Project Architecture

Adult Census Income Prediction

07/7/2024

Jyoti Pandey

1. Project Architecture

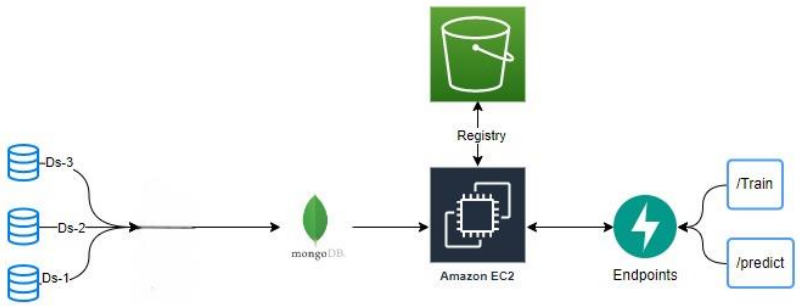


Fig 1: Data Ingestion

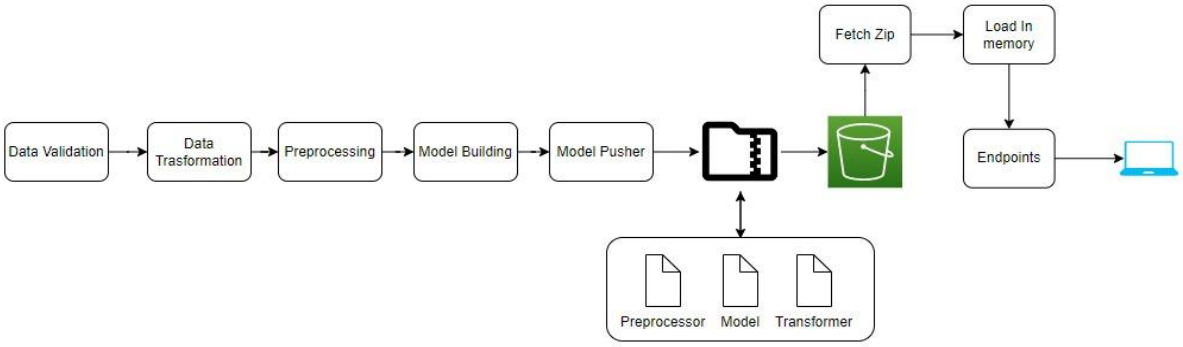
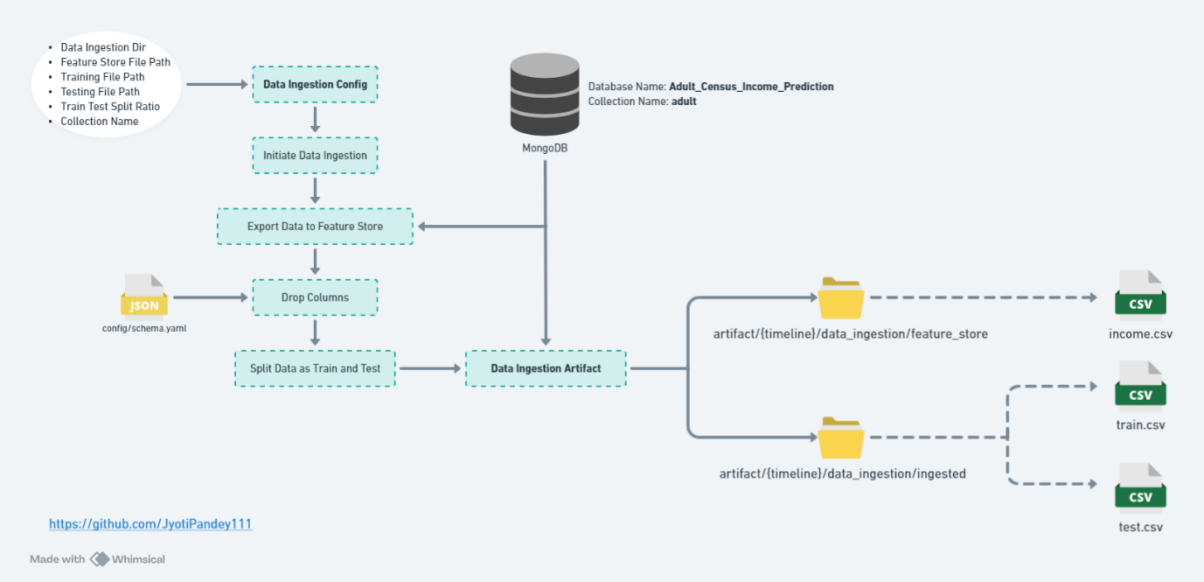


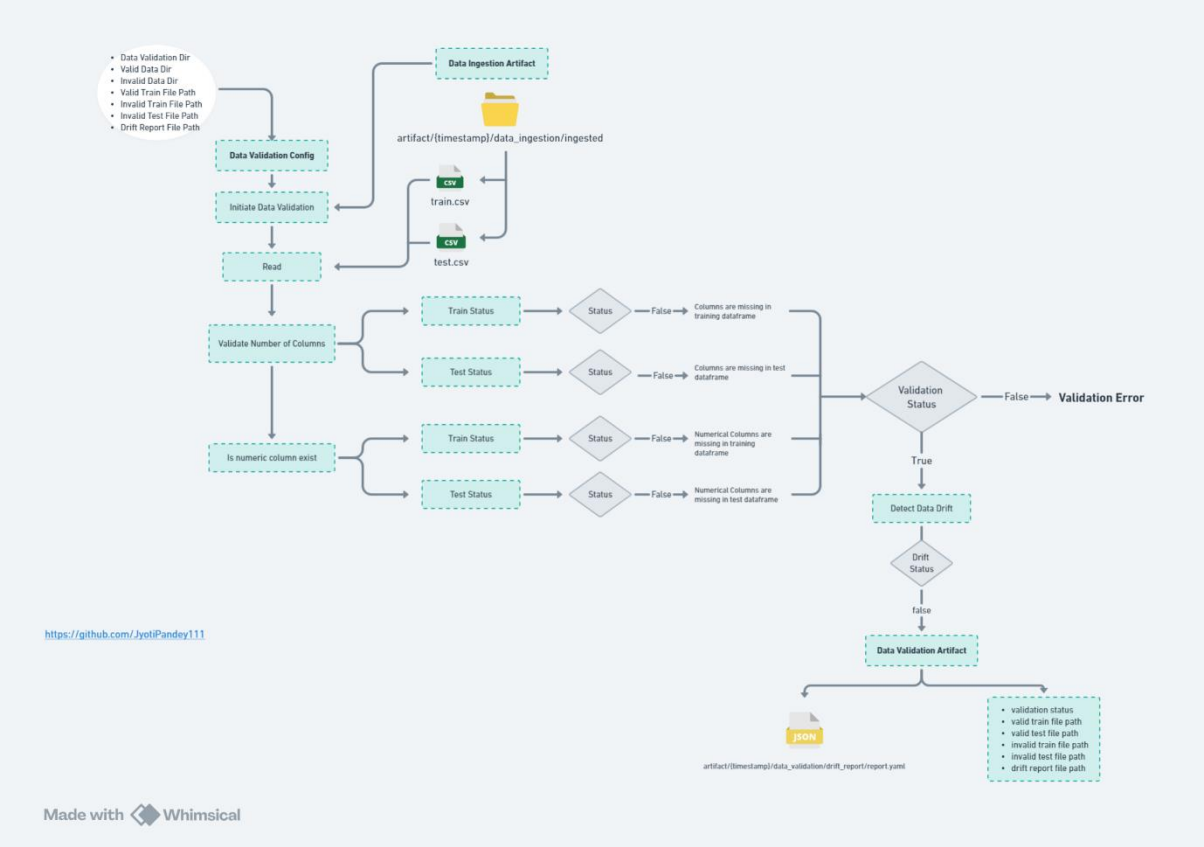
Fig 2: Project Architecture

DATA INGESTION:



1. Uploaded the data to MongoDB database and retrieved the data from MongoDB by creating connection in data ingestion component.
2. The duplicate rows and columns which are not relevant to our goal will be dropped.
3. After retrieving the data from MongoDB, split the dataset into train and test dataset according to defined train and test split ratio.
4. Save the data ingestion output as train file path and test file path as data ingestion artifacts

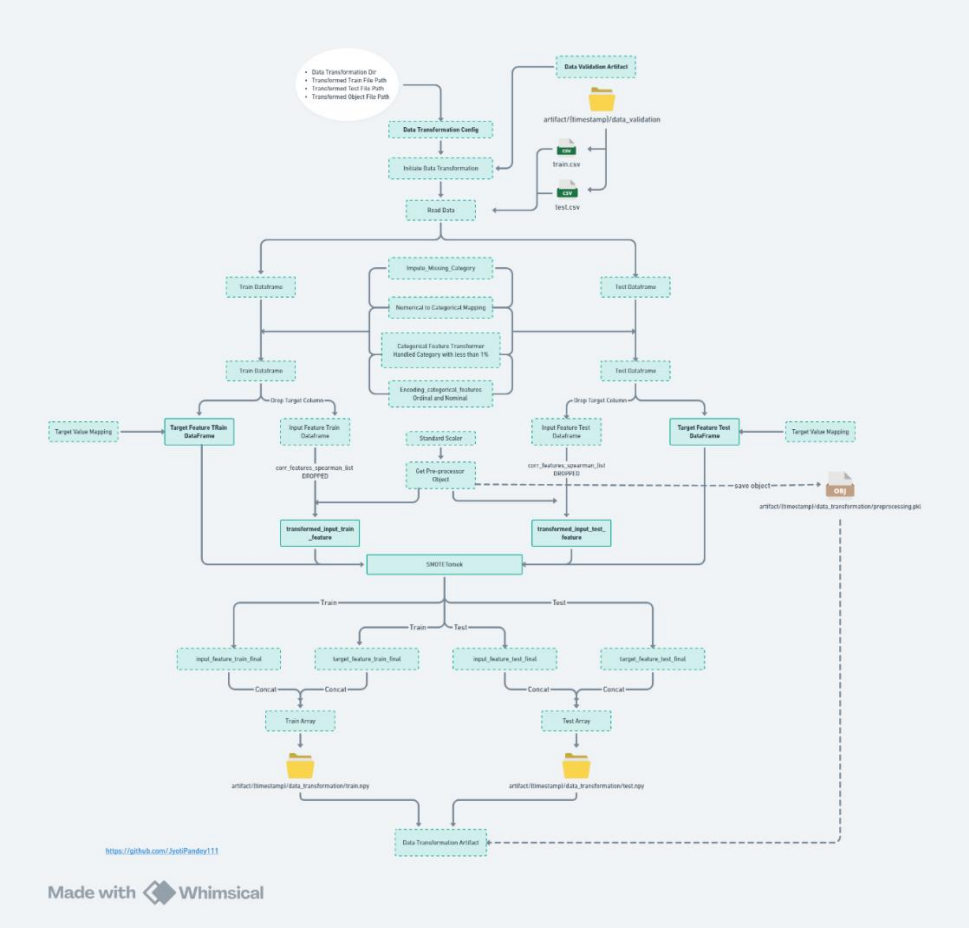
DATA VALIDATION:



In this component, data is validated data which is retrieved from MongoDB database. Following are steps in this component:

1. Reading data from train and test file path that is the artifact of data ingestion component
2. Validating the number of columns if all columns are not present then raise a error.
3. Validate the numerical columns if there is any missing column then log about it.
4. Check of data drift in order to determine that the sample is from same population.
5. Save the valid train dataset and valid test dataset as artifact for the data validation component.

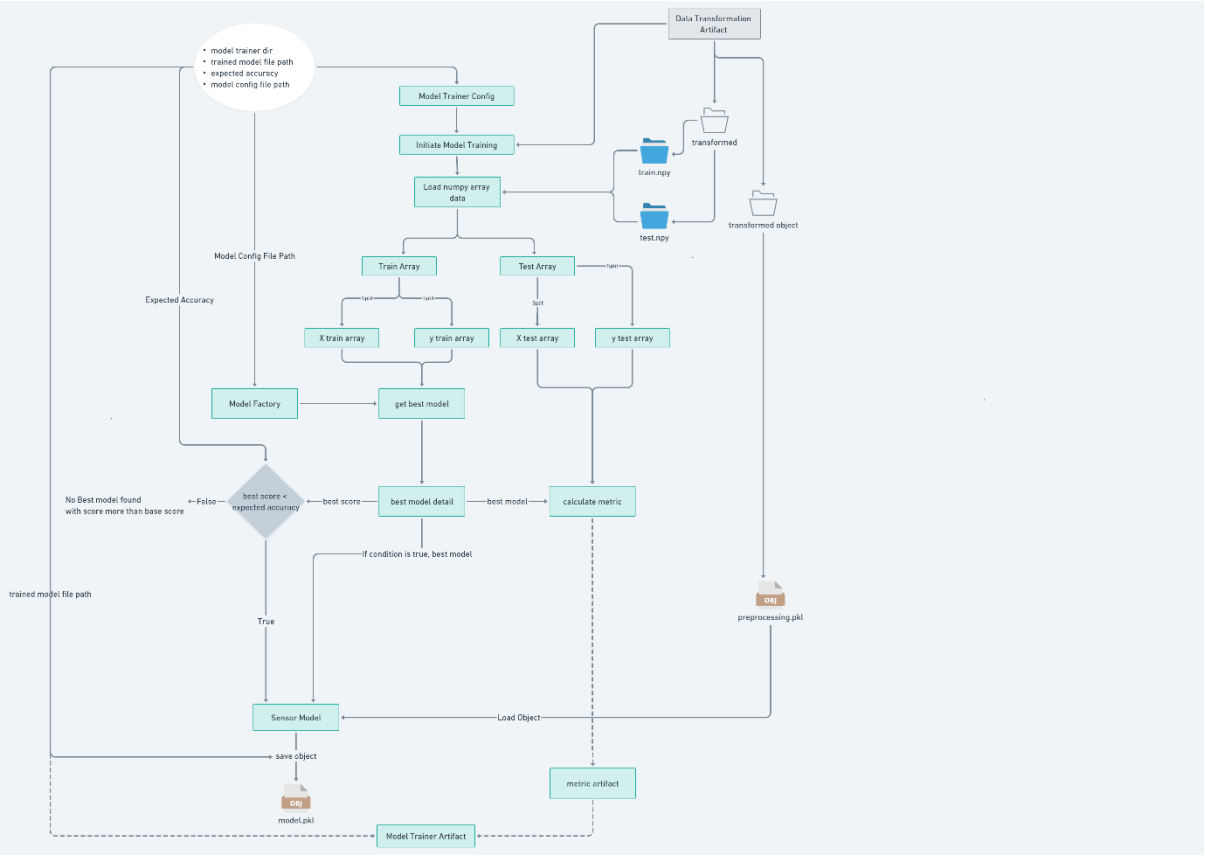
DATA TRANSFORMATION:



In this component, data has checked and proper techniques are used to handle missing values, incorrect values, categorical feature handling, outliers handling, feature extraction, feature manipulation. This component has following steps:

1. Load the files from the data validation artifact. Read the train and test dataset.
2. Decide the data transformation configuration and initiate the data transformation.
3. Handle the missing values, convert the numerical features to categorical features, handle the rare category, and encoding categorical features for train and test dataset separately.
4. Separate the input features and target features of train and test dataset separately and perform the target value mapping for encoding the target values in binary target values.
5. Drop the correlated input feature using spearman correlation, perform the standard scaler and build preprocessor object.
6. Perform the SmoteTomek Sampling on transformed train and transformed test dataset separately.
7. Save the Train and test array as data transformation artifact.

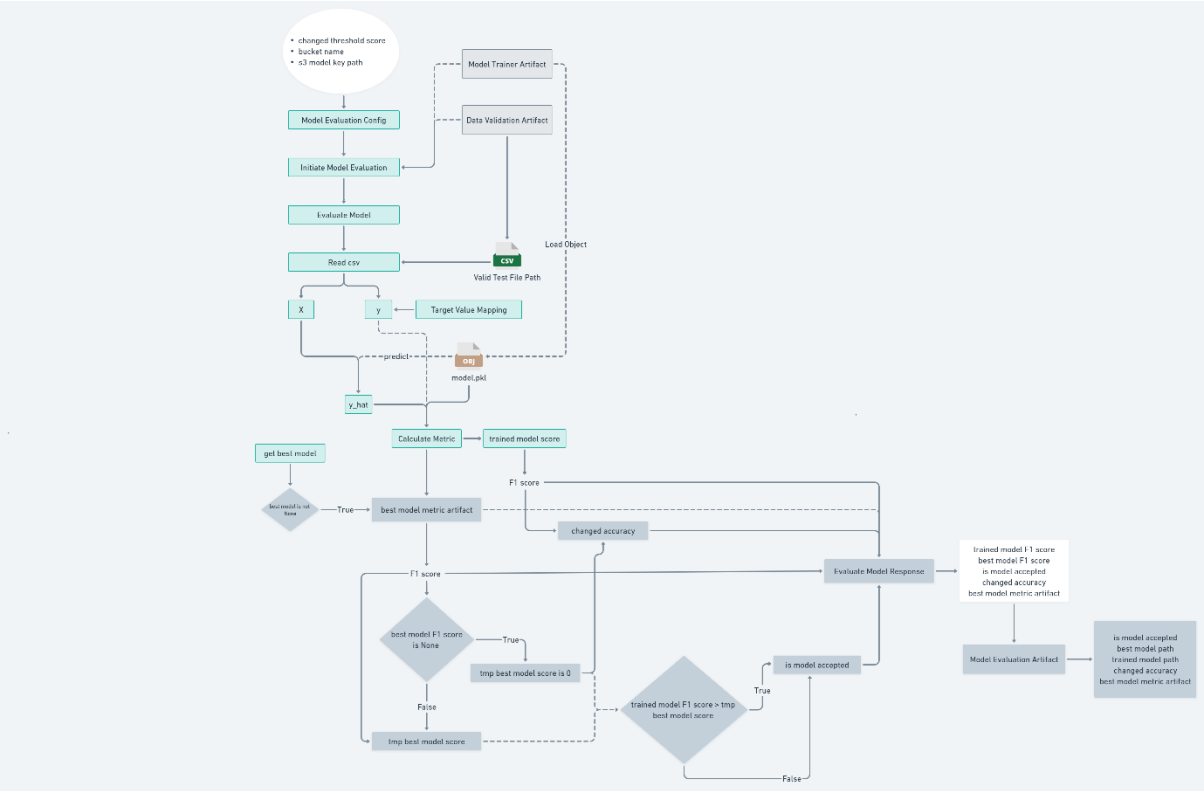
MODEL TRAINER:



In this component of project, feature selection has been done and a model is created which performs the best among all the classification Machine Learning algorithms. The steps followed in model trainer components are as follows:

1. Get the data transformation artifacts and create the model training configuration.
2. Split the train and test array into input feature and output (dependent) feature as X_train, y_train, X_test, and y_test.
3. Train the model on X_train and y_train and get the best model by testing in X_test and y_test.
4. Check is the model performance is better the base model if yes the create the pickle file and save it to the model artifact for future use.

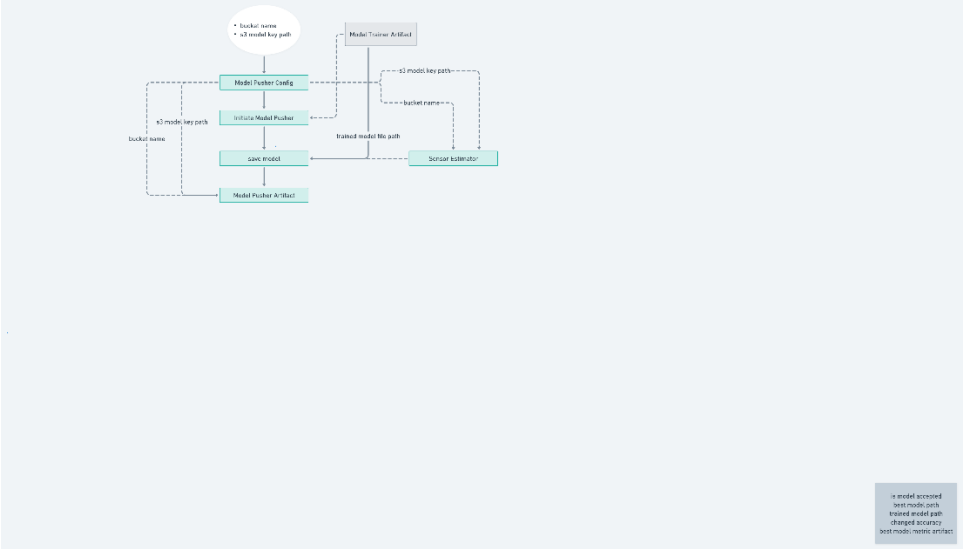
MODEL EVALUATION:



In model evaluation component, the best model performance is evaluated on the basis of the best model score. This component is used after evaluating the best model in order to maintain the performance of model for near future. The steps to be followed are as follows:

1. Load object from model trainer artifact and load valid test file from data validation artifact.
2. Split the test dataset into input and output data, make predictions from the best model we have loaded from model train artifact and evaluate the performance.
3. If trained model f1 score is better than temporary best model, accept the trained model as the best model and change the accuracy.
4. Save the model evaluation artifact.

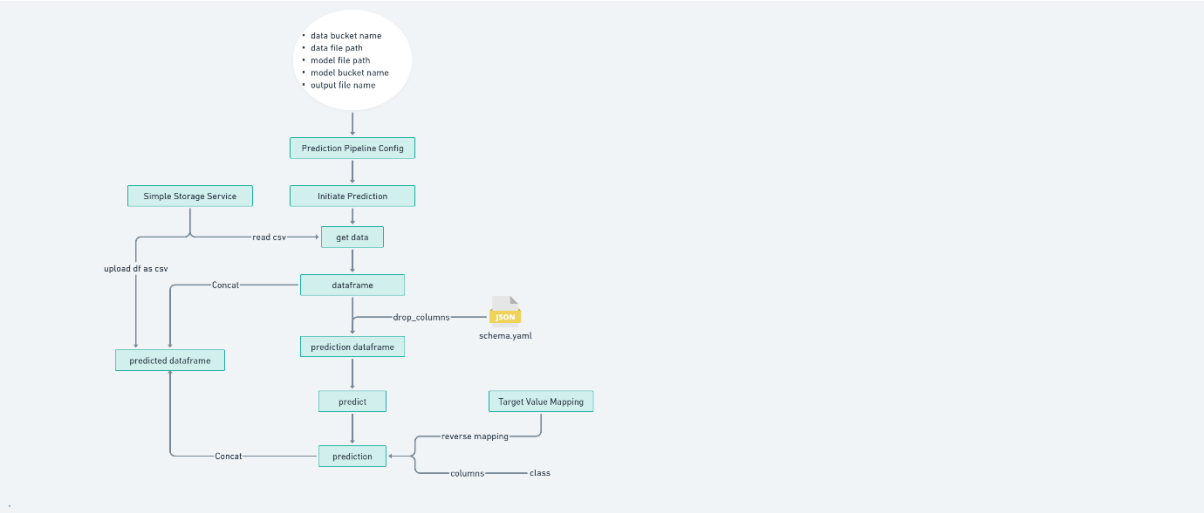
MODEL PUSHER:



This model pusher component is used to push the best model to system. The steps followed in the component are as follows:

1. Load the model trainer artifact and create the model pusher configuration.
2. Initiate the model pusher, create the save model directory and save the best model into this directory.
3. Also save the model in to the model pusher artifact.

PREDICTION PIPELINE:



The steps for prediction pipeline are as follows:

1. Get the data from FastAPI which user is entering to the respective fields.
2. Convert the data into dataset to make the prediction for the entries done by user.
3. Perform the respective feature engineering which was performed while training the model which are numerical to categorical mapping, encoding the categorical data.
4. If any column which is not present in the entries made by user assign zero as a value for that particular column value.
5. Drop the correlated feature from the dataset.
6. Model resolver will make prediction with the help of saved model and save the predicted value in a new column.
7. Create a HTML page to show the entries by the user and the prediction made by the model at one place.

Deployment Architecture Workflow

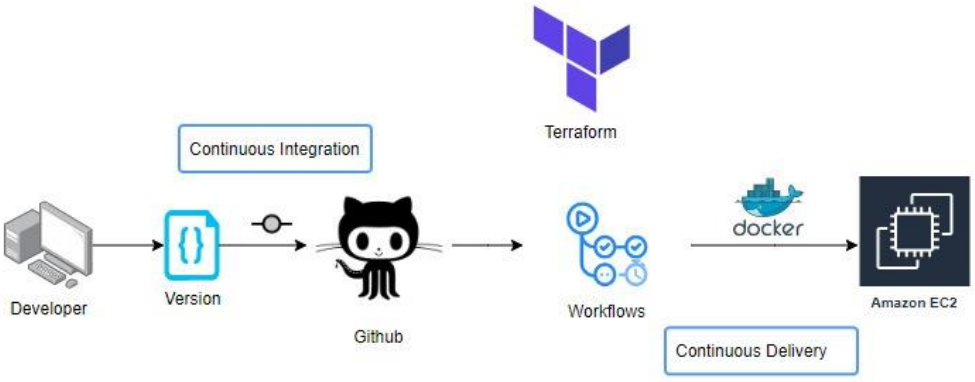


Fig 3: Deployment Architecture

2. Architecture Description

Income Dataset Overview

The dataset named Adult Census Income is available in kaggle and UCI repository. This data was extracted from the 1994 census bureau dataset by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics). **The prediction task is to determine whether a person makes over $50K a year or not.**

There are 32561 rows and 15 columns in the dataset.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	age	workclass	fnlwgt	education	education-	marital-sta	occupatio	relationshi	race	sex	capital-gai	capital-los	hours-per-	country	salary	
2	39	State-gov	77516	Bachelors	13	Never-ma	Adm-cleri	Not-in-fa	White	Male	2174	0	40	United-St	<=50K	
3	50	Self-emp-	83311	Bachelors	13	Married-c	Exec-man	Husband	White	Male	0	0	13	United-St	<=50K	
4	38	Private	215646	HS-grad	9	Divorced	Handlers-	Not-in-fa	White	Male	0	0	40	United-St	<=50K	
5	53	Private	234721	11th	7	Married-c	Handlers-	Husband	Black	Male	0	0	40	United-St	<=50K	
6	28	Private	338409	Bachelors	13	Married-c	Prof-speci	Wife	Black	Female	0	0	40	Cuba	<=50K	
7	37	Private	284582	Masters	14	Married-c	Exec-man	Wife	White	Female	0	0	40	United-St	<=50K	
8	49	Private	160187	9th	5	Married-s	Other-ser	Not-in-fa	Black	Female	0	0	16	Jamaica	<=50K	
9	52	Self-emp-	209642	HS-grad	9	Married-c	Exec-man	Husband	White	Male	0	0	45	United-St	>50K	
10	31	Private	45781	Masters	14	Never-ma	Prof-speci	Not-in-fa	White	Female	14084	0	50	United-St	>50K	
11	42	Private	159449	Bachelors	13	Married-c	Exec-man	Husband	White	Male	5178	0	40	United-St	>50K	
12	37	Private	280464	Some-coll	10	Married-c	Exec-man	Husband	Black	Male	0	0	80	United-St	>50K	
13	30	State-gov	141297	Bachelors	13	Married-c	Prof-speci	Husband	Asian-Pac	Male	0	0	40	India	>50K	
14	23	Private	122272	Bachelors	13	Never-ma	Adm-cleri	Own-child	White	Female	0	0	30	United-St	<=50K	
15	32	Private	205019	Assoc-acd	12	Never-ma	Sales	Not-in-fa	Black	Male	0	0	50	United-St	<=50K	

2.2 Predicting Income

1. User enters the values for age, capital gain, capital loss, hours per week.
2. The User chooses the work class, education marital status, occupation, relationship, race sex, country by clicking on one of the available options.
3. The system presents the set of inputs required from the user.
4. The user gives required information.
5. The system should be able to predict whether income is greater than 50K dollars or not based on the user information.