Architecture

Architecture

Insurance Premium Prediction

Revision Number – 1.0

Last Date of Revision – 18-10-2023

**ARUNKUMAR V**

1

Architecture



**Liabaries Requirment**

Streamlit

pandas

numpy

scikit-learn

seaborn

ARCHITECTURE

**Contents**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**Abstract**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.gjdgxs) | | | | | | | | | | | | | | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.gjdgxs) | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [**Introduction**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.30j0zll) | | |  | | | | | | | | | | | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.30j0zll) | | | |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [**Why this Architecture Design documentation?**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1fob9te) | | | | | | | | | | | | | |  | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1fob9te) |  |
| [**1 Architecture**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3znysh7) | | | |  | | | | | | | | | | | [4](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3znysh7) | | | |
|  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [**2 Architecture design**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3dy6vkm) | | | | | | | |  | | | | | | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3dy6vkm) | | |  |
|  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [**2.1 Data gathering from main source**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1t3h5sf) | | | | | | | | | | | | | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1t3h5sf) | | |  |
|  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [**2.2 Data description**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.4d34og8) | | | | | | | | | | | | |  | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.4d34og8) | | |  |
|  |  | | | |  |  |  |  |  |  |  |  |  | |  |  |  |  |
|  | [**2.3 Upload data into Cassandra**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.2s8eyo1) | | | | | | | | | | | |  | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.2s8eyo1) | | | |
|  |  | | | |  |  |  |  | |  |  |  |  | |  |  |  |  |
|  | [**2.4 Export data from database**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.17dp8vu) | | | | | | | | | | |  |  | | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.17dp8vu) | | |  |
|  |  | | | |  |  |  |  | |  |  |  | | |  |  |  |  |
|  | [**2.5 Data pre-processing**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3rdcrjn) | | | | | | | | | | |  | | |  | [5](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.3rdcrjn) | |  |
|  |  | | | |  |  |  |  | |  |  | | | |  | |  |  |
|  | [**2.6 Modelling**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.26in1rg) | | | |  | | | | |  | | | | |  | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.26in1rg) | |
|  |  | | | |  |  |  |  | | |  | | | |  | |  |  |
|  | [**2.7 UI integration**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.lnxbz9) | | | | |  | | | | | | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.lnxbz9) | | |  |
|  |  | | | | |  |  |  | | |  | | | |  | |  |  |
|  | [**2.8 Data from user**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.35nkun2) | | | | | | | | | | | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.35nkun2) | | |  |
|  |  | | | | |  |  |  | | |  | | | |  | |  |  |
|  | [**2.9 Data validation**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1ksv4uv) | | | | | | |  | | | | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.1ksv4uv) | | |  |
|  |  | | | | |  | |  | | |  | | | |  | |  |  |
|  | [**2.10 Rendering the results**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.44sinio) | | | | | | | | | |  | | | | [6](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.44sinio) | | |  |
|  | [**2.11 Deployment**](https://docs.google.com/document/d/14Wck6wDTC8bscpCxNjrSRu01-UC_tJ5Y/edit#heading=h.2jxsxqh) | | | | | | | | | |  | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

3

ARCHITECTURE

Abstract

Predicting adult income is a complex task that involves analyzing various factors related to an individual's demographics, education, occupation, and other relevant variables. While I can provide you with some general information, please note that predicting income accurately for specific individuals can be challenging due to the wide range of factors involved and the inherent uncertainty of future outcomes.

Here are some common details that can influence adult income prediction:

1. Age: Higher age often correlate with higher Premium. Factors such as the highest level of severity of Disease,

2. Sex: The type of job an individual holds has a significant impact on income.

3. BMI: Generally, individuals with more BMI tend to have higher diseases.

4. Demographics: Certain demographic factors can influence Premium prediction. These may include age, gender, marital status, and geographical location. However, it's important to note that these factors should not be used to discriminate or make unfair predictions.

5. Smoking:Health levels can vary significantly based on the Habit of smoking

6. Children Size: The industry in which a person have children and the size of the family they are employed by can impact Premium

7. Additional Factors: Other variables that can affect Premium prediction include certifications, professional affiliations, networking, job performance, and individual characteristics such as motivation, ambition, and personal circumstances.

It's important to note that Premium prediction models are often built using machine learning techniques and statistical analysis. These models are trained on historical data to identify patterns and relationships between various predictors and income levels. However, the accuracy of predictions can vary based on the quality and relevance of the data used for training.

It's also worth mentioning that income prediction models should be used responsibly and ethically, as they have the potential to perpetuate biases or unfair practices. Care must be taken to ensure that predictions are not used for discriminatory purposes and that individuals are treated fairly based on their qualifications and abilities.

Introduction

Why this Architecture Design documentation?

The main objective of the Architecture design documentation is to provide the internal logic understanding of the flight fare prediction code. The Architecture design documentation is designed in such a way that the programmer can directly code after reading each module description in the documentation.

1 Architecture



4

ARCHITECTURE

2 Architecture design

Designing an architecture involves creating a blueprint or plan that outlines the structure, components, and interactions of a system. Here are some key considerations when designing an architecture:

1. Identify Requirements: Understand the functional and non-functional requirements of the system. This includes determining the goals, features, scalability, performance, security, and other aspects that the architecture should fulfill.

2. Choose Architectural Style: Select an appropriate architectural style or pattern that best suits the system requirements. Common styles include client-server, layered, microservices, event-driven, and service-oriented architectures.

3. Define Components: Identify the major components or modules that will make up the system. Components can be logical divisions such as user interface, business logic, data storage, external interfaces, and integrations.

4. Component Interaction: Determine how the components will interact with each other. Define the communication protocols, data flow, and dependencies between components. Consider whether synchronous or asynchronous communication is needed.

5. Scalability and Performance: Design the architecture to handle the expected workload and scale efficiently. Consider factors such as load balancing, caching, horizontal or vertical scaling, and optimizing performance bottlenecks.

6. Security: Incorporate security measures to protect the system and data. This includes authentication, authorization, encryption, input validation, and other security best practices relevant to the system.

7. Data Management: Plan how data will be stored, accessed, and managed. Consider the type of database (relational, NoSQL, etc.), data synchronization, data consistency, and backup strategies.

8. Deployment and Infrastructure: Determine the deployment architecture, including the hardware and software infrastructure required to host the system. Consider factors such as cloud platforms, server configurations, network architecture, and monitoring.

9. Error Handling and Resilience: Define strategies for handling errors and failures. This may involve implementing retries, redundancy, fault tolerance, and graceful degradation to ensure the system remains operational in the face of issues.

10. Maintainability and Extensibility: Design the architecture to be easily maintainable and extensible. Use modular and decoupled components, follow coding best practices, and consider future changes and integrations.

11. Documentation and Communication: Document the architecture design, including diagrams, descriptions, and any relevant documentation that helps communicate the design to stakeholders, developers, and other team members.

2.1 Data gathering from main source

The data for the current project is being gathered from Kaggle dataset, the link to the data is:

<https://www.kaggle.com/somay/flight-fare-prediction-mh>

2.2 Data description

Predicting adult income is a complex task that involves analyzing various factors related to an individual's demographics, education, occupation, and other relevant variables. While I can provide you with some general information, please note that predicting income accurately for specific individuals can be challenging due to the wide range of factors involved and the inherent uncertainty of future outcomes.

2.3 Upload data into MongoDB

* Connection is made with the database.
* Created a database with name flightfare.
* Data Uploaded with Import option in the MongoDB

2.4 Export data from database

In the above created api, the download url is also being created, which downloads the data into a csv file format.

2.5 Data pre-processing

Steps performed in pre-processing are:

* First the data types are being checked and found only the price column is of type integer.
* Checked for null values as there are few null values, those rows are dropped.
* Converted all the required column into the date time format.
* Performed one-hot encoding for the required columns.
* Scaling is performed for required data.
* And, the data is ready for passing to the machine learning algorithm.

2.6 Modelling

The pre-processed data is then visualized and all the required insights are being drawn. Although from the drawn insights, the data is randomly spread but still modelling is performed with different machine learning algorithms to make sure we cover all the possibilities. And finally, as expected random forest regression performed well and further hyperparameter tuning is done to increase the model’s accuracy.

2.7 UI integration

All the required files are then integrated to the app.py file and tested locally

2.8 Data from user

The data from the user is retrieved from the created web page.

2.9 Data validation

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent for the prediction.

5

2.10 Rendering the results

The data sent for the prediction is then rendered to the web page.

2.11 Deployment

The tested model is then deployed to Azure. So, users can access the project from any internet devices.

6

**ScreenShot of the App Interface which I will deploy**



