Low-Level Design for Income Prediction Model

Components

1. ***Data Retrieval (retrieve_data.py):***

   MongoDB Connection: Connects to MongoDB using parameters from params.yaml.
   Query Data: Executes queries to retrieve relevant data.

2. ***Data Transformation (data_transformation.py):***

   Data Cleaning: Handles missing values, duplicates, and outliers.
   Feature Engineering: Creates new features to enhance model performance.
   Data Encoding: Encodes categorical variables for model compatibility.

3. ***Model Training (model_training.py):***

   Model Selection: Tests multiple models (Decision Tree, Random Forest, XGBoost, etc.).
   Hyperparameter Tuning: Grid search for optimal hyperparameters.
   Evaluation Metrics: Calculates and logs metrics like accuracy, precision, recall, and ROC AUC.
   Model Saving: Saves the best model for deployment.

4. ***Web Application (app.py):***

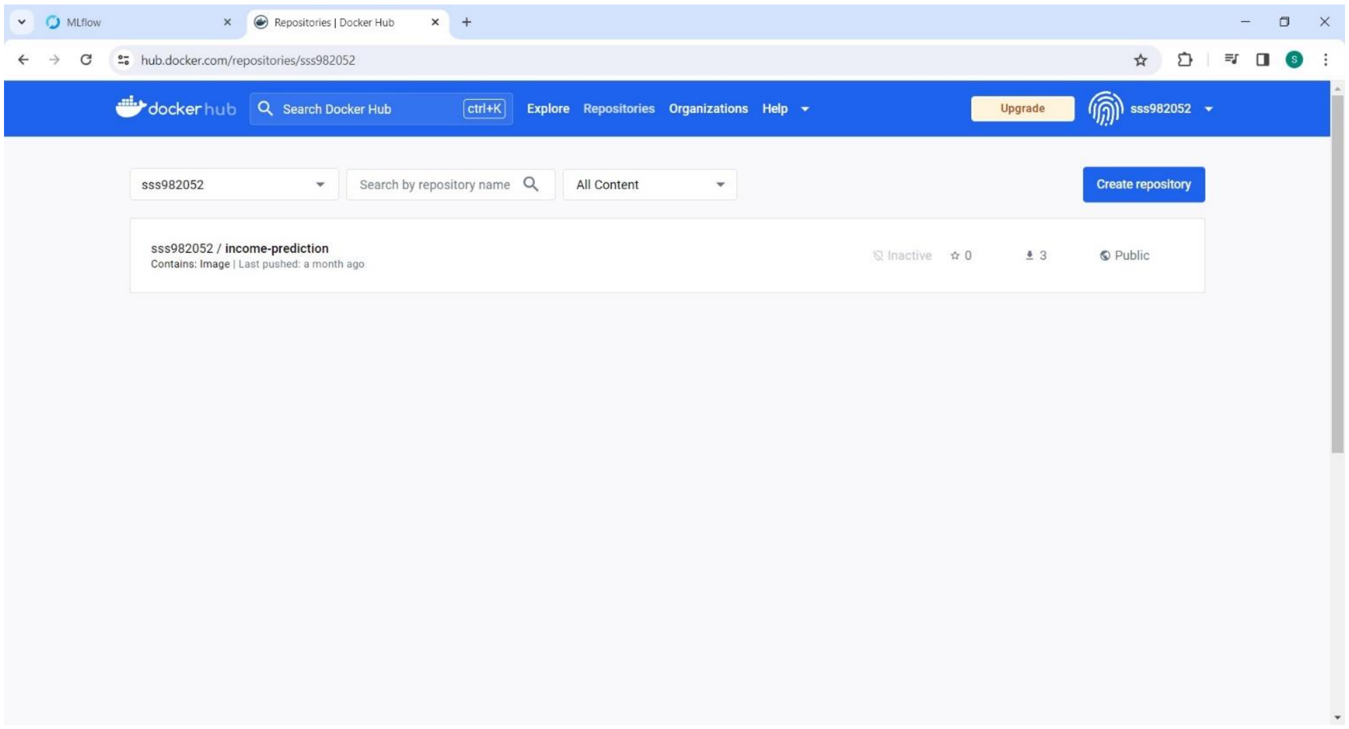
   Flask Routes: Defines routes for home page and prediction.
   HTML Templates: Uses templates for rendering web pages.
   Form Handling: Processes user input and sends data for prediction.
   Logging: Logs errors and messages for troubleshooting.

5. ***MLflow Integration (mlflow_artifacts/):***

   MLflow Tracking: Logs experiments, parameters, metrics, and artifacts.
   MLflow Server: Runs locally to facilitate tracking and management.

6. ***Docker Image:***

   Dockerfile: Specifies steps for building the Docker image.
   Dependencies: Includes necessary libraries and environment setup.
   Flask Configuration: Configures Flask app for deployment.



sss982052 / income-prediction
Contains: Image | Last pushed: a month ago
Inactive ☆ 0 ⭳ 3 Public

7. ***CI/CD Workflows:***

   AWS Deployment: Automates deployment to AWS infrastructure.
   Docker Image Push: Pushes Docker image to Amazon ECR.
   EC2 Deployment: Deploys the application on Amazon EC2 instances.

Execution Flow

1. ***Data Retrieval (retrieve_data.py):***

   Connects to MongoDB.
   Queries data based on parameters.
   Outputs cleaned dataset.

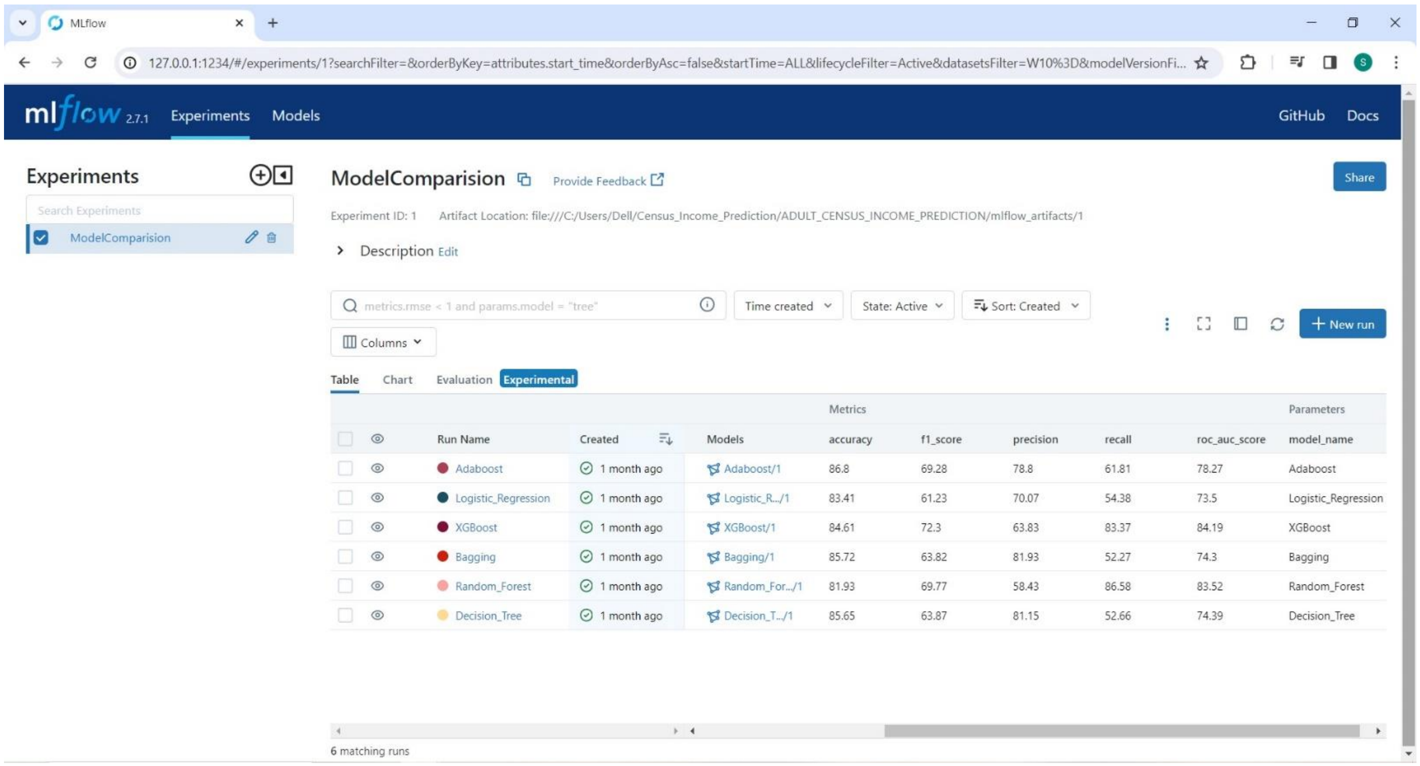
2. ***Data Transformation (data_transformation.py):***

   Cleans and preprocesses the dataset.
   Applies feature engineering.
   Encodes categorical variables.
   Outputs transformed data.

3. ***Model Training (model_training.py):***

   Tests multiple models and hyperparameters.
   Evaluates models using various metrics.

Saves the best-performing model.
Logs metrics and parameters using MLflow.



127.0.0.1:1234/#/experiments/1?searchFilter=&orderByKey=attributes.start_time&orderByAsc=false&startTime=ALL&lifecycleFilter=Active&datasetsFilter=W10%3D&modelVersionFi...

mlflow 2.7.1 Experiments Models GitHub Docs

Experiments

ModelComparision

ModelComparision Provide Feedback

Experiment ID: 1 Artifact Location: file:///C:/Users/Dell/Census_Income_Prediction/ADULT_CENSUS_INCOME_PREDICTION/mlflow_artifacts/1

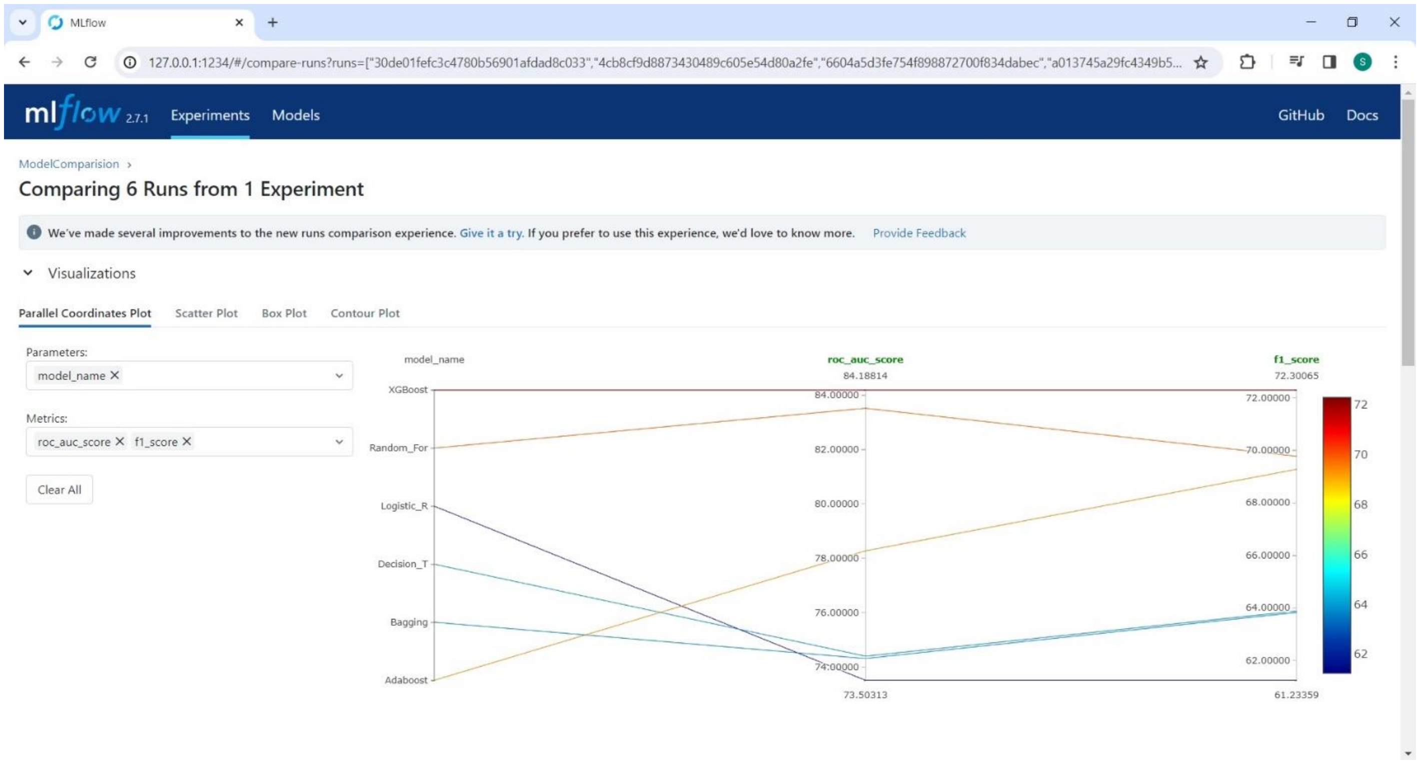
Description Edit

Time created · State: Active · Sort: Created · Columns · New run

Table Chart Evaluation Experimental

Run Name	Created	Models	accuracy	f1_score	precision	recall	roc_auc_score	model_name
Adaboost	1 month ago	Adaboost/1	86.8	69.28	78.8	61.81	78.27	Adaboost
Logistic_Regression	1 month ago	Logistic_R.../1	83.41	61.23	70.07	54.38	73.5	Logistic_Regression
XGBoost	1 month ago	XGBoost/1	84.61	72.3	63.83	83.37	84.19	XGBoost
Bagging	1 month ago	Bagging/1	85.72	63.82	81.93	52.27	74.3	Bagging
Random_Forest	1 month ago	Random_For.../1	81.93	69.77	58.43	86.58	83.52	Random_Forest
Decision_Tree	1 month ago	Decision_T.../1	85.65	63.87	81.15	52.66	74.39	Decision_Tree

6 matching runs



127.0.0.1:1234/#/compare-runs?runs=["30de01fefc3c4780b56901afdad8c033","4cb8cf9d8873430489c605e54d80a2fe","6604a5d3fe754f898872700f834dabec","a013745a29fc4349b5...

mlflow 2.7.1 Experiments Models GitHub Docs

ModelComparision ›

Comparing 6 Runs from 1 Experiment

We've made several improvements to the new runs comparison experience. Give it a try. If you prefer to use this experience, we'd love to know more. Provide Feedback

Visualizations

Parallel Coordinates Plot · Scatter Plot · Box Plot · Contour Plot

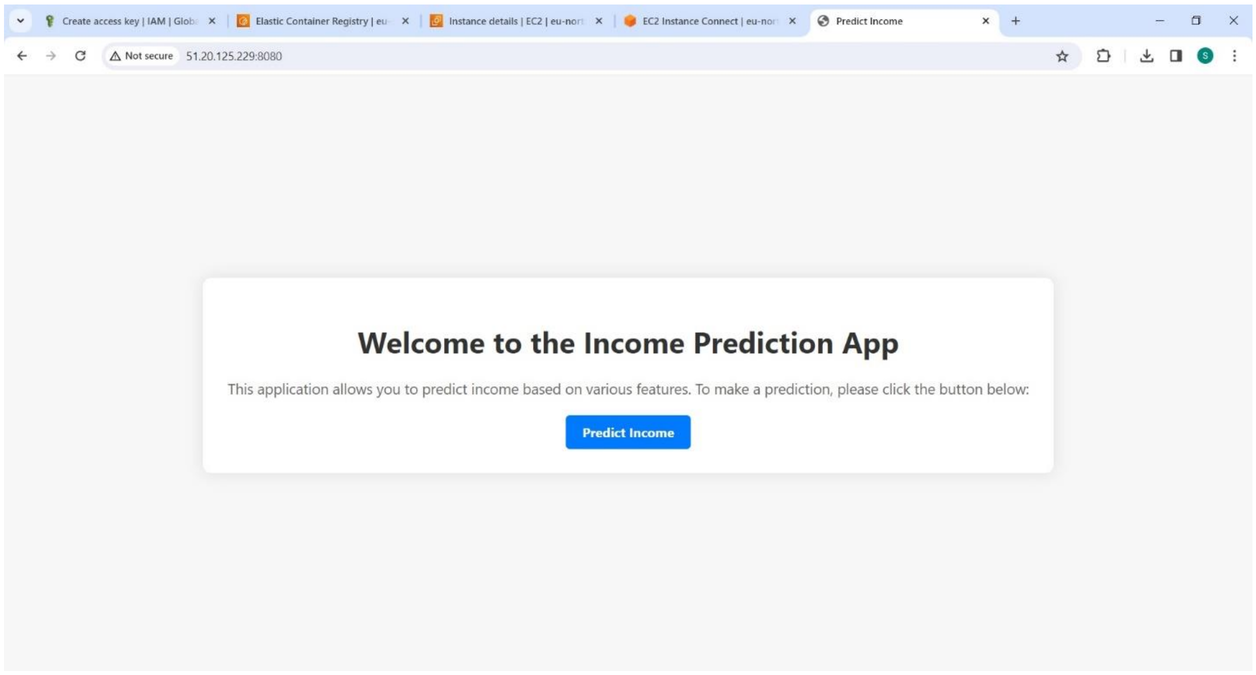
Parameters: model_name

Metrics: roc_auc_score, f1_score

Clear All

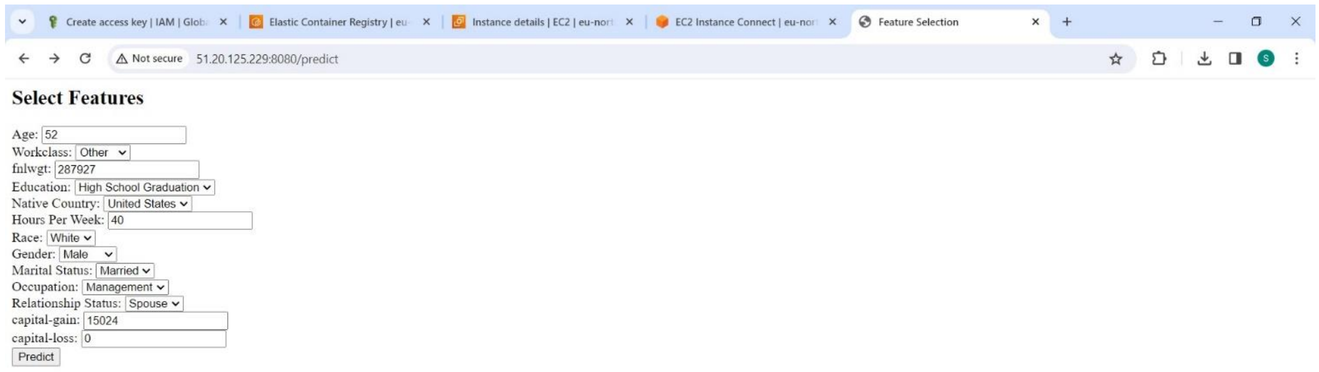
4. *Web Application (app.py):*

Renders home page and form for data input.
Processes user input for prediction.
Sends data to the model for prediction.
Displays prediction result on the web page.
Logs errors and messages.


Create access key | IAM | Glob
Elastic Container Registry | eu-
Instance details | EC2 | eu-nort
EC2 Instance Connect | eu-nor
Predict Income
Not secure
51.20.125.229:8080

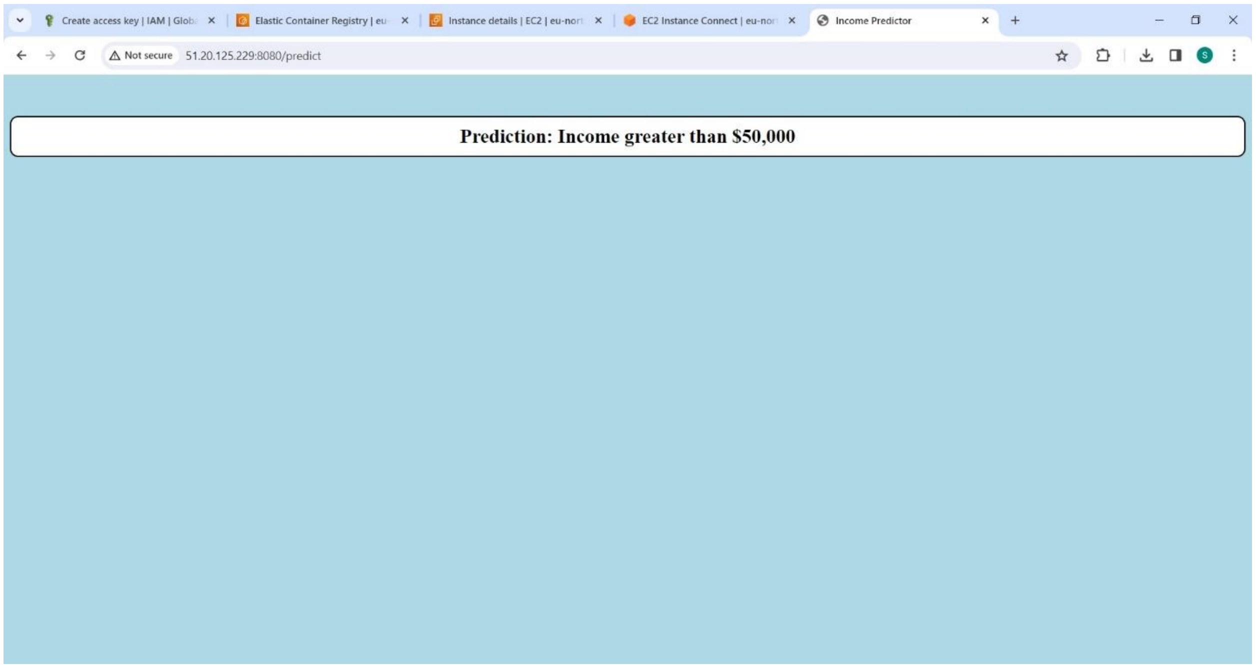
Welcome to the Income Prediction App

This application allows you to predict income based on various features. To make a prediction, please click the button below:


Create access key | IAM | Glob
Elastic Container Registry | eu
Instance details | EC2 | eu-nort
EC2 Instance Connect | eu-nor
Feature Selection
Not secure
51.20.125.229:8080/predict

Select Features

Age: 52
Workclass: Other
fnlwgt: 287927
Education: High School Graduation
Native Country: United States
Hours Per Week: 40
Race: White
Gender: Male
Marital Status: Married
Occupation: Management
Relationship Status: Spouse
capital-gain: 15024
capital-loss: 0


Create access key | IAM | Glob
Elastic Container Registry | eu
Instance details | EC2 | eu-nort
EC2 Instance Connect | eu-nor
Income Predictor
Not secure
51.20.125.229:8080/predict

Prediction: Income greater than $50,000

5. ***MLflow:***

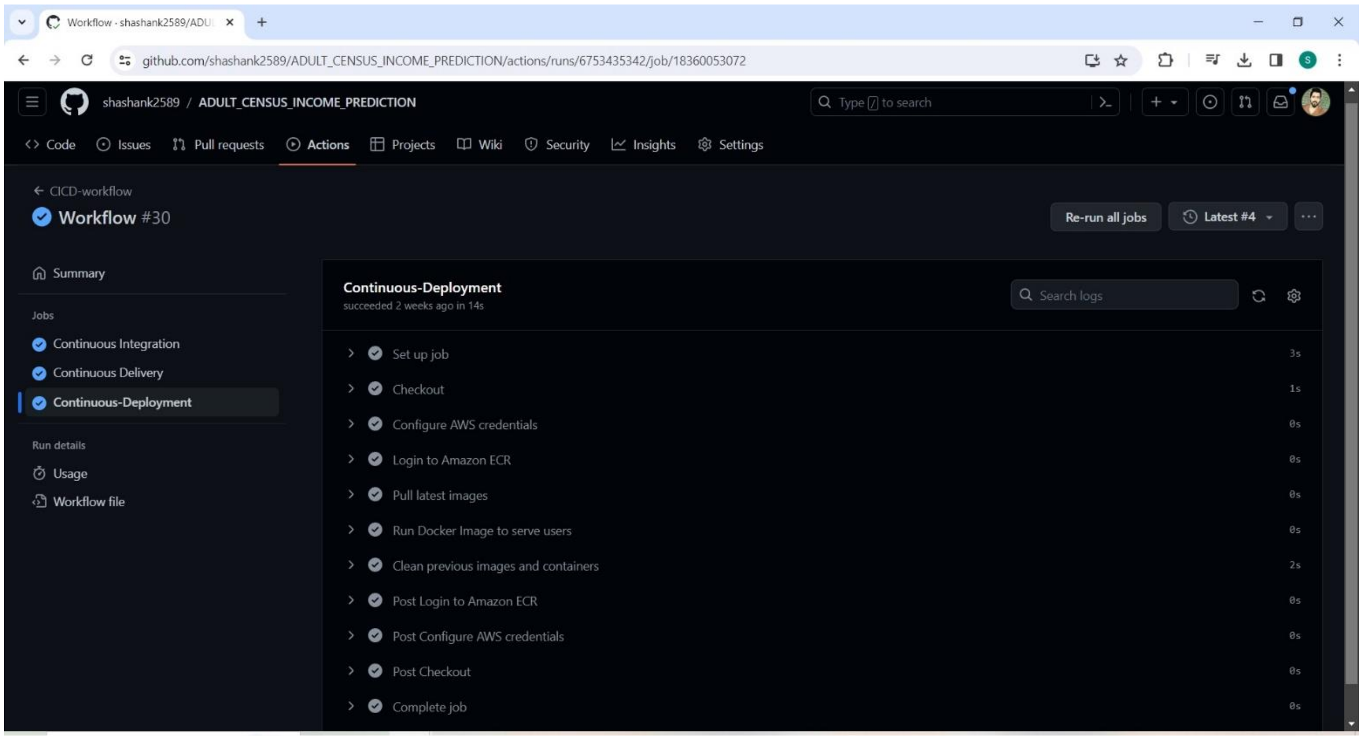
   Artifacts Folder (mlflow_artifacts):
   Stores MLflow artifacts: models, metrics, parameters.
   MLflow Server:
   Runs locally using mlflow server.
   Uses SQLite as a backend store.

6. ***Continuous Integration:***

   GitHub Actions:
   Triggers CI workflows on GitHub push events.
   Builds Docker images, pushes to ECR.

7. ***Continuous Deployment:***

   AWS EC2:
   Pulls the latest Docker image from ECR.
   Deploys the Flask application.


github.com/shashank2589/ADULT_CENSUS_INCOME_PREDICTION/actions/runs/6753435342/job/18360053072

shashank2589 / ADULT_CENSUS_INCOME_PREDICTION

CICD-workflow

Workflow #30

Jobs
- Continuous Integration
- Continuous Delivery
- Continuous-Deployment

Continuous-Deployment
succeeded 2 weeks ago in 14s

```
Set up job                                3s
Checkout                                  1s
Configure AWS credentials                 0s
Login to Amazon ECR                       0s
Pull latest images                        0s
Run Docker Image to serve users           0s
Clean previous images and containers      2s
Post Login to Amazon ECR                  0s
Post Configure AWS credentials            0s
Post Checkout                             0s
Complete job                              0s
```

This low-level design provides a detailed perspective on the functionalities and interactions of each component in the Income Prediction Model. It focuses on the specifics of data processing, model training, web application development, and deployment workflows.