Flask App

Riya Gaur

Dec 28th,
Data Glacier!

- The dataset used for this project is `winequality-red.csv`, which contains information about the physicochemical properties of red wine samples and their corresponding quality scores.
- The dataset was loaded using `pandas`, and the features (`X`) and target variable (`y`) were separated. The `quality` column was used as the target variable, while all other columns were used as features.
- - The dataset was split into training and testing sets using an 80-20 split with the `train_test_split` function from `scikit-learn`.
- A `RandomForestClassifier` was chosen for the task due to its robustness and ability to handle multivariate features effectively.
- The model was trained using the training dataset (`X_train` and `y_train`).
- The trained model was serialized and saved as `wine_quality_model.pkl` using Python's `pickle` module.
- This allows the model to be reused in the Flask app without needing to retrain it every time.

The following are snapshots of output produced from training and testing the model / dataset.

```
Dataset Head:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5

Null Values in Dataset:
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

2.

```
Column Names:
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')

First Few Rows:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5

Model Accuracy: 65.94%

Model saved as wine_quality_model.pkl

Loaded Model Accuracy: 65.94%
```

save_model.py - /Users/riyagaur/Downloads/flask app/save_model.py (3.11.3)

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import pickle

# Load the dataset
data = pd.read_csv('winequality-red.csv', delimiter=';')

# Prepare features and target
X = data.drop(columns=['quality'])
y = data['quality']

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_

# Train the model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Save the model
with open('wine_quality_model.pkl', 'wb') as file:
    pickle.dump(model, file)

print("Model saved as wine_quality_model.pkl")
```

- The saved `wine_quality_model.pkl` file was loaded into the app using `pickle`.
- This ensures that the trained model is available for making predictions.
- The Flask app was run locally on `http://127.0.0.1:5000`.
- Debug mode was enabled to allow troubleshooting during development.

The following is the deployment code for flask app, runtime on terminal & web snapshot.

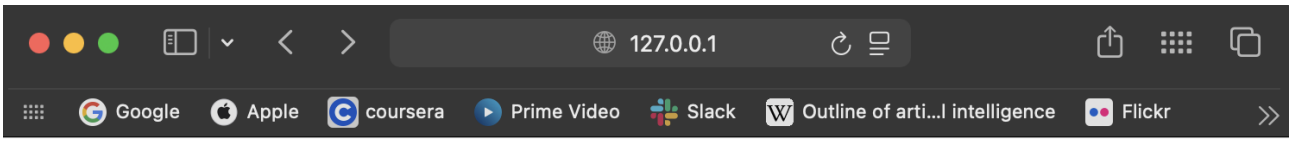
app.py - /Users/riyagaur/Downloads/flask app/app.py (3.11.3)

```python
from flask import Flask, request, jsonify
import pickle
import numpy as np

app = Flask(__name__)

# Load the saved model
with open('wine_quality_model.pkl', 'rb') as file:
    model = pickle.load(file)

@app.route('/')
def home():
    return "Welcome to the Wine Quality Prediction API!"

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Get features from the request
        data = request.json
        features = np.array(data['features']).reshape(1, -1)

        # Make prediction
        prediction = model.predict(features)
        return jsonify({'prediction': int(prediction[0])})
    except Exception as e:
        return jsonify({'error': str(e)})

if __name__ == '__main__':
    app.run(debug=True, port=5000)
```

1.

```
(base) riyagaur@Riyas-MacBook flask app % 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
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with watchdog (fsevents)
 * Debugger is active!
 * Debugger PIN: 203-921-581
127.0.0.1 - - [28/Dec/2024 11:43:12] "GET / HTTP/1.1" 200 -
```

2.



Welcome to the Wine Quality Prediction API!

3.
4. The `/predict` endpoint is designed to take input features (as a JSON payload) and return the predicted wine quality.
5. The endpoint returns a JSON response with the predicted wine quality.

```
(base) riyagaur@Riyas-MacBook flask app % curl -X POST -H "Content-Type: application/json" \
-d '{"features": [7.4, 0.7, 0.0, 1.9, 0.076, 11.0, 34.0, 0.9978, 3.51, 0.56, 9.4]}' \
http://127.0.0.1:5000/predict

{
  "prediction": 5
}
(base) riyagaur@Riyas-MacBook flask app %
```