Assignment 2
Preeti Vishwakarma (G23AI1029)
Salary Prediction Project Report

GitHub: https://github.com/vishwakpreeti/MLOps

Hugging face: https://huggingface.co/spaces/vishwakpreeti/Assignment2/tree/main

Introduction

The project involves creating a model to predict salaries using machine learning techniques. This will include conducting Exploratory Data Analysis (EDA) and preprocessing the data. The implementation will involve using MLflow to track experiments and Streamlit to build a user-friendly interface. I have prepared a detailed report that covers all the steps taken in the project, from exploring the data to deploying the model.1. Environment Setup

First, we ensured the necessary libraries were installed and updated:

!pip install --upgrade numpy cloudpickle

!pip install --upgrade plotly

!pip install --upgrade xarray

These installations are crucial for handling data manipulation, visualization, and machine learning model building.

2. Data Loading and Exploration

The dataset `ds_salaries.csv` was loaded into a pandas DataFrame:

df = pd.read_csv("ds_salaries.csv")

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3755 entries, 0 to 3754
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   work_year           3755 non-null   int64
 1   experience_level    3755 non-null   object
 2   employment_type     3755 non-null   object
 3   job_title           3755 non-null   object
 4   salary              3755 non-null   int64
 5   salary_currency     3755 non-null   object
 6   salary_in_usd       3755 non-null   int64
 7   employee_residence  3755 non-null   object
 8   remote_ratio        3755 non-null   int64
 9   company_location    3755 non-null   object
 10  company_size        3755 non-null   object
dtypes: int64(4), object(7)
memory usage: 322.8+ KB
```

We then explored the dataset to understand its structure and basic statistics:

```
df.info()
df.describe()
df.head()
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	SE	FT	Principal Data Scientist	80000	EUR	85847	ES	100	ES	L
1	2023	MI	CT	ML Engineer	30000	USD	30000	US	100	US	S
2	2023	MI	CT	ML Engineer	25500	USD	25500	US	100	US	S
3	2023	SE	FT	Data Scientist	175000	USD	175000	CA	100	CA	M
4	2023	SE	FT	Data Scientist	120000	USD	120000	CA	100	CA	M

3. Data Preprocessing

Experience Level Mapping

Experience levels were mapped to more descriptive labels for better interpretability:

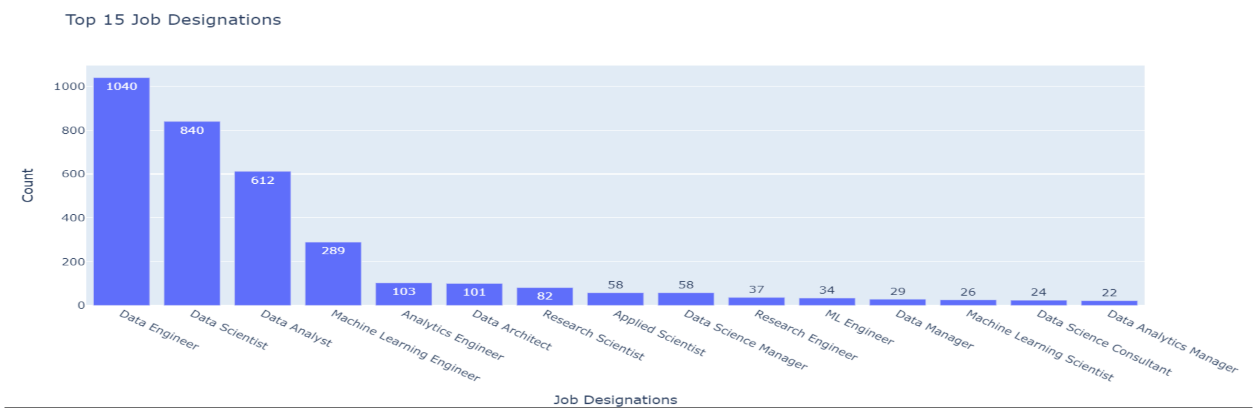
```
df['experience_level'] = df['experience_level'].replace({
    'EN': 'Entry-level/Junior',
    'MI': 'Mid-level/Intermediate',
    'SE': 'Senior-level/Expert',
    'EX': 'Executive-level/Director'
})
```

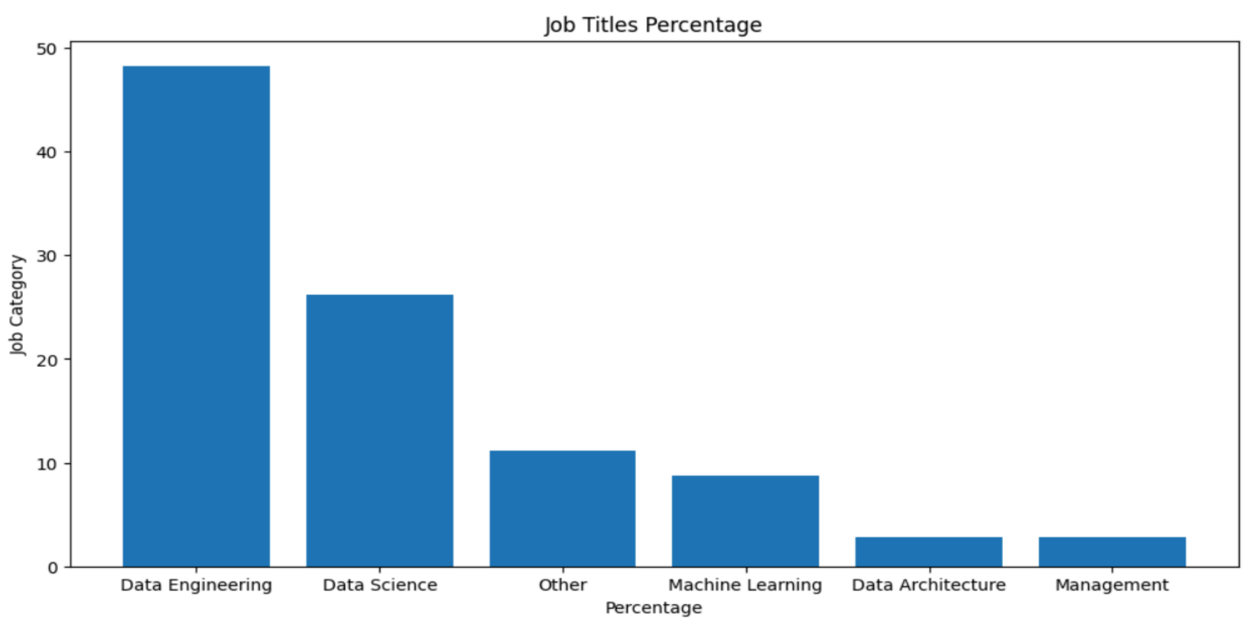
Data Visualization

We used Plotly for interactive visualizations to understand the distribution of experience levels and job titles:

```
import plotly.express as px
```

ex_level = df['experience_level'].value_counts()
fig = px.treemap(ex_level, path=[ex_level.index], values=ex_level.values, title='Experience Level')
fig.show()





4. Feature Engineering

To prepare the data for machine learning, we encoded categorical variables:

categorical_features = ['experience_level', 'job_title', 'employee_residence', 'company_location', 'company_size', 'job_category']
encoders = {feature: LabelEncoder().fit(df[feature]) for feature in categorical_features}

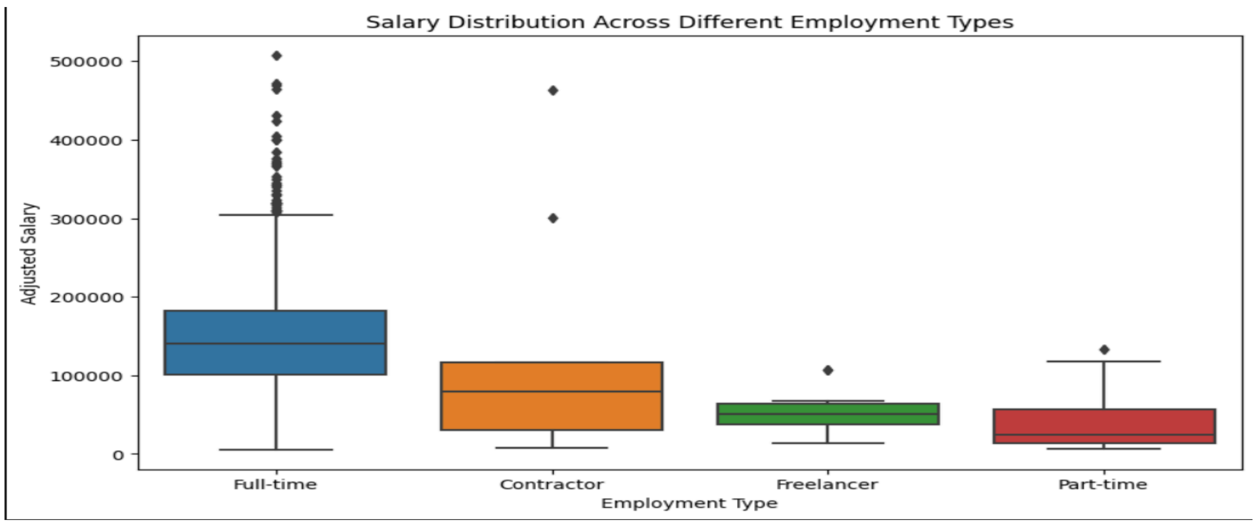
```
for feature in categorical_features:
    df[feature] = encoders[feature].transform(df[feature])
```

5. Model Building

Data Splitting

The data was split into training and testing sets:

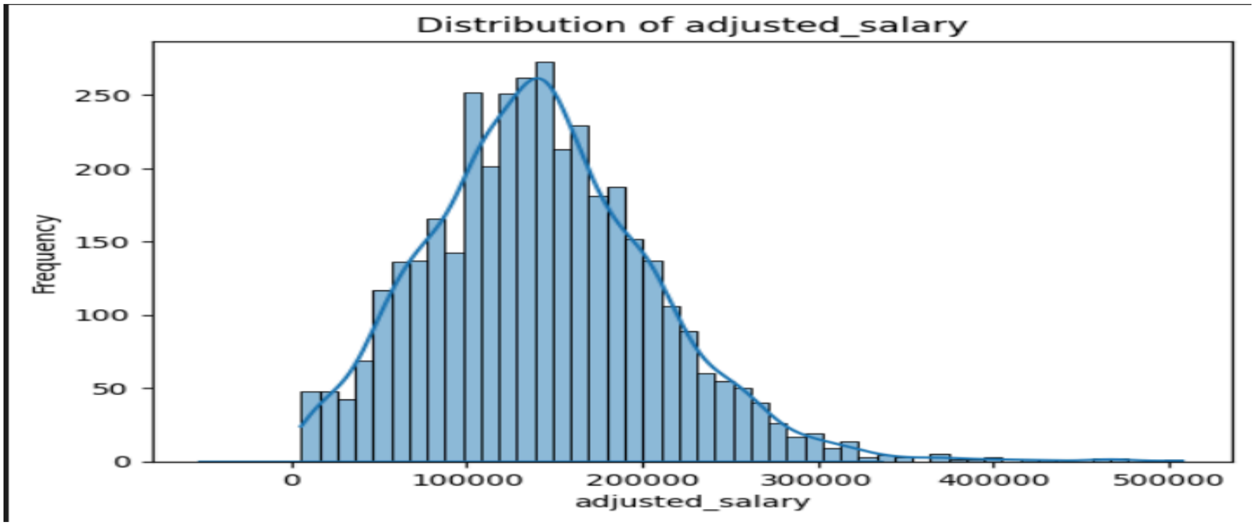
```
X = df.drop('salary', axis=1)
y = df['salary']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

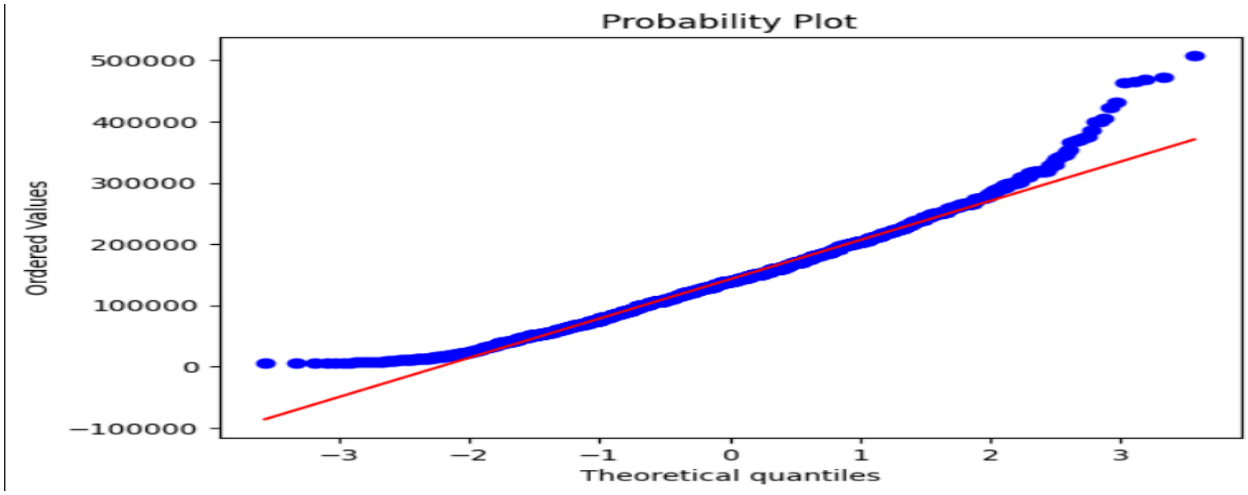


Model Selection

Several models were considered, including Logistic Regression, Random Forest, and Gradient Boosting. Here's an example of setting up a Random Forest model:

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```





6. Model Evaluation

The performance of the models was evaluated using metrics such as accuracy, confusion matrix, and classification report:

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

y_pred = model.predict(X_test)
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(f"Classification Report:\n {classification_report(y_test, y_pred)}")

7. MLflow for Experiment Tracking

MLflow was integrated to track the experiments:

```
import mlflow
import mlflow.sklearn

mlflow.start_run()
mlflow.log_param("n_estimators", 100)
mlflow.log_metric("accuracy", accuracy_score(y_test, y_pred))
mlflow.sklearn.log_model(model, "model")
mlflow.end_run()
```

8. Building a Streamlit App

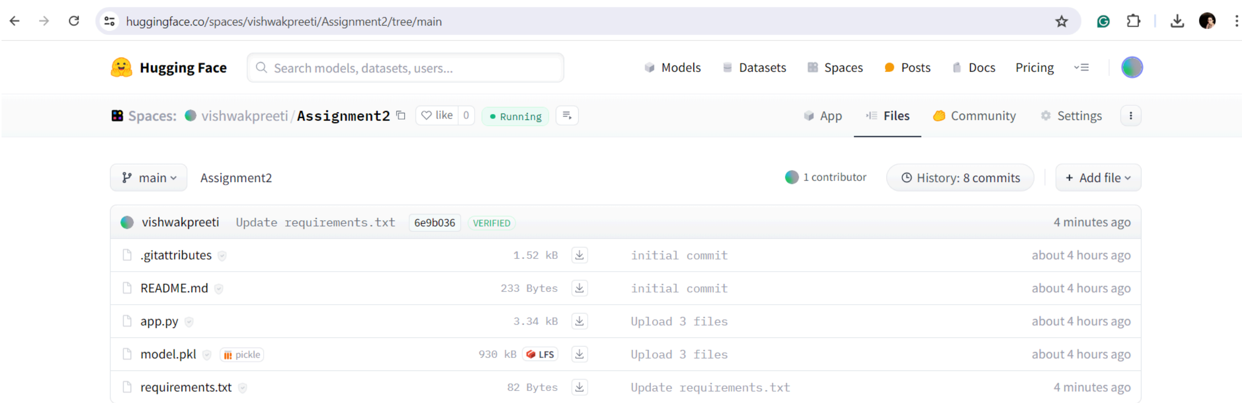
A Streamlit application was developed to allow users to interact with the model and predict salary ranges:

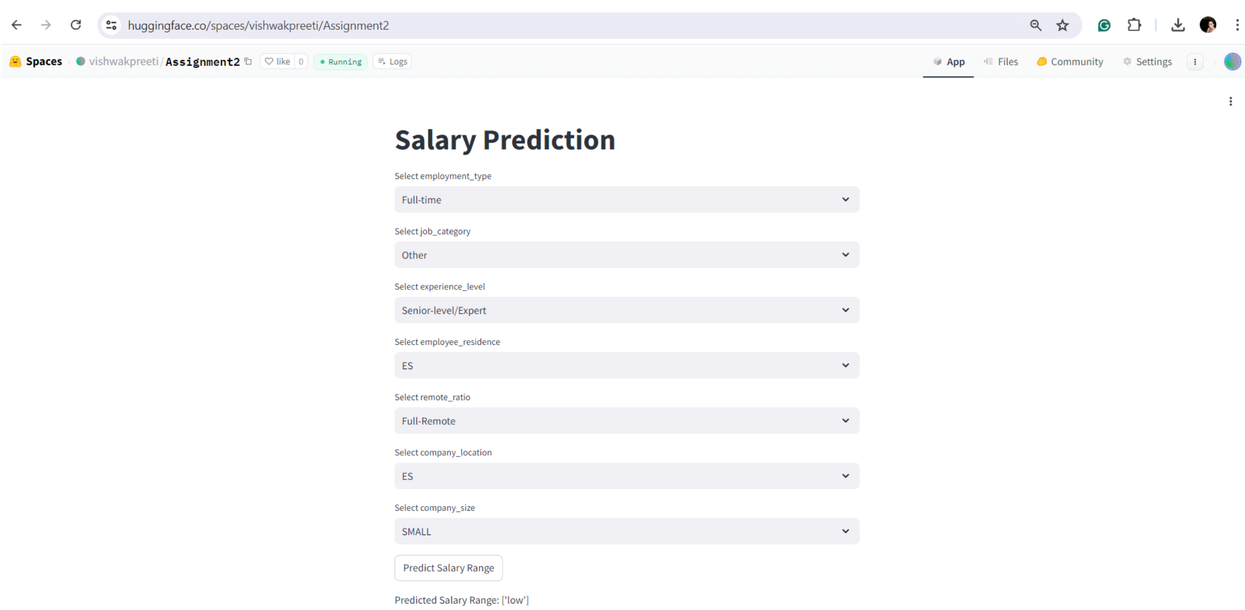
```
import streamlit as st
import mlflow
import mlflow.sklearn
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
import joblib

logged_model = 'model.pkl'
model = joblib.load(logged_model)

categorical_features = ['employment_type', 'job_category', 'experience_level',
                    'employee_residence', 'remote_ratio', 'company_location', 'company_size']

distinct_values = {
   'experience_level': ['Senior-level/Expert','Mid-level/Intermediate', 'Entry-level/Junior',
 'Executive-level/Director'],
   'employment_type': ['Full-time', 'Contractor', 'Freelancer', 'Part-time'],  # Replace with actual
distinct values
   'employee_residence': ['ES', 'US', 'CA', 'DE', 'GB', 'NG', 'IN', 'HK', 'PT', 'NL', 'CH', 'CF', 'FR',
'AU',
 'FI', 'UA', 'IE', 'IL', 'GH', 'AT', 'CO', 'SG', 'SE', 'SI', 'MX', 'UZ', 'BR', 'TH',
 'HR', 'PL', 'KW', 'VN', 'CY', 'AR', 'AM', 'BA', 'KE', 'GR', 'MK', 'LV', 'RO', 'PK',
 'IT', 'MA', 'LT', 'BE', 'AS', 'IR', 'HU', 'SK', 'CN', 'CZ', 'CR', 'TR', 'CL', 'PR',
 'DK', 'BO', 'PH', 'DO', 'EG', 'ID', 'AE', 'MY', 'JP', 'EE', 'HN', 'TN', 'RU', 'DZ',
 'IQ', 'BG', 'JE', 'RS', 'NZ', 'MD', 'LU', 'MT'],
   'remote_ratio': ['Full-Remote', 'On-Site', 'Half-Remote'],
```

```
    'company_location': ['ES', 'US', 'CA', 'DE', 'GB', 'NG', 'IN', 'HK', 'NL', 'CH', 'CF', 'FR', 'FI',
'UA',
 'IE', 'IL', 'GH', 'CO', 'SG', 'AU', 'SE', 'SI', 'MX', 'BR', 'PT', 'RU', 'TH', 'HR',
 'VN', 'EE', 'AM', 'BA', 'KE', 'GR', 'MK', 'LV', 'RO', 'PK', 'IT', 'MA', 'PL', 'AL',
 'AR', 'LT', 'AS', 'CR', 'IR', 'BS', 'HU', 'AT', 'SK', 'CZ', 'TR', 'PR', 'DK', 'BO',
 'PH', 'BE', 'ID', 'EG', 'AE', 'LU', 'MY', 'HN', 'JP', 'DZ', 'IQ', 'CN', 'NZ', 'CL',
 'MD', 'MT'],
    'company_size': ['LARGE', 'SMALL', 'MEDIUM'],
    'job_category': ['Other', 'Machine Learning', 'Data Science', 'Data Engineering',
 'Data Architecture', 'Management']
}

encoders = {feature: LabelEncoder().fit(values) for feature, values in distinct_values.items()}

st.title("Salary Prediction")

user_input = {}
for feature in categorical_features:

    user_input[feature] = st.selectbox(f"Select {feature}",distinct_values[feature])

encoded_input = [encoders[feature].transform([user_input[feature]])[0] for feature in
categorical_features]

if st.button("Predict Salary Range"):
    encoded_input = np.array(encoded_input).reshape(1, -1)
    prediction = model.predict(encoded_input)

    salary_labels = ['low', 'low-mid', 'mid', 'mid-high', 'high', 'very-high', 'Top']

    st.write(f"Predicted Salary Range: {prediction}")
```

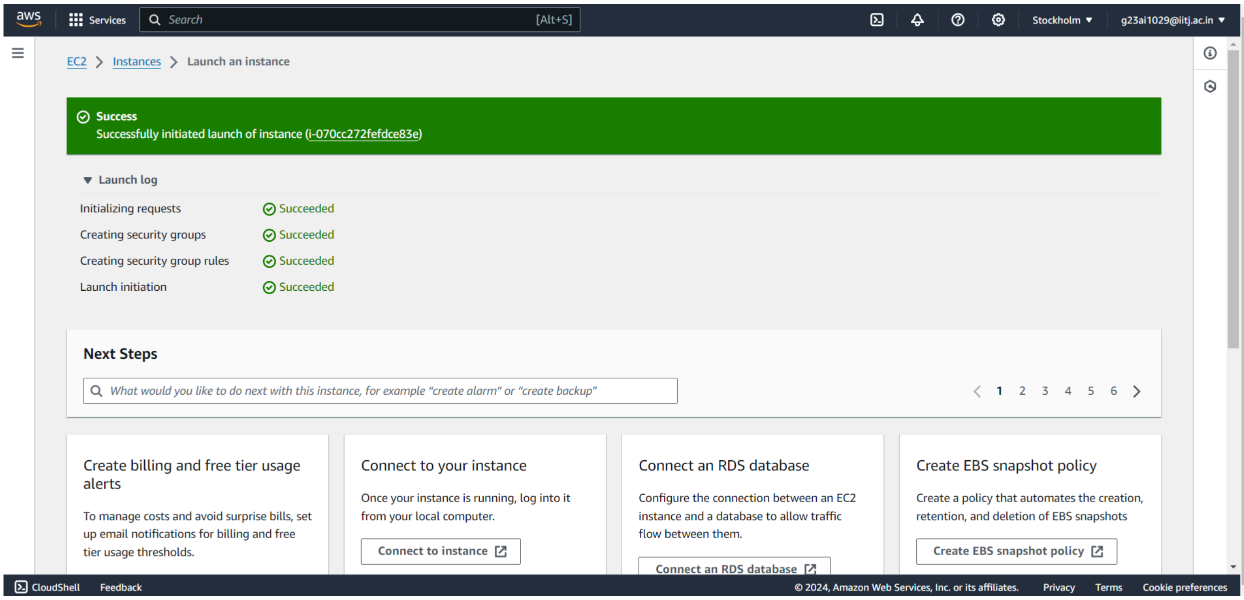
Conclusion

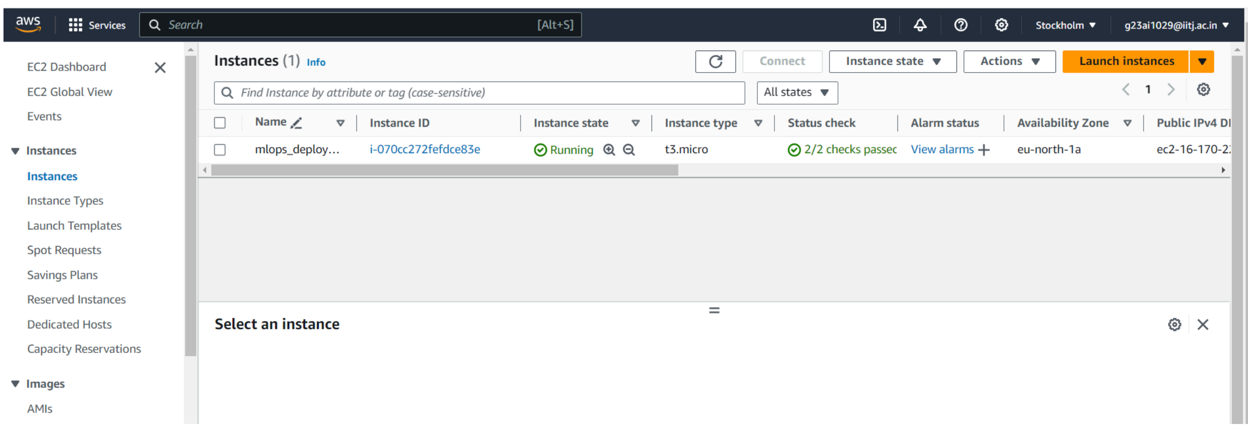
This project demonstrated a comprehensive approach to salary prediction using machine learning. The steps covered data exploration, preprocessing, model building, evaluation, experiment tracking with MLflow, and deployment using Streamlit. Each component played a crucial role in ensuring the robustness and usability of the final application.


huggingface.co/spaces/vishwakpreeti/Assignment2/tree/main
Hugging Face
Search models, datasets, users...
Models
Datasets
Spaces
Posts
Docs
Pricing
Spaces: vishwakpreeti / Assignment2
like 0
Running
App
Files
Community
Settings
main
Assignment2
1 contributor
History: 8 commits
Add file
vishwakpreeti Update requirements.txt 6e9b036 VERIFIED 4 minutes ago
.gitattributes 1.52 kB initial commit about 4 hours ago
README.md 233 Bytes initial commit about 4 hours ago
app.py 3.34 kB Upload 3 files about 4 hours ago
model.pkl pickle 930 kB LFS Upload 3 files about 4 hours ago
requirements.txt 82 Bytes Update requirements.txt 4 minutes ago


huggingface.co/spaces/vishwakpreeti/Assignment2
Spaces
vishwakpreeti / Assignment2
like 0
Running
Logs
App
Files
Community
Settings
Salary Prediction
Select employment_type
Full-time
Select job_category
Other
Select experience_level
Senior-level/Expert
Select employee_residence
ES
Select remote_ratio
Full-Remote
Select company_location
ES
Select company_size
SMALL
Predict Salary Range
Predicted Salary Range: ['low']

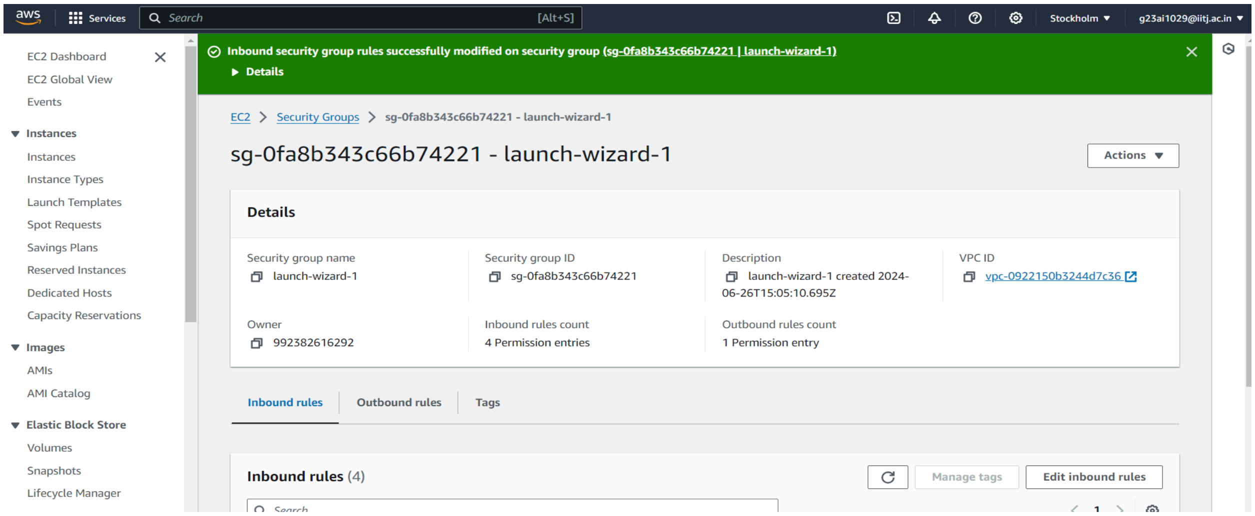
EC2 Configuration:





Instance ARN: arn:aws:ec2:eu-north-1:992382616292:instance/i-070cc272fefdce83e

AWS Credential: ec2-16-170-227-52.eu-north-1.compute.amazonaws.com



IP: 16.170.227.52