Architecture Design

Adult Census Income Prediction

Written By	SUDHANSU GOUDA
Version	1.0
Date	25/02/2023

Document Control

- Change Record:

Version	Date	Author	Comments

- **Approval Status:**

Version	Review Date	Reviewed By	Approved By	Comments

Index

Abstract

This data was extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics). A set of reasonably clean records was extracted. Certain attributes of each product and store have been defined. The aim is to build a predictive model and find out whether the income of the person is greater than or less than $50,000. Using this model we will get insights as well as we get to determine the income group.

1. Introduction

1.1 What is Architecture Design?

The goal of Architecture Design (AD) or a low-level design document is to give the internal design of the actual program code for the '**Adult Census Income Prediction** '. AD 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 (AD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, 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 income category based on occupation, workclass, education, gender, age, work hours, marital status about the candidate.

2. Technical Specification

2.1 Dataset

In this dataset there is data about people who work at different sector with different gender type and marital status which gives us a variety of combinations of data. Using all the observations it is inferred what role certain properties of an candidate play and how they affect their salary and lifestyle. The dataset looks like as follow:

```
In [4]: df.head()
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	co
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	Ur St
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	Ur St
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	Ur St
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	Ur St
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cu

rkclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	country	salary
e-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
-emp-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
ate	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
ate	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
ate	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

The data set consists of various data types from integer to floating to object as shown in Fig.

```
In [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education-num   32561 non-null  int64
 5   marital-status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital-gain    32561 non-null  int64
 11  capital-loss    32561 non-null  int64
 12  hours-per-week  32561 non-null  int64
 13  country         32561 non-null  object
 14  salary          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

In the raw data, there can be various types of underlying patterns which also gives an in-depth knowledge about the subject of interest and provides insights into the problem. But caution should be observed with respect to data as it may contain null values, or redundant values, or various types of ambiguity, which also demands pre-processing of data.

The dataset should therefore be explored as much as possible.
Various factors important by statistical means like mean, standard deviation, median, count of values and maximum value, etc. are shown below for numerical attributes.

```
In [6]: df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	32561.0	38.581647	13.640433	17.0	28.0	37.0	48.0	90.0
fnlwgt	32561.0	189778.366512	105549.977697	12285.0	117827.0	178356.0	237051.0	1484705.0
education-num	32561.0	10.080679	2.572720	1.0	9.0	10.0	12.0	16.0
capital-gain	32561.0	1077.648844	7385.292085	0.0	0.0	0.0	0.0	99999.0
capital-loss	32561.0	87.303830	402.960219	0.0	0.0	0.0	0.0	4356.0
hours-per-week	32561.0	40.437456	12.347429	1.0	40.0	40.0	45.0	99.0

Preprocessing of this dataset includes doing analysis on the independent variables like checking for null values in each column and then replacing or filling them with supported appropriate data types so that analysis and model fitting is not hindered from their way to accuracy. Shown above are some of the representations obtained by using Pandas tools which tell about variable count for numerical columns and model values for categorical columns. Maximum and minimum values in numerical columns, along with their percentile values for median, play an important factor in deciding which value to be chosen at priority for further exploration tasks and analysis. Data types of different columns are used further in label processing and a one-hot encoding scheme during the model building.

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.
- Developers can choose logging methods. Also, can choose database logging.
- 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 Database

The system needs to store every request into the database and we need to store it in such a way that it is easy to retain and look into the records.

The system should capture every data that any user gave and the prediction that has been made by that input.

2.4 Deployment

For the deployment process, we will be using Streamlit Sharing cloud to Deploy our model. Streamlit is used to make web apps as well as a cloud platform that lets companies build, deliver, monitor and scale apps.

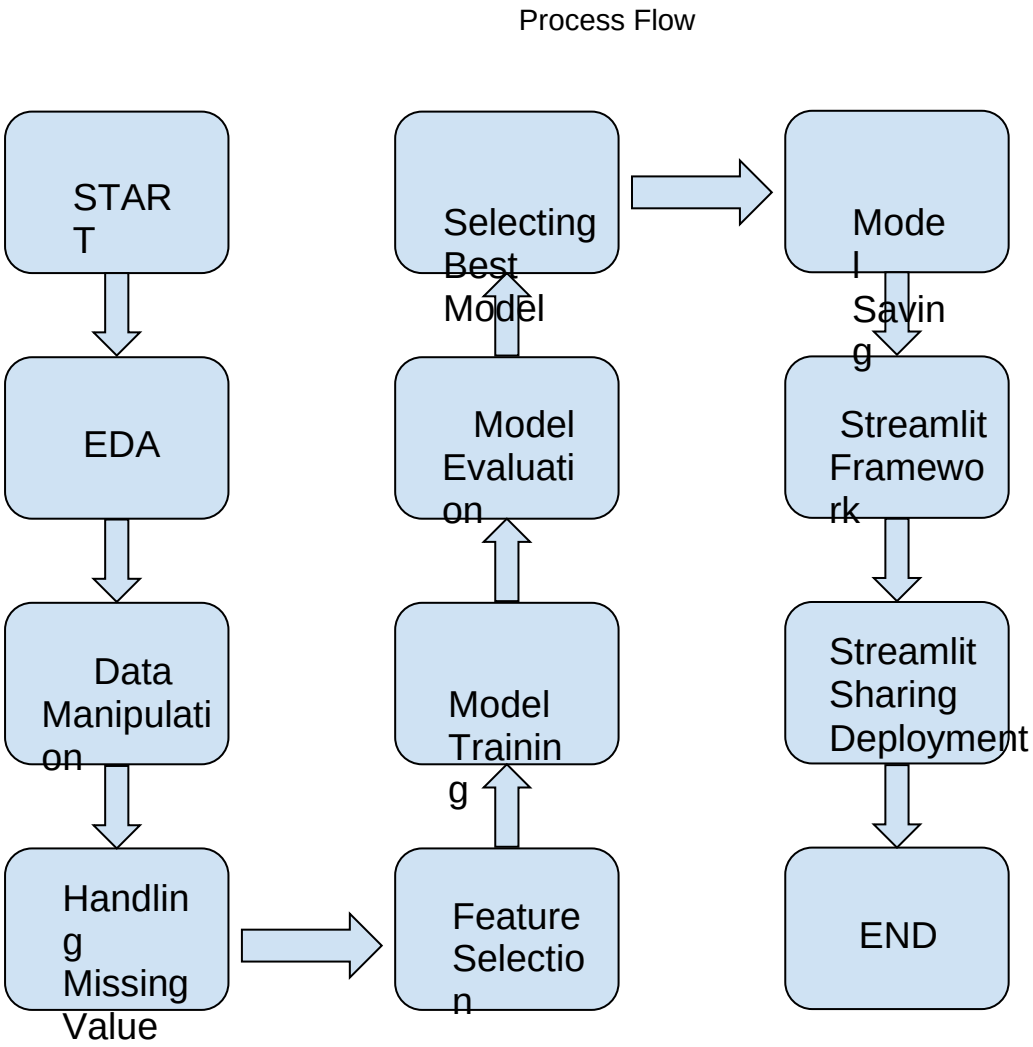
3. Technology Stack

Web App	Streamlit
Back End	Python
Deployment	Streamlit Sharing

4. Proposed Solution

We will use Exploratory Data Analysis (EDA) to find the Key relations between different attributes and will use a ML algorithm to predict the future sales demand. We can tell the company what are all the challenges they may face, what are the brands or products which is sold the most & other such kind of things, this helps sales team to understand which product to sell & which product to promote & other such kind of things.

5. Architecture



5.1 Data Gathering

Data source: https://www.kaggle.com/datasets/overload10/adult-census-dataset

5.2 Raw Data Validation

After data is loaded, various types of validation are required. 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 be much of use in determining of the salary category.

Like if any attribute is having zero standard deviation, it means that's all the values are the same, its mean is zero. Missing data in the training data stands as a problem in front of us.

5.3 Data Transformation

Before sending the data into the database, data transformation is required so that data are converted into such a form with which it can easily be inserted into the database. Here, the 'capital-gain' and 'capital-loss' attributes contain a vast amount of missing values. So, they are removed.

5.4 New Feature Generation

We haven't derived a new category.

5.5 Data Pre-processing

In data pre-processing all the processes required before sending the data for model building are performed. Like, here 'capital-gain' and 'capital-loss' the attributes are having some values equal to 0 no doubt both of these attributes are viable for prediction due to maximum entry being zero in the model they can't contribute much. So they have been removed from the dataset. In 'workclass' there were some fields '?' which ment null so the rows were removed. The column of 'hours-per-week' having numerical values was further categorised into brackets of 10 to 10 values. The column of 'marital-status' having string values with 7 different values were categorised into two categories.

5.6 Feature Engineering

After preprocessing it was found that some of the attributes are not important to the item sales for the particular outlet. So those attributes are removed. There are some columns that needs to be dropped as they don't seem to help in our analysis.

5.7 Model Building

After doing all kinds of preprocessing operations mention above and performing model training and testing the accuracy we came to the conclusion that Random Forest model have the highest accuracy with 81.68% accuracy.

5.8 Model Saving

Model is saved using pickle library in '.pkl' format.

5.9 Web app setup

After saving the model in .pkl file format we then create an app.py streamlit web app framework (Written in python) and then use requests to extract all the form selection selected by the user and then we predict the salary prediction by using the selected records by the user.

5.10 GitHub

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

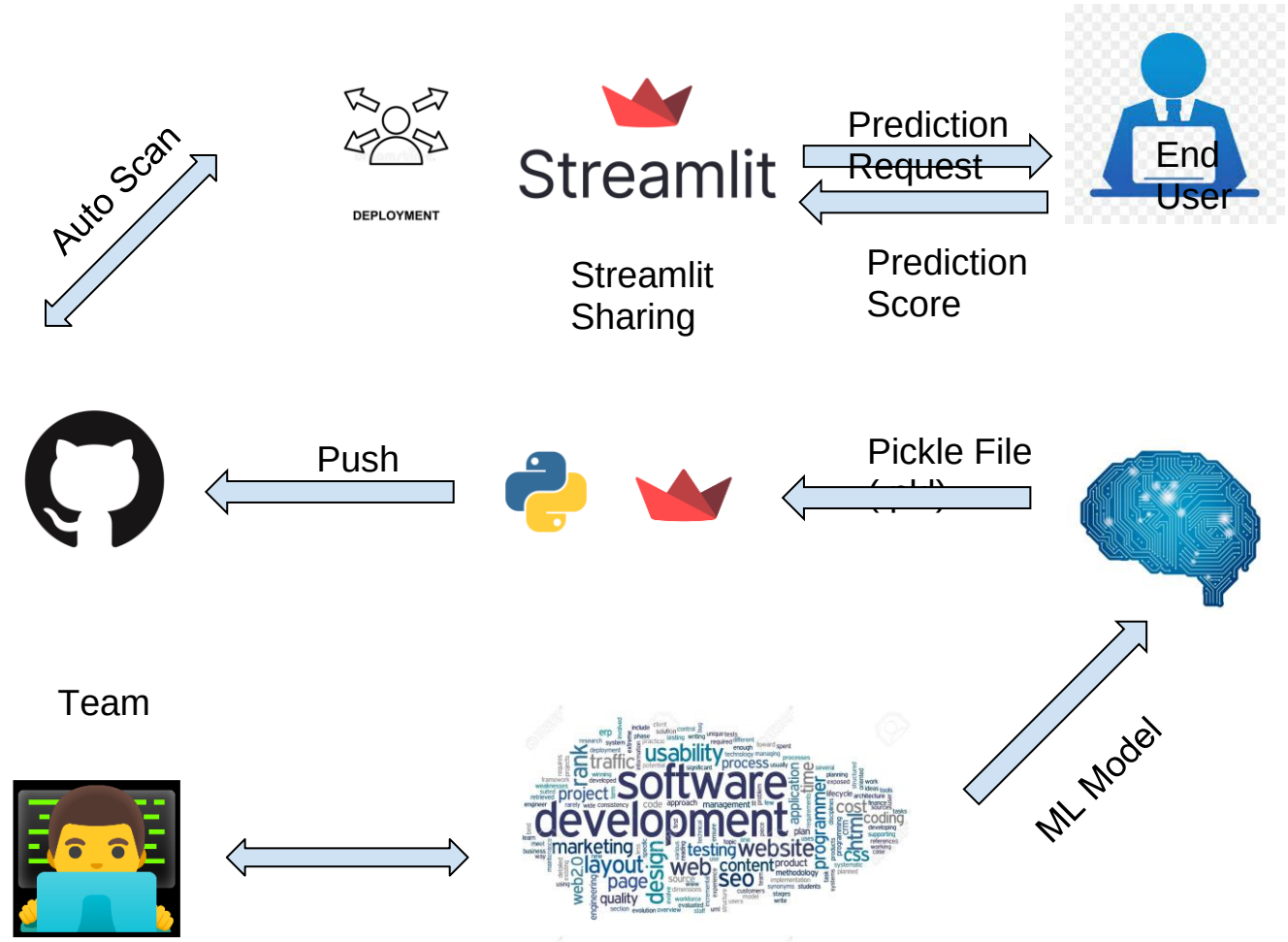
GitHub Project link: https://github.com/cursD15/Adult_Census_Income_Prediction

5.11 Deployment

The cloud environment was set up and the project was deployed from GitHub into the Heroku cloud platform.

WebApp link - https://cursd15-ineuron-project-app-te9xub.streamlit.app/

6. User Input / Output Workflow.


Auto Scan
DEPLOYMENT
Streamlit
Streamlit Sharing
Prediction Request
Prediction Score
End User
Push
Pickle File
Team
ML Model