

Architecture documentation

Adult census income Prediction

Document Version: 1.0

Date: 25-Sept-2023

Author: Abhishek Pandit

Contents

1.Introduction

2.Architecture

3.Architecture Design

1.User Interface (Streamlit)

2.2. Data Ingestion

3.3. Data Transformation

4.4. Model Training

5.5. Prediction Pipeline

4.Model training/validation workflow

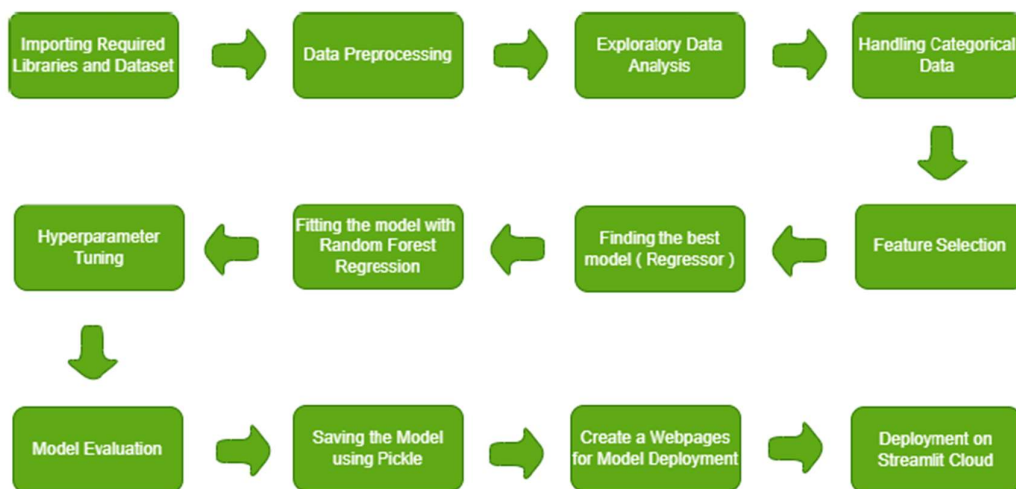
5.User I/O workflow

6.Conclusion

Introduction

The architecture of the Machine Learning Income Prediction App is designed to ensure scalability, maintainability, and reliability. This section provides an overview of the application's architecture and the key components that make up the system.

1. Architecture



2. Architecture Design

The application follows a modular and layered architecture, comprising the following key components:

1. User Interface (Streamlit)

- **Description:** The user interface is developed using Streamlit, a web application framework for Python.
- **Responsibilities:** It handles user input, displays prediction results, and interacts with other components.

- **Key Features:** User-friendly interface, input validation, and seamless integration with the prediction pipeline.

2. Data Ingestion ([data_ingestion.py](#))

- **Description:** Responsible for data loading and initial preprocessing.
- **Responsibilities:** Reads data from external sources, ensures data quality, and prepares data for further processing.
- **Key Features:** Data validation, error handling, and data storage management.

3. Data Transformation ([data_transformation.py](#))

- **Description:** Manages data transformation and preprocessing.
- **Responsibilities:** Includes pipelines for feature scaling, encoding, and imputing missing values.
- **Key Features:** Flexible data transformation for different input data types and consistency checks.

4. Model Training ([model_trainer.py](#))

- **Description:** Handles machine learning model training and evaluation.
- **Responsibilities:** Trains multiple models, evaluates their performance, and selects the best-performing model.
- **Key Features:** Grid search for hyperparameter tuning and model evaluation.

5. Prediction Pipeline (prediction_pipeline.py)

- **Description:** Handles user input and data preprocessing for predictions.
- **Responsibilities:** Utilizes the trained machine learning model to make income predictions.
- **Key Features:** Real-time prediction and result presentation.

6. Artifacts Folder (artifacts/)

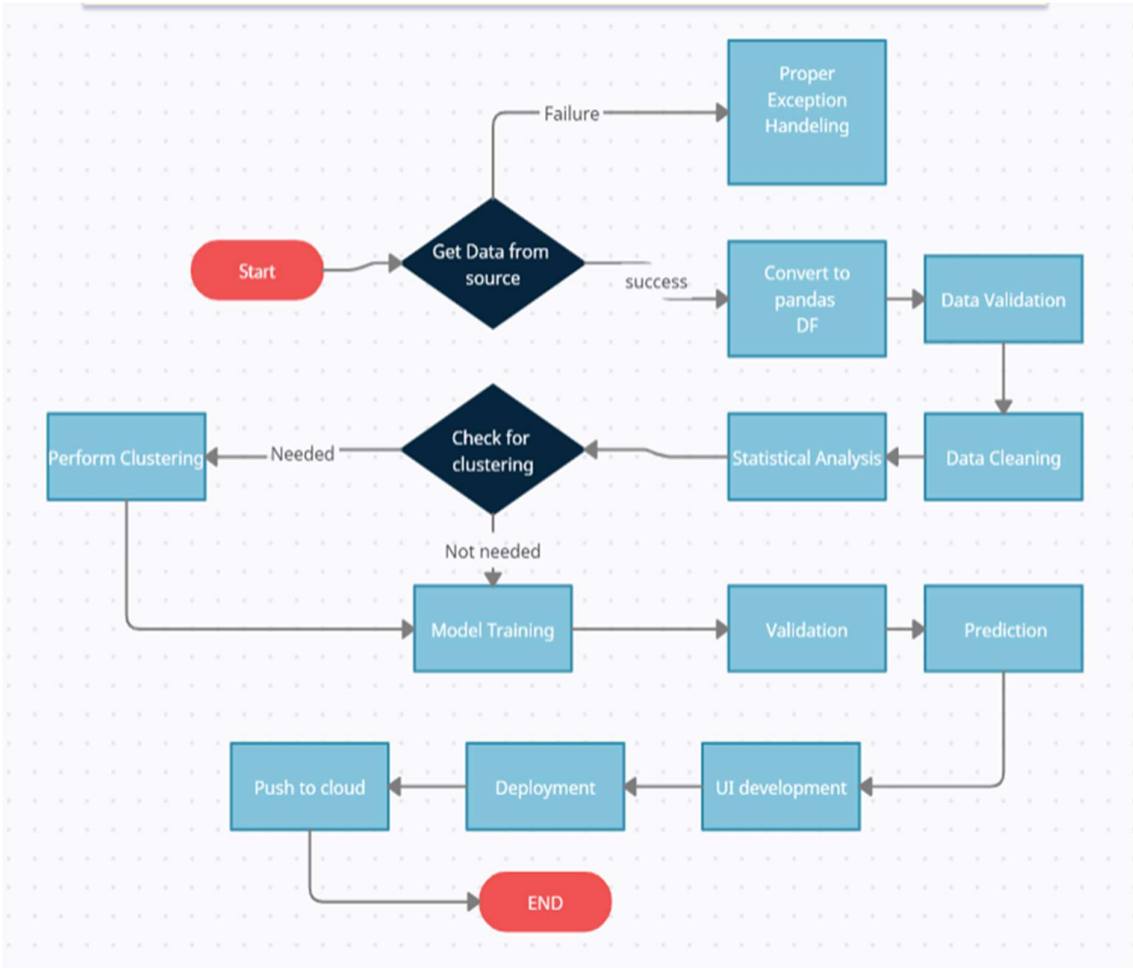
- **Description:** A folder for storing model and preprocessor objects.
- **Responsibilities:** Safely stores the trained machine learning model and preprocessing objects.
- **Key Features:** Controlled access and version management of artifacts.

Technology Stack

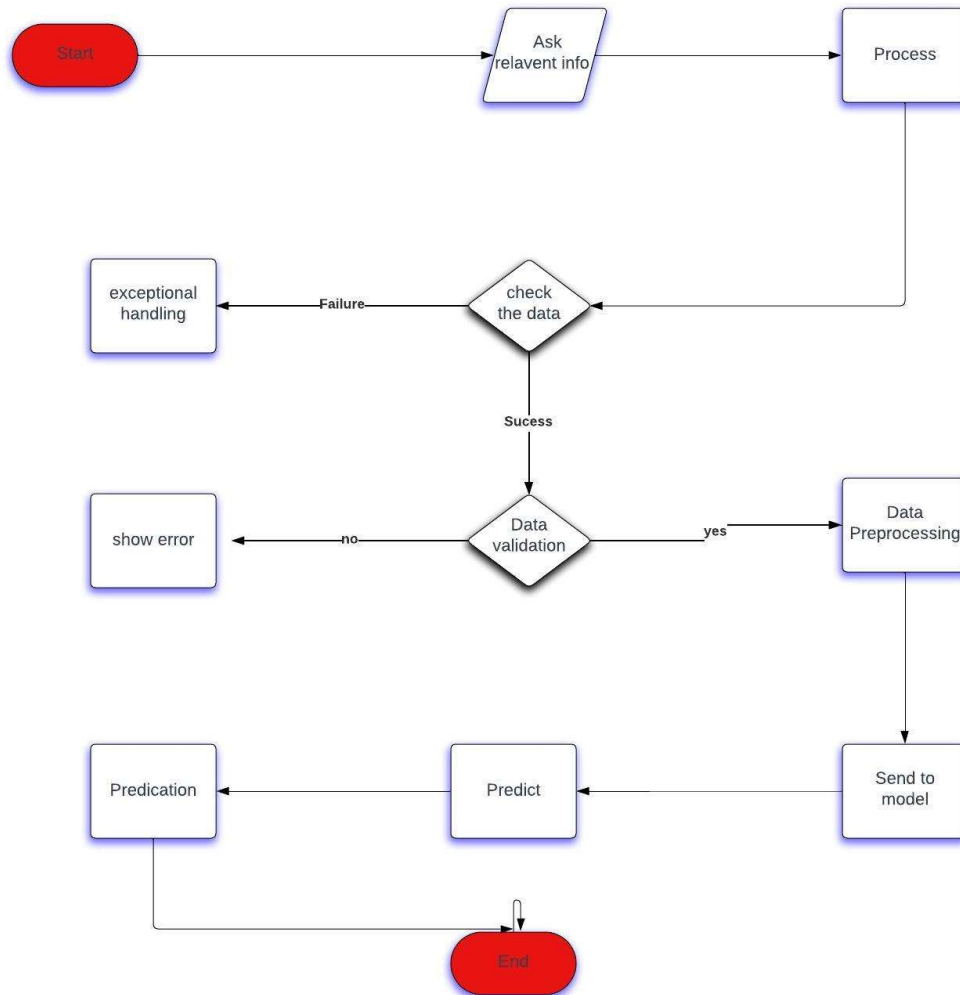
The application is built using the following technology stack:

- **Programming Language:** Python
- **Libraries:** pandas, scikit-learn, Streamlit, numpy

3. Model training/validation workflow



4. User I/O workflow



5. Conclusion

The architecture of the Machine Learning Income Prediction App is designed to be modular, scalable, and maintainable. It comprises distinct components responsible for different aspects of the application, ensuring efficient data processing and accurate predictions.