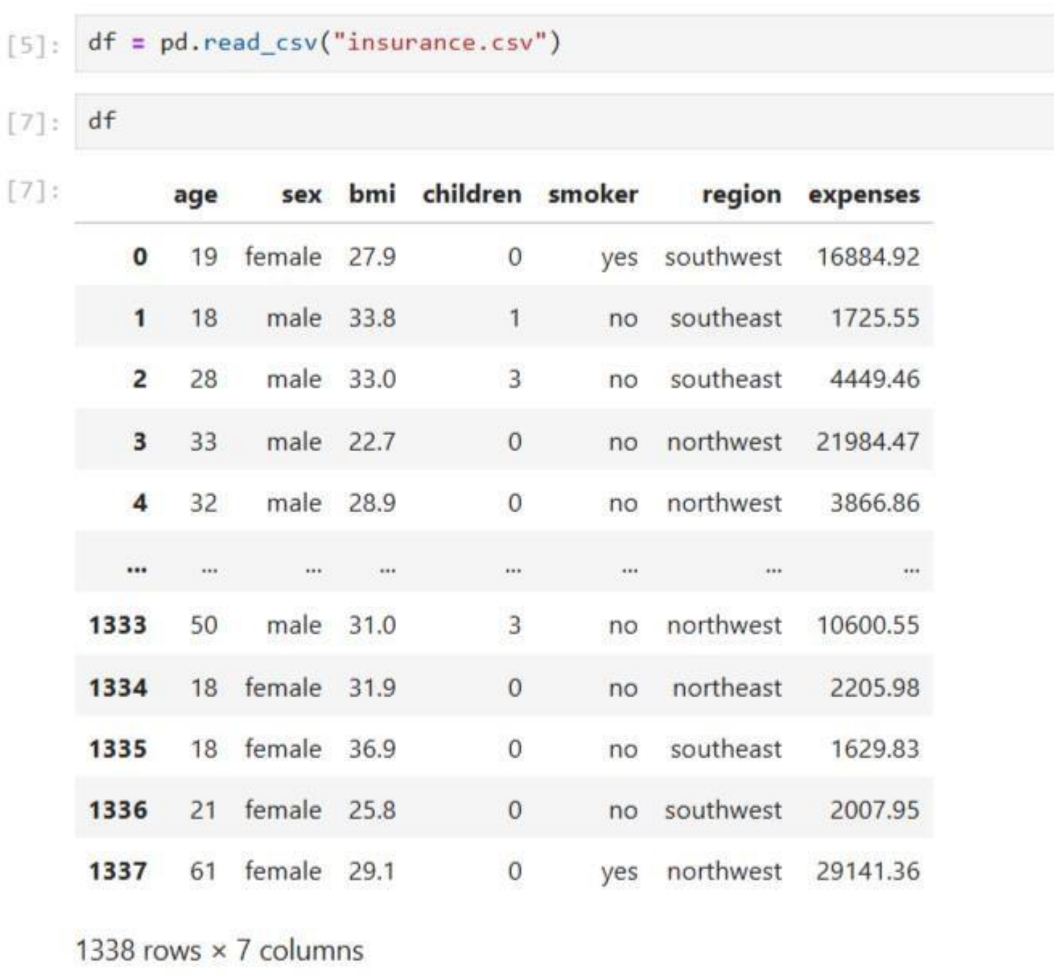
1.3 Constraints

We only predict the expected estimating cost of expenses customers based on some personal health information.

2. Technical Specification

2.1 Dataset

The dataset containing verified historical data, consisting of the aforementioned information and the actual medical expenses incurred by over 1300 customers. The objective is to find a way to estimate the value in the "expenses" column using the values in the other columns like their age, sex, BMI, no. of children, smoking habits and region. Using all the observations it is inferred what role certain properties of user and how they affect their expenses. The dataset looks like as follow:



```
[5]: df = pd.read_csv("insurance.csv")
```

```
[7]: df
```

[7]:

	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
...	...	...	...	...	...	...	...
1333	50	male	31.0	3	no	northwest	10600.55
1334	18	female	31.9	0	no	northeast	2205.98
1335	18	female	36.9	0	no	southeast	1629.83
1336	21	female	25.8	0	no	southwest	2007.95
1337	61	female	29.1	0	yes	northwest	29141.36

1338 rows × 7 columns

2.2 Logging

We should be able to log every activity done by the user

- The system identifies at which step logging require.
- The system should be able to log each and every system flow.
- The system should be not be hung even after using so much logging. Logging just because we can easily debug issuing so logging is mandatory to do.

2.3 Deployment

For the hosting of the project, we are using

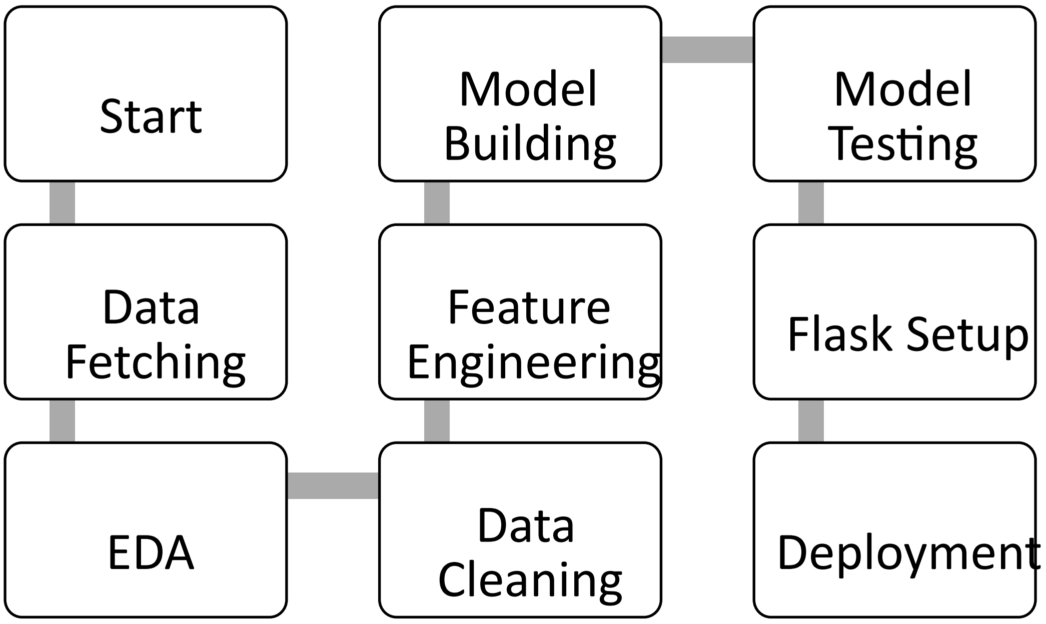
3. Technology Stack

Front End	HTML/CSS
Backend	Python/ Flask
Deployment	AWS

4. Proposed Solution

We will use performed EDA to find the important relation between different attributes and will use a machine-learning algorithm to estimate the cost of expenses. The client will be filled the required feature as input and will get results through the web application. The system will get features and it will be passed into the backend where the features will be validated and preprocessed and then it will be passed to a hyperparameter tuned machine learning model to predict the final outcome.

5. Architecture



5.1 Data Gathering

Data source: Insurance Premium Prediction (kaggle.com) Dataset is stored in .csv format.

5.2 Raw Data Validation

After data is loaded, various types of validation are required before we proceed further with any operation. Validations like checking for zero standard deviation for all the columns, checking for complete missing values in any columns, etc. These are required because the attributes which contain these are of no use. It will not play role in contributing to the estimating cost of the premium.

5.3 Exploratory Data Analysis

Visualized the relationship between the dependent and independent features. Also checked relationship between independent features to get more insights about the data.

5.4 Feature Engineering

After pre-processing standard scalar is performed to scale down all the numeric features. Even one hot encoding is also performed to convert the categorical features into numerical features. For this process, pipeline is created to scale numerical features and encoding the categorical features.

5.5 Model Building

After doing all kinds of pre-processing operations mention above and performing scaling and encoding, the data set is passed through a pipeline to all the models, Linear Regression, Decision tree, Random Forest, Gradient boost, KNN and XGBoost regressor using EvalML

5.6 Model Saving

Model is saved using pickle library in pickle` format.

5.7 Flask Setup For Web Application

After saving the model, the API building process started using Flask. Web application creation was created in Flask for testing purpose. Whatever user will enter the data and then that data will be extracted by the model to estimate the premium of insurance, this is performed in this stage.

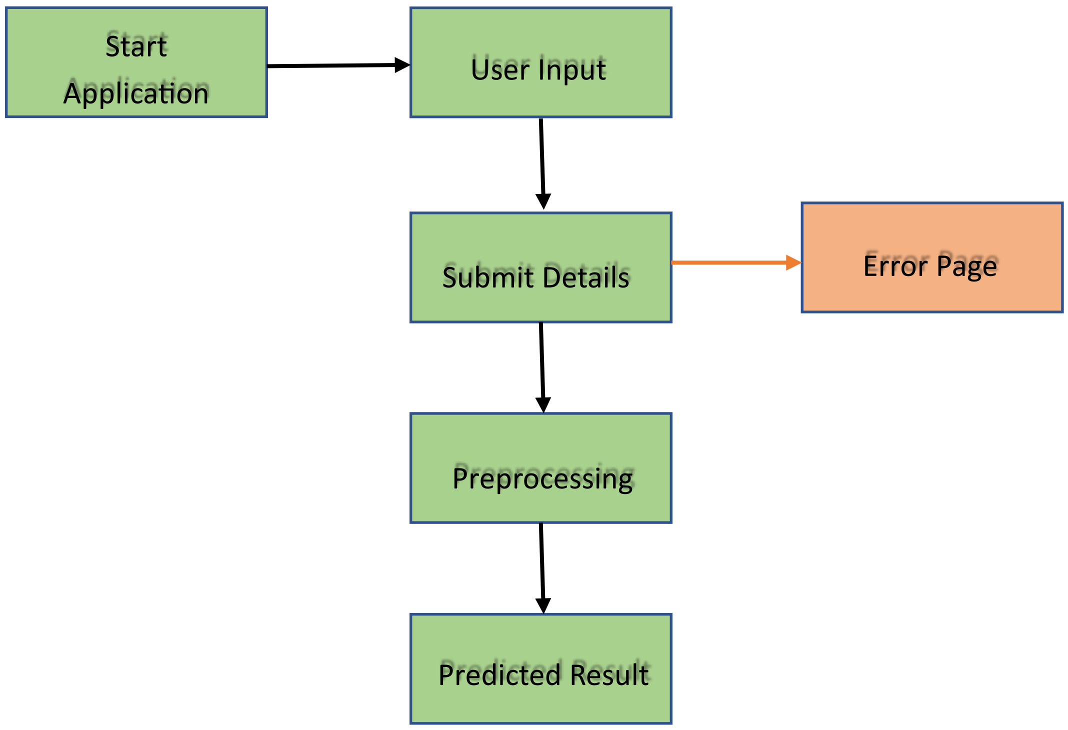
5.8

The whole project directory will be pushed into the GitHub repository.

5.9 Deployment

The project was deployed from GitHub into the Heroku platform.

6. User Input / Output Workflow


Start Application
User Input
Submit Details
Error Page
Preprocessing
Predicted Result