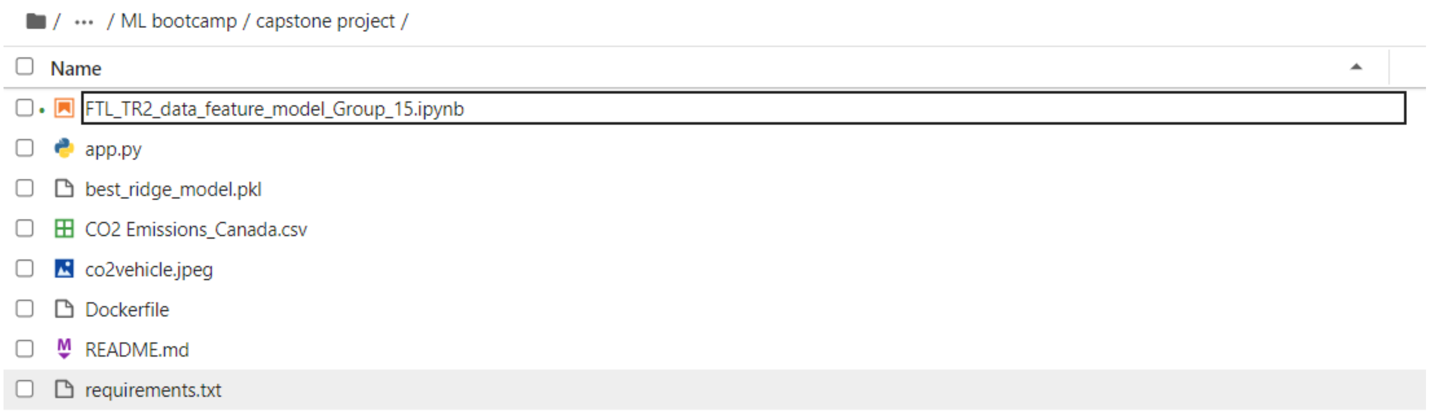
Machine Learning Project Documentation

Deployment

1. **Overview**

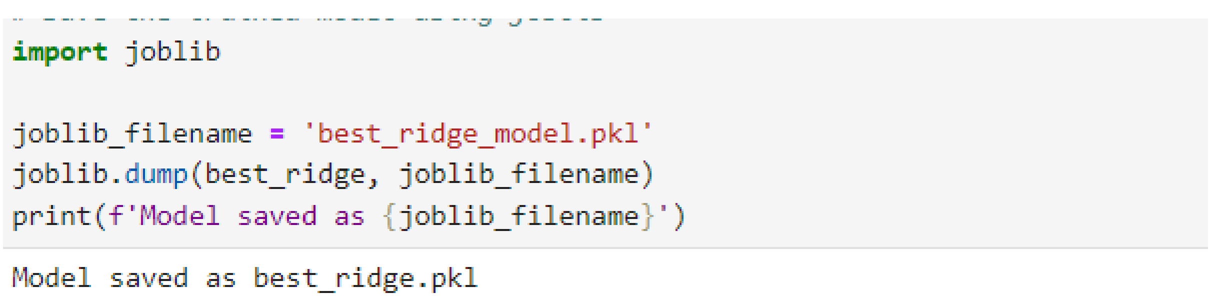
In the deployment phase, the trained machine learning model is packaged and made accessible to end users through a web application. The deployment process ensures that the model can be used in real-world scenarios, allowing users to input vehicle specifications and receive CO2 emissions predictions via an interactive interface built with Streamlit and served using Docker



```
/ ... / ML bootcamp / capstone project /
Name
FTL_TR2_data_feature_model_Group_15.ipynb
app.py
best_ridge_model.pkl
CO2 Emissions_Canada.csv
co2vehicle.jpeg
Dockerfile
README.md
requirements.txt
```

2. **Model Serialization**

The Ridge regression model, trained on vehicle specifications to predict CO2 emissions, was serialized using 'joblib' for efficient storage. The serialized model file, 'best_ridge_model.pkl', ensures that the trained model's state is preserved and can be loaded into the deployment environment for prediction tasks without retraining.



```python
import joblib

joblib_filename = 'best_ridge_model.pkl'
joblib.dump(best_ridge, joblib_filename)
print(f'Model saved as {joblib_filename}')
```

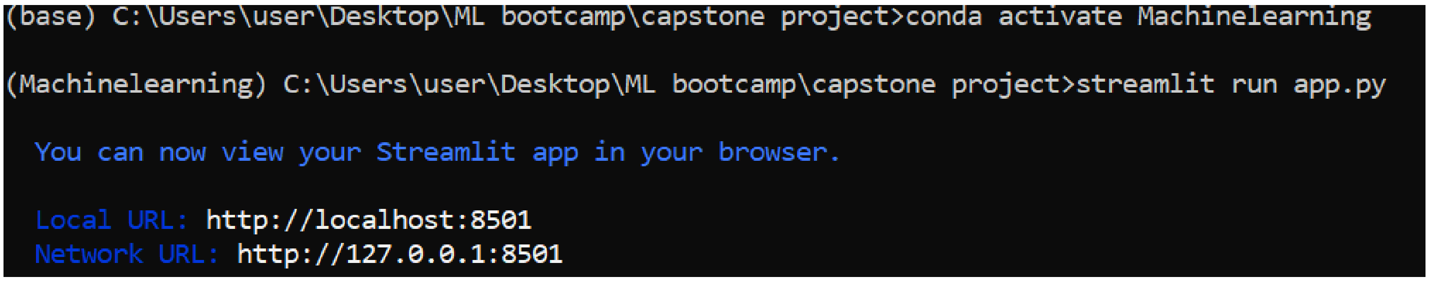
```
Model saved as best_ridge.pkl
```

3. **Model Serving**

The serialized model is served via a Streamlit application. Streamlit provides an intuitive interface for users to input data and see predictions in real-time. The app runs inside a Docker container, ensuring consistency across different deployment environments. The containerized app can be deployed on any platform supporting Docker, such as cloud platforms or on-premises servers.

4. **API Integration**

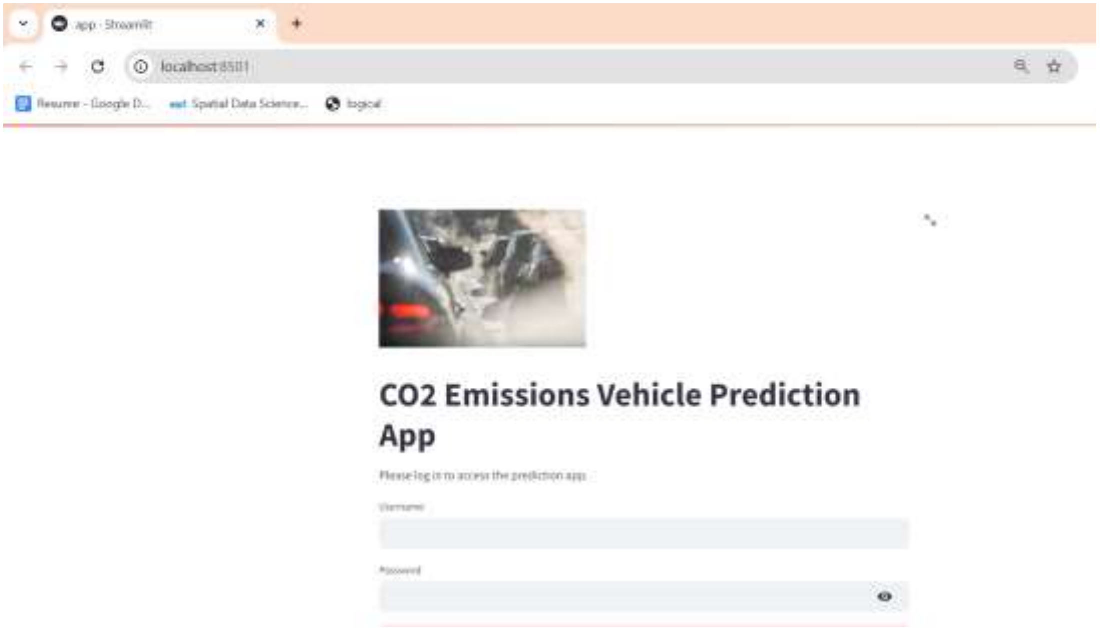
In this deployment, there is no separate API. The Streamlit web app serves as the interface for input and output. However, if an API was needed, a REST API could be created using Flask or FastAPI, exposing endpoints that accept vehicle specification data and return predicted CO2 emissions in JSON format.



```
(base) C:\Users\user\Desktop\ML bootcamp\capstone project>conda activate Machinelearning

(Machinelearning) C:\Users\user\Desktop\ML bootcamp\capstone project>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://127.0.0.1:8501
```

5. **Security Considerations**

The app includes a simple username and password authentication mechanism to restrict access to authorized users only by using login credential . Although this setup is minimal, it can be extended to include stronger security measures such as OAuth for authentication, HTTPS for encrypted communication, and API tokens for external API requests after making deploying to Cloud services



CO2 Emissions Vehicle Prediction App

Please log in to access the prediction app

Username

Password

6. **Monitoring and Logging**

The deployed app can be monitored using Docker's built-in logging and monitoring capabilities.

Additionally,

Streamlit provides useful logs about the app's usage, which can be extended with tools like Prometheus and
Grafana for more robust monitoring of API requests, response times, and model performance.