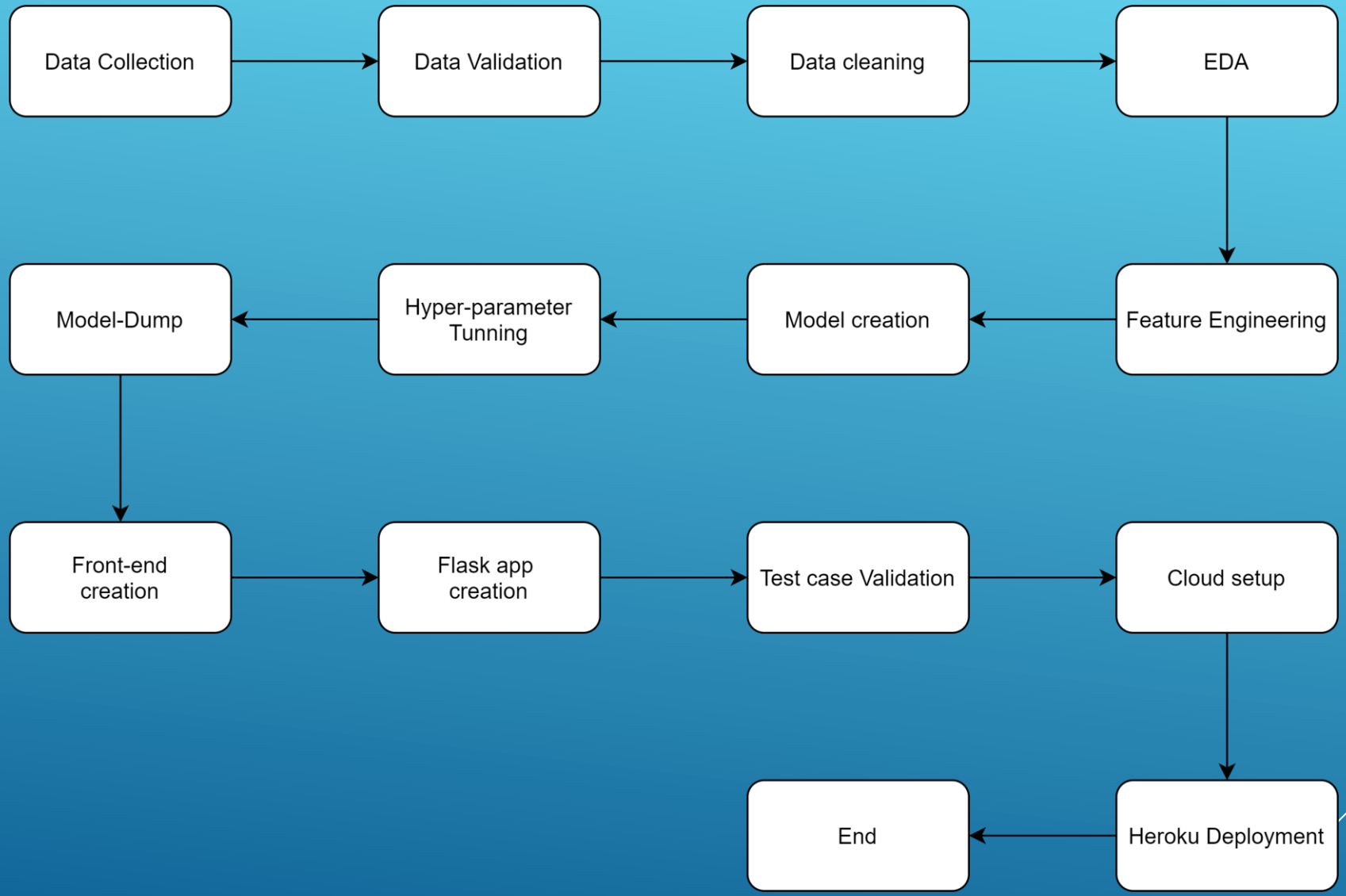
ADULT CENSUS INCOME PREDICTION

Objective:

- Adult Census Income prediction (ACI) is a Machine Learning classification model-based modelling where we try to predict the income group of a person and take necessary actions.

Benefits:

- The solution proposed here is ACI (Adult Census Income prediction) is used to perform above use cases, if there any person with less than 50k income it categorizes into less category and vice versa. This study helps in knowing the economy of the country and it helps in solving income equality problem


Data Collection
Data Validation
Data cleaning
EDA
Feature Engineering
Model creation
Hyper-parameter Tunning
Model-Dump
Front-end creation
Flask app creation
Test case Validation
Cloud setup
Heroku Deployment
End

- Data Collection:

We have Dataset of row columnar which includes various columns like age , occupation, education ,etc . The information is given in csv format. These data is collected from kaggle site.

- Data Validation:

Data collected from Data source is validated on certain criteria and data after getting validated is send to next stage for Data cleaning step .

- Data cleaning:

In the Cleaning process, We have cleaned up all the data because data is present in very bad format which was cannot recognized by machine. So data engineering is done very first.

- EDA:

In EDA we try to perfrom necessary Exploratory data analysis steps like finding missing values, replacing missing values, removing duplicate values, classifying features into categorical and numerical.

- Feature Engineering:

In Feature Engineering we try to encode variable, perform standardization by using standard scalar, split data into dependent and Independent variables for futher model building purpose

- Model Creation:

After cleaning the data and completing the feature engineering. we have done data splitting in the train data and test data and implemented classification algorithms like Random Forest classifier and XGboost classifier also calculated their accuracies on test data

- Hyperparameter Tuning: In hyperparameter tuning we have implemented various ensemble techniques like random forest regressor, bagging and boosting we also done randomized search cv or grid search cv and from that we also implemented cross validation techniques for that. From that we have choosen best parameters according to hyperparameter tunning and best score from their accuracies so we got 80% accuracy in our XGboost Classifier after hyper parameter tuning.

- Model Dump: After comparing all accuracies and checked all roc, auc curve we have choosen hyperparameterized XG boost classifier as our best model by their results so we have dumped these model in a pickle file format with the help of python pickle module.

- Front-end creation:

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 is of HTML. These html user input data is transferred in json format to backend. Made these html fully in a decoupled format

- Flask-app creation:

Here we try create Flask app which acts as an api between user and and our backend ,by hitting our flask our application gets starts and gets running.

- Deployment:

We will be deploying the model to Heroku.

Q1) What’s the source of data?

Ans: We collect the data from Kaggle platform ,where the data is in the form of rows and columns

Q2) What was the type of data?

Ans: The data was the combination of numerical and Categorical values.

Q 3) What’s the complete flow you followed in this Project?

Ans: Refer slide 3 for better Understanding

Q4) How logs are managed?

Ans: We are using different logs as per the steps that we follow in validation and modeling like Data validation log , Data Insertion ,Model Training log , prediction log ,etc.

Q5) What techniques were you using for data pre-processing?

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output variables
- Checking and changing Distribution of continuous values
- Handling missing values
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.
- Scaling the data

Q6) How training was done or what models were used?

- After performing required preprocessing we try to divide data into train and test .
- The scaling was performed over training and validation data
- Algorithms like XGBoost and Random forest classifier are used for modelling and model is saved for further Prediction .

Q8) How Prediction was done?

The test data is collected from user then data is sent for required for preprocessing and model training then after performing required scaling and transformation we use pickle file for prediction of user data

Q9) What are the different stages of deployment?

Deployment is done in Heroku platform, where we try to first push our code into github repo, then we try to connect our Heroku login with github repo to deploy our project .

Finally after deploying our project code we will be getting deploy link where we can access our project