**Adult Censes Income Prediction**

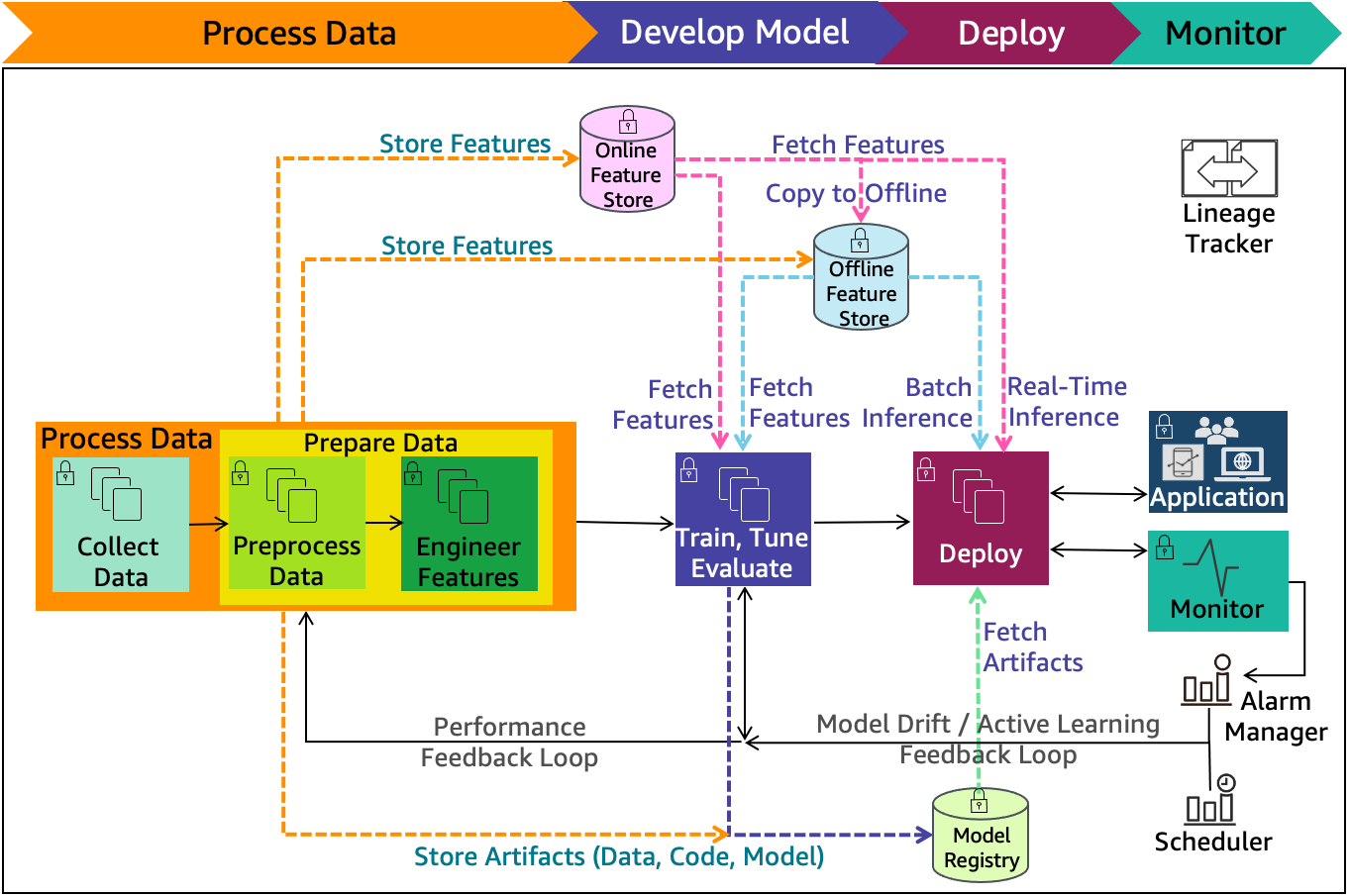


**SRIPHANI**

**Problem Statement**

To build a classification methodology to determine whether a person makes over 50K per year.

**Architecture**



**Data Description**

The client will send data in multiple sets of files in batches at a given location. The data has been extracted from the census bureau.

The data contains 32561 instances with the following attributes:

**Features:**

1. **age**: continuous. It denotes the age of the person.
2. **workclass**: It denotes the working class of the person. Sample values: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
3. **fnlwgt**: continuous.
4. **education**: It denotes the educational qualification of the person. Sample values: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
5. **education**-num: continuous. It denotes the quantitative values with reference to education.
6. **marital-status**: It denotes the marital status of the person. Sample values: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
7. **occupation**: It denotes the occupation of a person. Sample values: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
8. **relationship**: It denotes the people present in the family. Sample values: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
9. **race**: It denotes the person’s origins. Sample values: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
10. **sex**: It denotes the person's gender. Sample values: Female, Male.
11. **capital-gain**: continuous. It denotes the monitory gains by the person.
12. **capital-loss:** continuous. It denotes the monitory loss by the person.
13. **hours-per-week:** continuous. It denotes the number of working hours per week by the person.
14. **native-country:** It denotes the country to which the person belongs. Sample values: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

**Target Label:**

Whether a person earns more or less than 50000 dollars per year.

1. Income: >50K, <=50K.

Apart from training files, we also require a "schema" file from the client, which contains all the relevant information about the training files such as:

Name of the files, Length of Date value in FileName, Length of Time value in FileName, Number of Columns, Name of the Columns, and their datatype.

**Model Training**

1) **Data Export from Db** - The data in a stored database is exported as a CSV file to be used for model training.

2) **Data Preprocessing**

1. Drop the columns not required for prediction.
2. Remove the unwanted spaces in data.
3. For this dataset, the null values were replaced with ‘?’ in the client data. Those ‘?’ have been replaced with NaN values.
4. Check for null values in the columns. If present, impute the null values using the categorical imputer.
5. Replace and encode the categorical values with numeric values.
6. Scale the numeric values using the standard scaler.
7. Handle the imbalanced dataset using oversampling.

3) **Clustering -** KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using "KneeLocator" function. The idea behind clustering is to implement different algorithms

The Kmeans model is trained over preprocessed data, and the model is saved for further use in prediction.

4) **Model Selection –** After the clusters have been created, we find the best model for each cluster. We are using two algorithms, "Random Forest" and "XGBoost". For each cluster, both the algorithms are passed with the best parameters derived from GridSearch. We calculate the AUC scores for both models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction.

**Prediction Data Description**

The Client will send the data in multiple sets of files in batches at a given location. Data will contain the annual income of various persons.

Apart from prediction files, we also require a "schema" file from the client, which contains all the relevant information about the training files such as:

Name of the files, Length of Date value in FileName, Length of Time value in FileName, Number of Columns, Name of the Columns and their datatype.

**Prediction**

1) Data Export from Db - The data in the stored database is exported as a CSV file to be used for prediction.

2) Data Preprocessing :

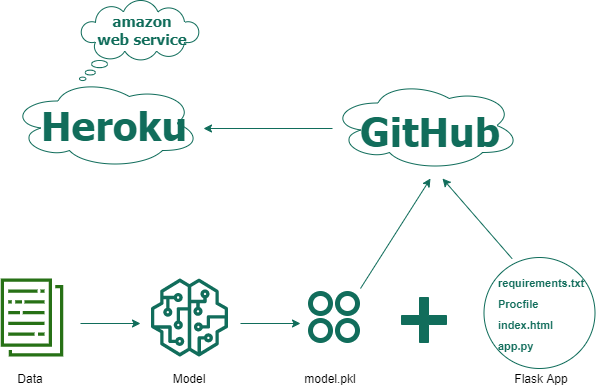
1. Drop the columns not required for prediction.
2. Remove the unwanted spaces in data.
3. For this dataset, the null values were replaced with ‘?’ in the client data. Those ‘?’ have been replaced with NaN values.
4. Check for null values in the columns. If present, impute the null values using the categorical imputer.
5. Replace and encode the categorical values with numeric values.
6. Scale the numeric values using the standard scaler.
7. Handle the imbalanced dataset using oversampling

3) Clustering - KMeans model created during training is loaded, and clusters for the preprocessed prediction data is predicted.

4) Prediction - Based on the cluster number, the respective model is loaded and is used to predict the data for that cluster.

5) Once the prediction is made for all the clusters, the predictions along with the Wafer names are saved in a CSV file at a given location, and the location is returned to the client.

**Deployment**



**requirements.txt** file consists of all the packages that you need to deploy the app in the cloud.

**main.py** is the entry point of our application, where the flask server starts. Here we will have two routes, one for training and the other for prediction.

**Procfile**:- It contains the entry point of the app.