

# Architecture

## ADULT CENSUS INCOME PREDICTION

### LLD History :-

Date	Version	Author
22/09/2021	V1.2	Umang Tank

### Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

## **Contents**

### Abstract

1. Introduction.....	6
1.1. Why this Architecture Design Documents?.....	6
2. Architecture.....	7
3. Architecture Description.....	8
3.1. Data Description.....	8
3.2. Data Transformation.....	8
3.3. Exploratory Data Analysis.....	9
3.4. Data Insertion into Database.....	10
3.5. Export Data from Database.....	11

3.6. Data Pre-processing.....	13
3.7. Model Building .....	13
3.8. Hyper Parameter Tuning.....	14
3.9. Model Dump.....	14
3.10. Data from User.....	15
3.11. Data Validation.....	15
3.12. Model Call for Specific Inputs.....	15
3.13. User Interface.....	16
3.14. Deployment.....	17
4. Technology Stack.....	18
6. Directory Tree Structures.....	19
5. Unit Test Cases .....	23

---

# 1. Introduction

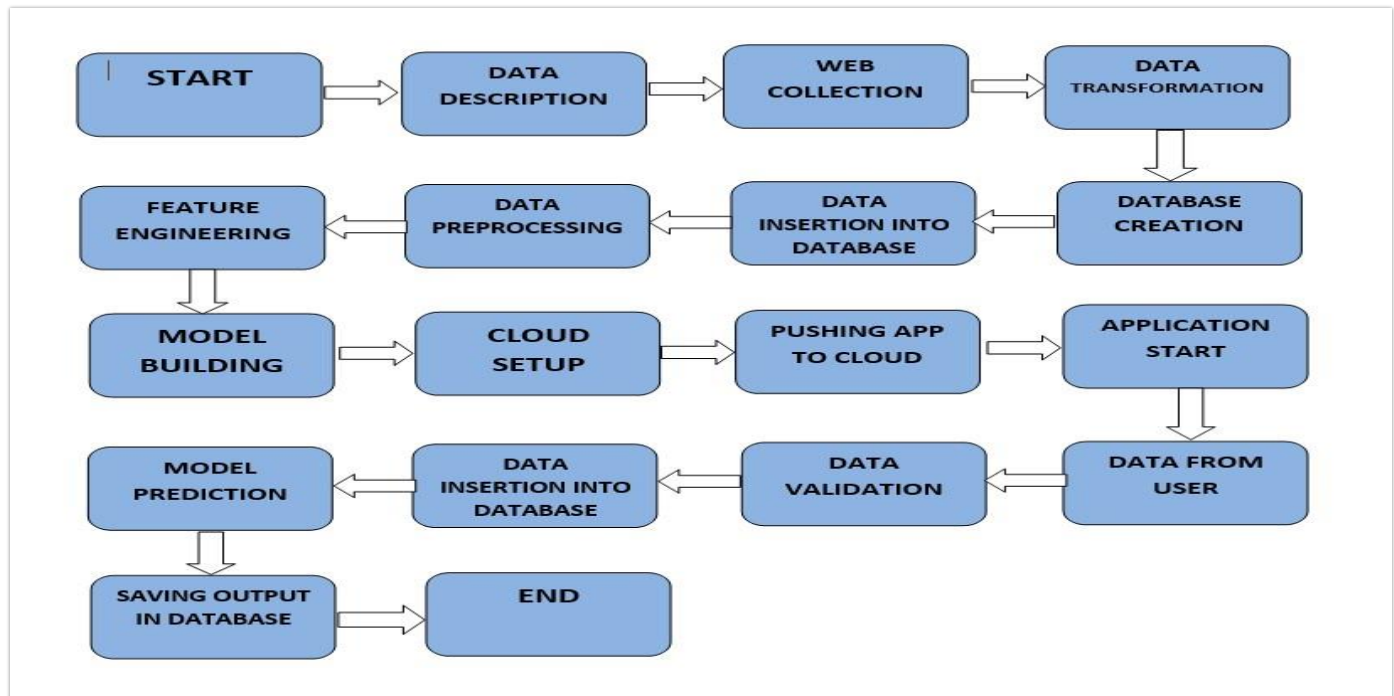
Why this Architecture Design Document?

The purpose of this document is to provide a detailed architecture design of the Adult Census Income Prediction Project by focusing on four key quality attributes:

**usability, availability, maintainability, testability.**

This document will address the background for this project, and the architecturally significant function requirements. The intension of this document is to help the development team to determine how the system will be structured at the highest level. Finally, the project coach can use this document to validate that the development team is meeting the agreed-upon requirements during the evaluation of the team's efforts.

## 2. Architecture



## 3. Architecture Description

### 3.1. Data Description

The dataset provided to us contains many rows, and 13 different independent features. We aim to predict if a person earns more than 50k\$ per year or not. Since the data predicts 2 values ( $>50K$  or  $\leq 50K$ ), this clearly is a classification problem, and we will train the classification models to predict the desired outputs.

**Age** — The age of an individual, this ranges from 17 to 90.

**Workclass** — The class of work to which an individual belongs.

**Education** — Highest level of education

**Education\_num** — Number of years for which education was taken

**Marital\_Status** — Represents the category assigned on the basis of marriage status of a person

**Occupation** — Profession of a person

**Relationship** — Relation of the person in his family

**Race** — Origin background of a person

**Sex** — Gender of a person

**Capital\_gain** — Capital gained by a person

**Capital\_loss** — Loss of capital for a person

**Hours\_per\_week** — Number of hours for which an individual works per week

**Native\_Country** — Country to which a person belongs

**Income** — The target variable, which predicts if the income is higher or lower than 50K\$.

### 3.2. Data Cleaning / Data Transformation

In the Cleaning process, we have cleaned up all the data because data is present in very bad format which was can not recognized by machine (Categorical data). So data Cleaning is done very first by data validation methods. Get\_dummies method use for Convert Categorical data in numerical format.

### 3.3. Exploratory Data Analysis

In EDA we have seen various insights from the data so we have selected which column is most important and dropped some of the columns by observing them plotting their heatmap from seaborn library also we done null value managed in an efficient manner and also implemented categorical to numerical transfer of column method here.

### 3.4. Data Insertion into Database

Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.

Table creation in the database.

Insertion of data in the table

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	age	capital_gain	capital_loss	education_year	working_hours	occupation	sex	marital	country	race	work_class	education	relation
▶	22	0	0	16	60	Adm-clerical	Male	Divorced	India	Amer-Indian-Eskimo	Federal-gov	10th	Husband
	40	0	0	16	60	Adm-clerical	Male	Divorced	India	Amer-Indian-Eskimo	Federal-gov	10th	Husband
	28	22000	0	12	40	Transport-moving	Male	Never-married	United-States	White	Private	Bachelors	Unmarried
	32	22000	0	18	34	Prof-specialty	Female	Married-civ-spouse	India	Other	Federal-gov	Doctorate	Not-in-family
	20	0	0	10	20	Sales	Female	Never-married	India	Amer-Indian-Eskimo	Private	Preschool	Unmarried
	65	10000	0	16	40	Protective-serv	Male	Married-spouse-absent	Jamaica	Black	Self-emp-inc	Some-college	Other-relative

### 3.5. Export Data from Database

Data Export from Database - The data in a stored database to be used for Data Pre-processing and Model Training.



### 3.6. Data Pre-processing

Data Pre-processing steps we could use are Null value handling, Categorical to Numerical Transformation of columns , Splitting Data into Dependent and Independent Features , Remove those columns which are does not participate in model building Processes , Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

### 3.7 Model Creation / Model Building

After cleaning the data and completing the feature Engineering/ data Per processing. we have done splitted data in the train data and test data using method build in pre-processing file and implemented various Classification Algorithm like RandomForestClassifier and DecisionTree, AdaBoost, Svm also calculated their accuracies on test data and train data.

	model	Precision	recall	f1_score	accuracy
0	LogisticRegression	0.727834	59.365738	65.393405	0.843738
1	DecisionTreeClassifier	0.634412	62.181387	62.804969	0.816835
2	AdaBoostClassifier	0.766728	62.151749	68.652807	0.858849
3	RandomForestClassifier	0.710831	63.604031	67.135930	0.845139
4	SVC	0.768444	51.244813	61.486486	0.840348

### 3.8 Hyperparameter Tuning

In hyperparameter tuning we have implemented grid search cv and from that we also implemented cross validation techniques for that.

### 3.9 Model Dump

After comparing all accuracies and checked all ROC, AUC curve accuracy we have choose AdaBoostClassifier as our best model by their results so we have dumped this model in a pickle file format with the python module.

### 3.10 Data from User

Here we will collect user's requirement to predict whether a Income is more than 50K a year or not.

## 3.11 Data Validation

Here Data Validation will be done, given by the user.

## 3.12 Model Call for specific input

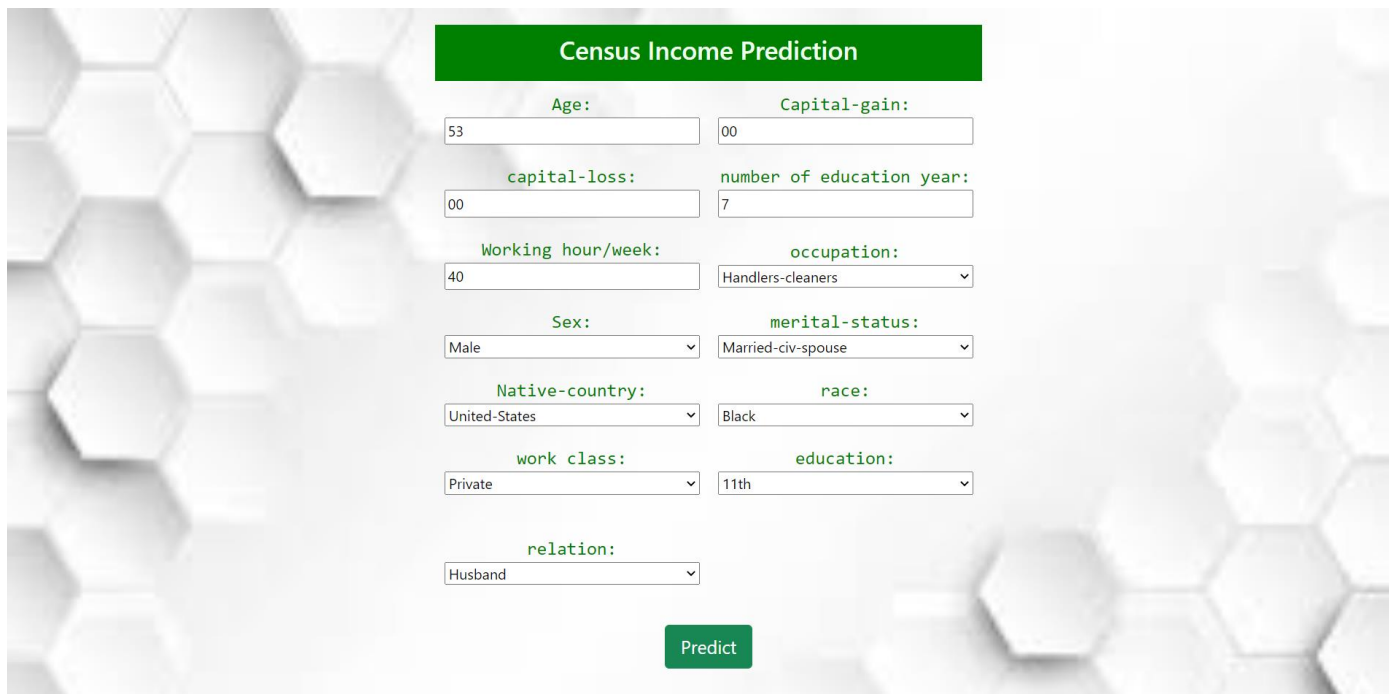
Based on the User input will be throwing to the backend in the variable format then it converted into pandas data frame then we are loading our pickle file in the backend and predicting whether product come in backorder or not as an output and sending to our html page.

## 3.13 User Interface

In Frontend creation we have made a user interactive page where user can enter their input values to our application. In these frontend page we have made a form which has beautiful styling with CSS and bootstrap. These HTML user input data is transferred in variable

format to backend. Made these html fully in a decoupled format.

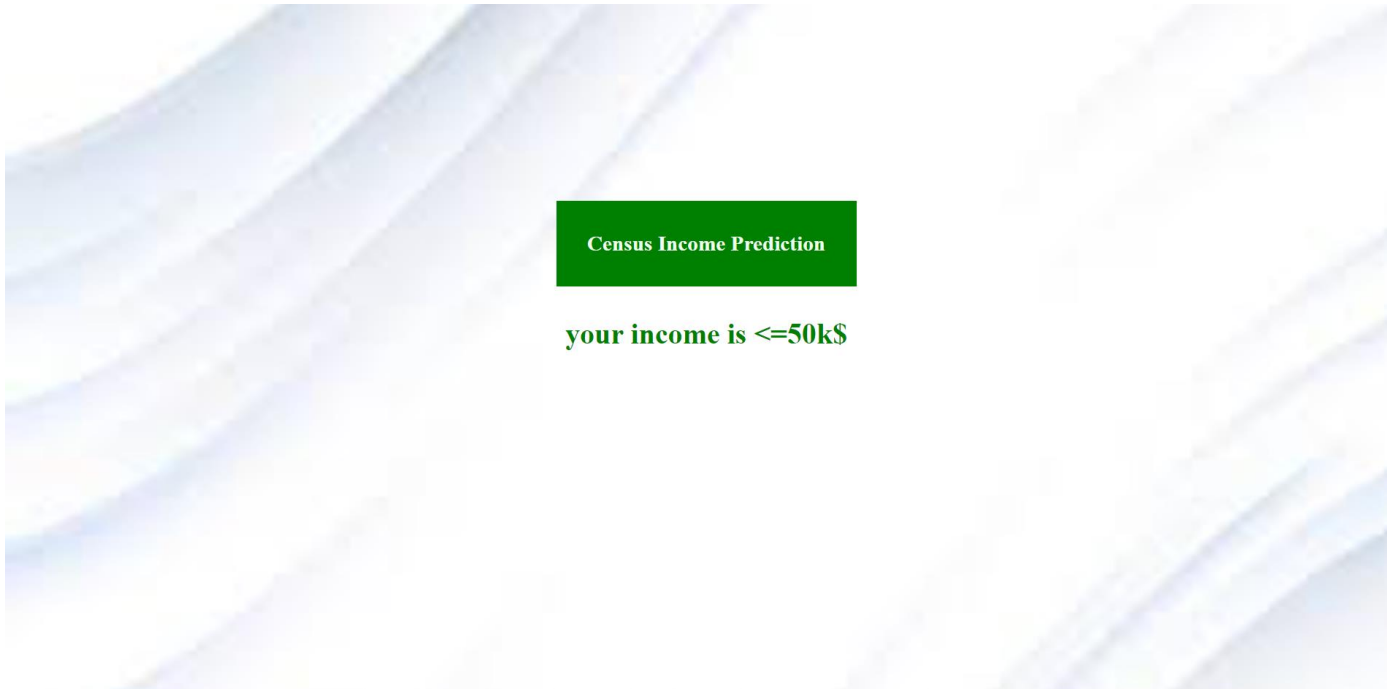
### Input page :-



The screenshot shows a web form titled "Census Income Prediction" with a green header. The form contains several input fields and dropdown menus for user data. The fields are arranged in a grid-like fashion. At the bottom right, there is a green "Predict" button.

Census Income Prediction	
Age:	Capital-gain:
<input type="text" value="53"/>	<input type="text" value="00"/>
capital-loss:	number of education year:
<input type="text" value="00"/>	<input type="text" value="7"/>
Working hour/week:	occupation:
<input type="text" value="40"/>	<input type="text" value="Handlers-cleaners"/>
Sex:	marital-status:
<input type="text" value="Male"/>	<input type="text" value="Married-civ-spouse"/>
Native-country:	race:
<input type="text" value="United-States"/>	<input type="text" value="Black"/>
work class:	education:
<input type="text" value="Private"/>	<input type="text" value="11th"/>
relation:	
<input type="text" value="Husband"/>	
<input type="button" value="Predict"/>	

### Output page :-



## 3.14 Deployment

We will be deploying the model with the help of Heroku cloud platforms.

This is a workflow diagram for the Back-Order Prediction Application .....

## 4. Technology Stack

Front End	HTML, CSS, Bootstrap File
Back End	Flask, Pandas, NumPy, scikit-learn etc
Database	MySQL
Deployment	Heroku

## 5.Directory Tree Structure

```
├── static
│   ├── images
├── templates
│   ├── welcome.html
│   ├── result.html
├── utils
│   ├── all_utils.py
```

## Architecture

```

|  ├── __init__.py
|  ├── Procfile
|  ├── README.md
|  ├── app.py
|  ├── Census_Income_Prediction.ipynb
|  ├── models
|  └── requirements.txt

```

## 6.Unit Test Case

Test Case Description	Pre – Requisite	Expected Resulted
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed

Verify whether the User is able to Enter the data	1. Application is accessible	The User should be able to enter the data
Verify whether user is able to successfully Enter all the data.	1. Application is accessible 2. User is Enter all the data	User should be able to successfully Enter all the data.
Verify whether user is giving standard input.	1. Handled test cases at backends.	User should be able to see successfully valid results.
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is logged in to the application	User should be able to edit all input fields
Verify whether the model giving the predicted output.	1.Application is accessible 2. User is able to see predicted output.	User is able to see predicted output.
Verify whether user is presented with recommended results on clicking submit	1.Application is accessible 2. user is presented with recommended results on clicking submit	User should be presented with recommended results on clicking submit