Name: Manisha Barse
Internship Batch: LISUM30
Submission Date: February 28, 2024
Submitted to: Data Glacier

Week4: Deployment on Flask

Task:

1. Select any toy data (simple data).
2. Save the model
3. Deploy the model on flask (web app)
4. Create pdf document (Name, Batch code, Submission date, Submitted to) which should contain snapshot of each step of deployment)
5. Upload the document to Github
6. Submit the URL of the uploaded document.

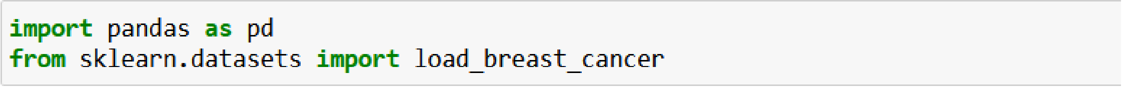
Breast Cancer Model Deployment

Introduction:

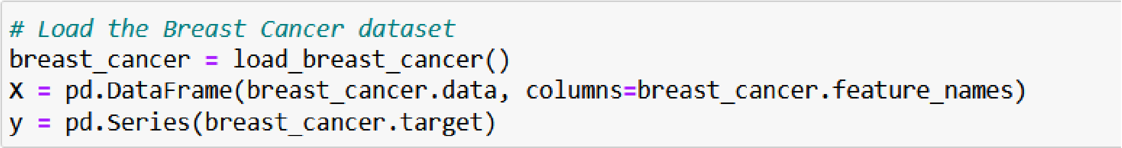
This script demonstrates the process of building, training, and deploying a machine learning model using the Breast Cancer dataset.
The task involves preprocessing the data, training a Support Vector Machine (SVM) model, deploying the model on Flask, and creating a PDF report documenting the entire process.

Data Preprocessing:

- Loaded the Breast Cancer dataset.

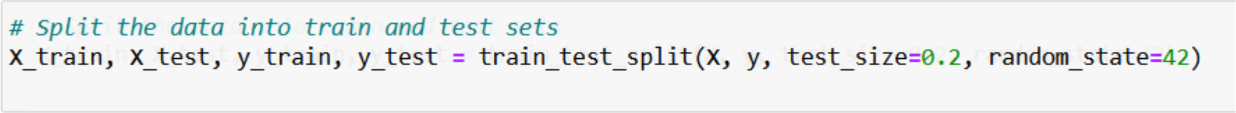


```
import pandas as pd
from sklearn.datasets import load_breast_cancer
```



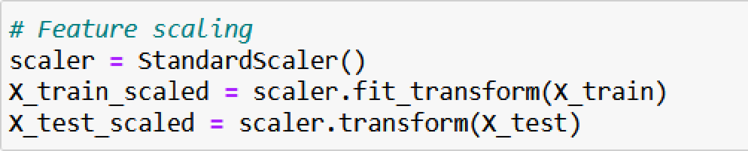
```
# Load the Breast Cancer dataset
breast_cancer = load_breast_cancer()
X = pd.DataFrame(breast_cancer.data, columns=breast_cancer.feature_names)
y = pd.Series(breast_cancer.target)
```

- Split the dataset into train and test sets.



```
# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

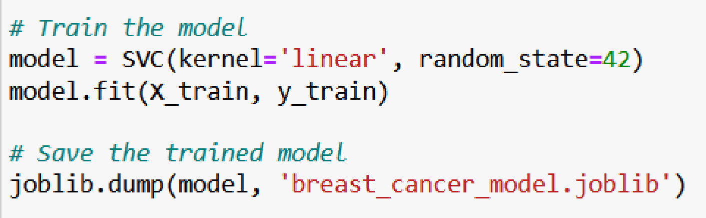
- Scaled the features using StandardScaler.



```
# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Model Training:

- Trained a Support Vector Machine (SVM) model with a linear kernel.



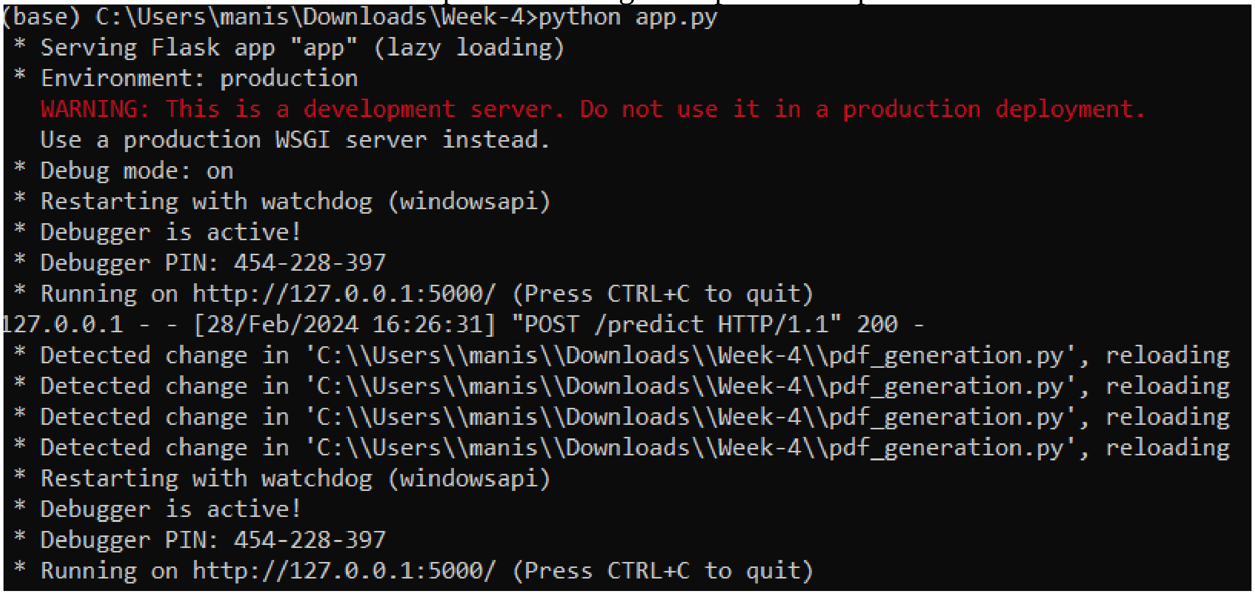
```
# Train the model
model = SVC(kernel='linear', random_state=42)
model.fit(X_train, y_train)

# Save the trained model
joblib.dump(model, 'breast_cancer_model.joblib')
```

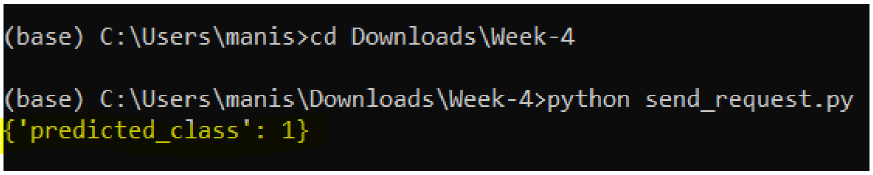
- Achieved an accuracy of 0.96 on the test set.

Flask Deployment:

- Created a Flask application with a prediction endpoint (/predict).
- Loaded the trained model and exposed it through the /predict endpoint.

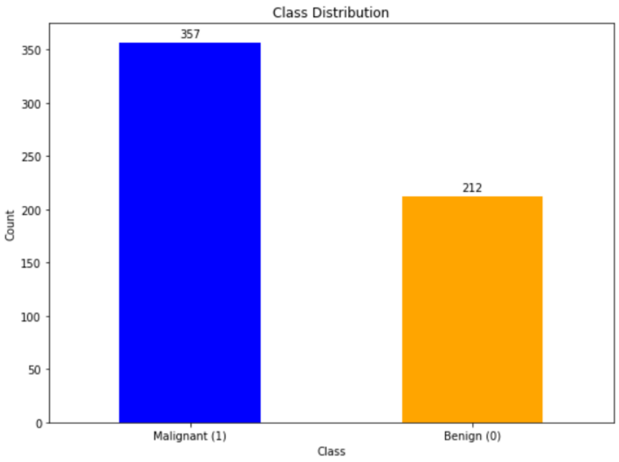


```
(base) C:\Users\manis\Downloads\Week-4>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 454-228-397
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [28/Feb/2024 16:26:31] "POST /predict HTTP/1.1" 200 -
 * Detected change in 'C:\\Users\\manis\\Downloads\\Week-4\\pdf_generation.py', reloading
 * Detected change in 'C:\\Users\\manis\\Downloads\\Week-4\\pdf_generation.py', reloading
 * Detected change in 'C:\\Users\\manis\\Downloads\\Week-4\\pdf_generation.py', reloading
 * Detected change in 'C:\\Users\\manis\\Downloads\\Week-4\\pdf_generation.py', reloading
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 454-228-397
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



```
(base) C:\Users\manis>cd Downloads\Week-4

(base) C:\Users\manis\Downloads\Week-4>python send_request.py
{'predicted_class': 1}
```


Class Distribution
Count
Malignant (1)
Benign (0)
Class