Detail Project Report Salary Prediction

Revision Number – 1.2

Last Date of Revision: 26 - 11 -2022

Nitesh
Sharma

Document Version Control

Date	Version	Description	Author
26 – 11- 2022	1.0	Introduction General Description	Nitesh
26– 11- 2022	1.1	Technical Requirements Data Requirements Data Preprocessing Design Flow	Nitesh
26 – 11 - 2022	1.2	Deployment Conclusion	Nitesh

Contents

1. Introduction

1.1 Why this DPR Document ?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and the written code. This also provides the detailed description on how the entire project has been designed end-to-end.

Key points :

- Describes the design flow
- Implementations
- Software requirements Architecture of the project Non-functional attributes like:
- Reusability
- Portability
- Resource utilization

2. General Description

2.1 Problem Statement

The Goal is to predict whether a person has an income of more than 50K a year or not. This is basically a binary classification problem where a person is classified into the >50K group or <=50K group.

2.2 Proposed Solution

To solve the problem, we have created a User interface for taking the input from the user to predict the Salary using our trained ML model after processing the input and at last the output (predicted value) from the model is communicated to the User.

2.3 Further Improvements

We also analyze the data used for training the ML model by considering different angles of business. If we use such information and predict the salary it will help the organization .

3. Technical Requirements

As technical requirements, we don't need any specialized hardware for virtualization of the application. The user should have the device that has the access to the web and the fundamental understanding of providing the input.

3.1 Tools Used

- Python 3.9 is employed because of the programming language and frameworks like NumPy, Pandas, Scikit - learn and alternative modules for building the model.
- Jupyter - Notebook is employed as an IDE.
- For Data visualizations, seaborn and components of matplotlib are getting used.
- For information assortment prophetess info is getting used.
- Front end development i use streamlit framewok.
- GitHub is employed for version management.
- AWS beanstalk is employed for deployment.

4. Data Requirements

The Data requirements totally supported the matter statement and also the dataset is accessible on the kaggle within the file format of (.csv).

4.1 Data Collection

The data for these project is collected from the Kaggle Dataset, by using kaggle api.

4.2 Data Description

Adult Census dataset is 30K+ dataset publicly available on kaggle. dataset contain 15 columns named age, workclass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, country, salary.

5. Data Preprocessing

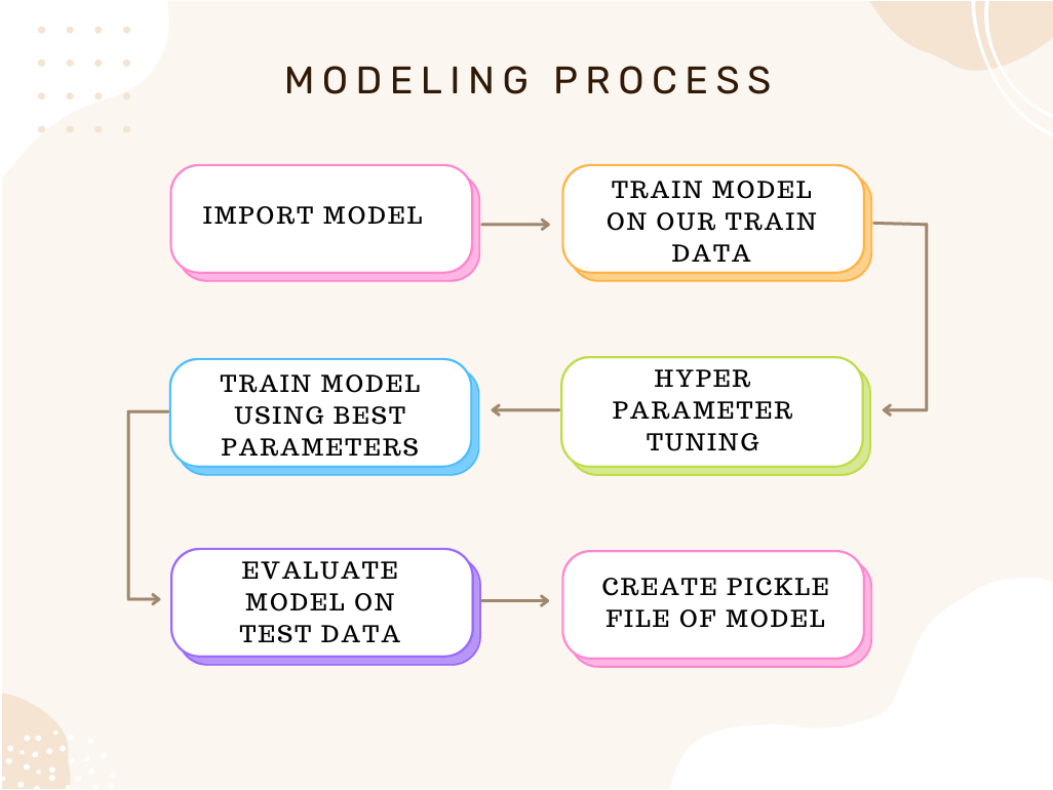
- Checked for info of the Dataset, to verify the correct datatype of the Columns.
- Checked for Null values, because the null values can affect the accuracy of the model.
- Converted all the desired columns into Datetime format.
- Performed One - Hot encoding on the desired columns.
- Checking the distribution of the columns to interpret its importance.

Now, the info is prepared to train a Machine Learning Model.

6. Design Flow

6.1 Model Creation and Evaluation

After preprocessing the data, we randomly spread our data into train and test data. using train data to train our model and test data to evaluate our trained model. we use logistic Regression on train data to predict the Salary is <=50K or >50k. create a pickle file of our train model with our data preprocessing steps. upload pickle file in aws S3 bucket.



6.2 UI Interface

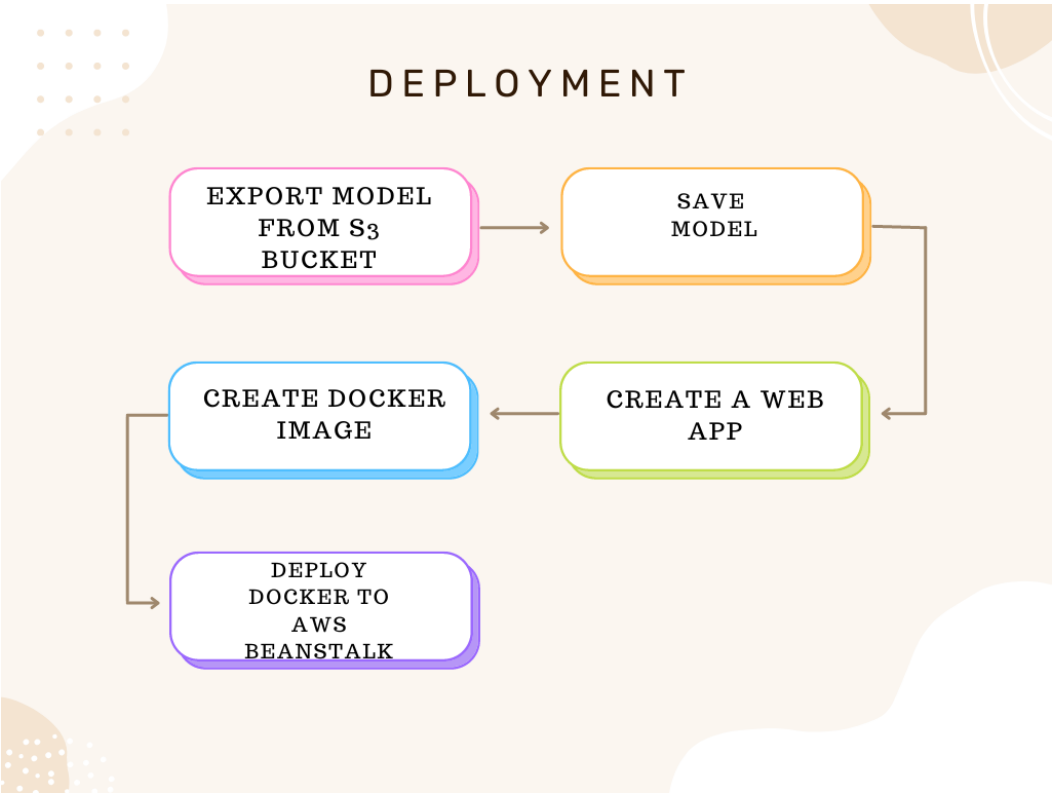
- UI is created using streamlit framework. take input from user unsing streamlit objects like slider, container, expender etc

6.3 Logging

In logging, at each time an error or an exception occurs, the event is logged into the system log file with reason and timestamp. This helps the developer to debug the system bugs and rectify the error.

7 Deployment

Using ci-cd pipeline, we config github action to create docker image of continuous changes and deploy that docker image to aws beanstalk. so that user can access the project from internet.



8. Conclusion

The salary prediction system is help the organization and government in taking decision based on people's salary based on changes in their occupation and education etc.