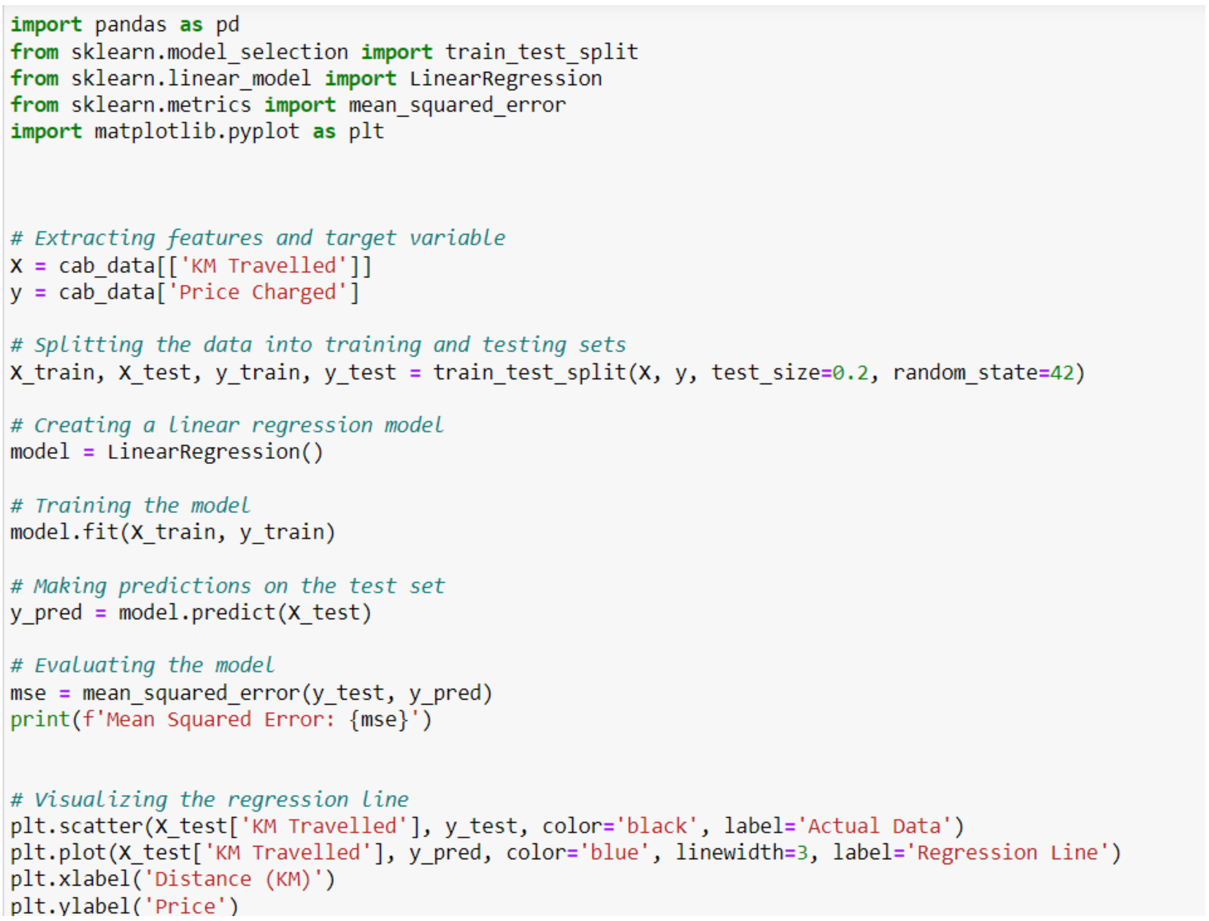
MODEL DEPLOYMENT ON FLASK

NAME: Aysha Abdul Azeez

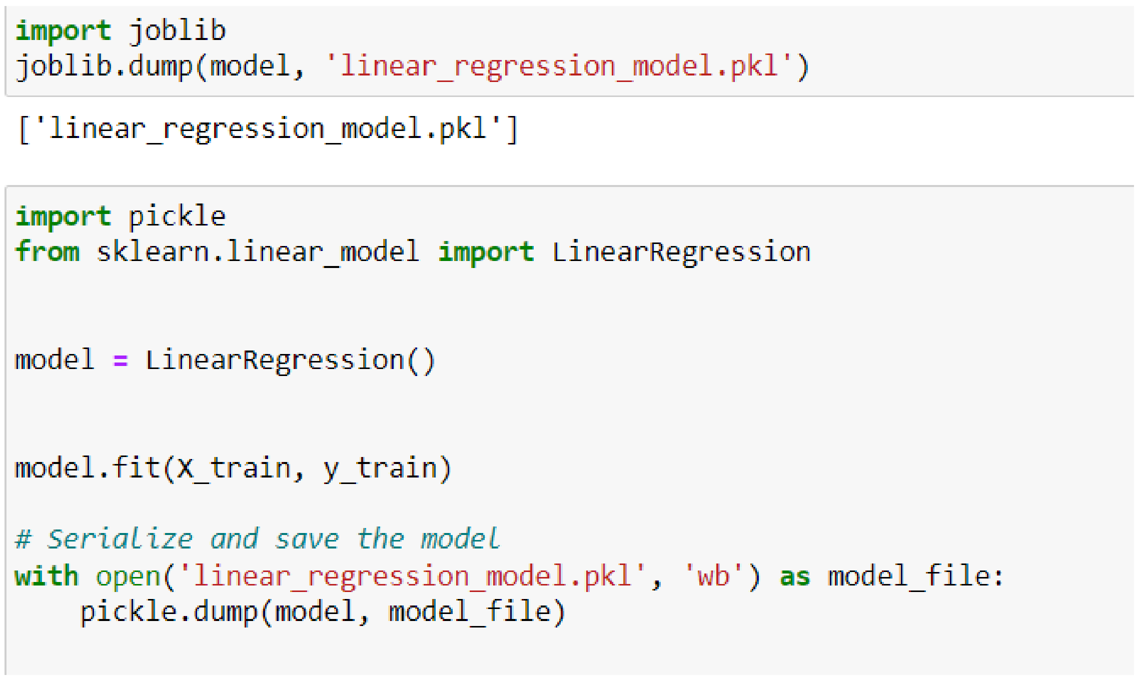
BATCH CODE: LISUM27

Submission Date:23/11/23



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt



# Extracting features and target variable
X = cab_data[['KM Travelled']]
y = cab_data['Price Charged']

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Creating a linear regression model
model = LinearRegression()

# Training the model
model.fit(X_train, y_train)

# Making predictions on the test set
y_pred = model.predict(X_test)

# Evaluating the model
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')


# Visualizing the regression line
plt.scatter(X_test['KM Travelled'], y_test, color='black', label='Actual Data')
plt.plot(X_test['KM Travelled'], y_pred, color='blue', linewidth=3, label='Regression Line')
plt.xlabel('Distance (KM)')
plt.ylabel('Price')
```

Figure 1.Creation of model

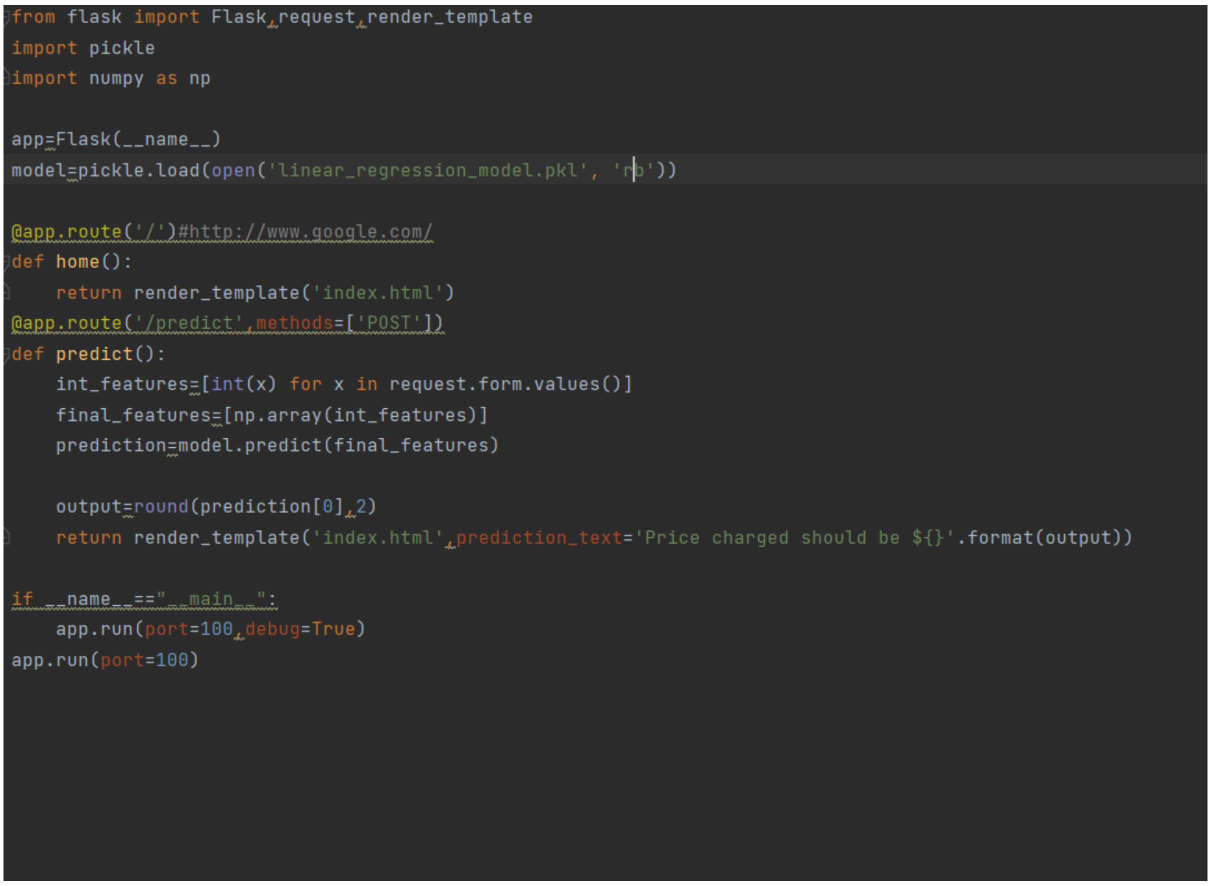


```python
import joblib
joblib.dump(model, 'linear_regression_model.pkl')
```

```
['linear_regression_model.pkl']
```

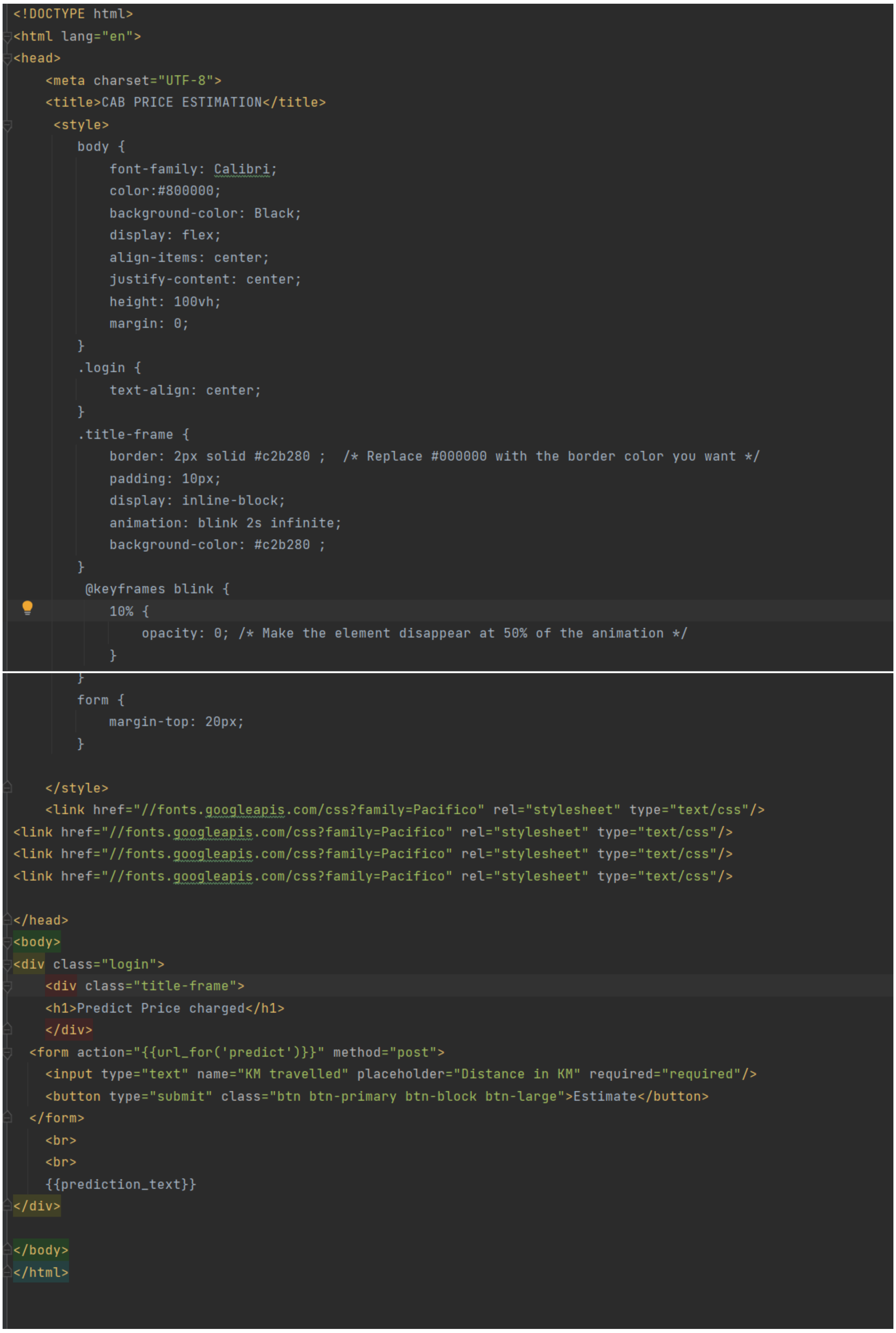
```python
import pickle
from sklearn.linear_model import LinearRegression


model = LinearRegression()


model.fit(X_train, y_train)

# Serialize and save the model
with open('linear_regression_model.pkl', 'wb') as model_file:
    pickle.dump(model, model_file)
```

Figure 2.Saving model as a pickle file



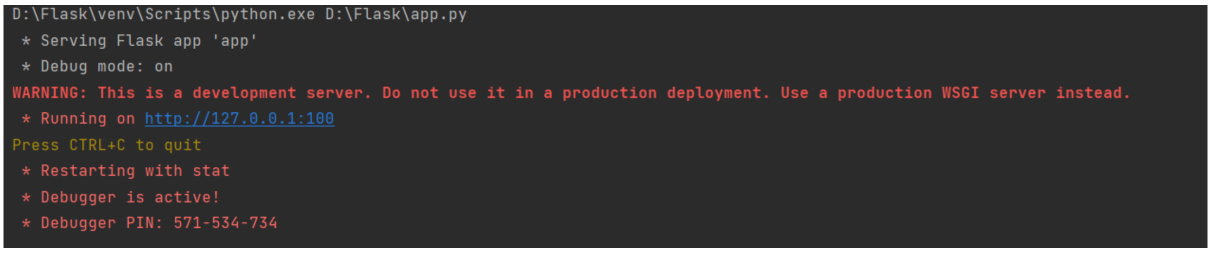
```python
from flask import Flask,request,render_template
import pickle
import numpy as np

app=Flask(__name__)
model=pickle.load(open('linear_regression_model.pkl', 'rb'))

@app.route('/')#http://www.google.com/
def home():
    return render_template('index.html')
@app.route('/predict',methods=['POST'])
def predict():
    int_features=[int(x) for x in request.form.values()]
    final_features=[np.array(int_features)]
    prediction=model.predict(final_features)

    output=round(prediction[0],2)
    return render_template('index.html',prediction_text='Price charged should be ${}'.format(output))

if __name__=="__main__":
    app.run(port=100,debug=True)
app.run(port=100)
```

Figure 3.Python code for deployment in Flask



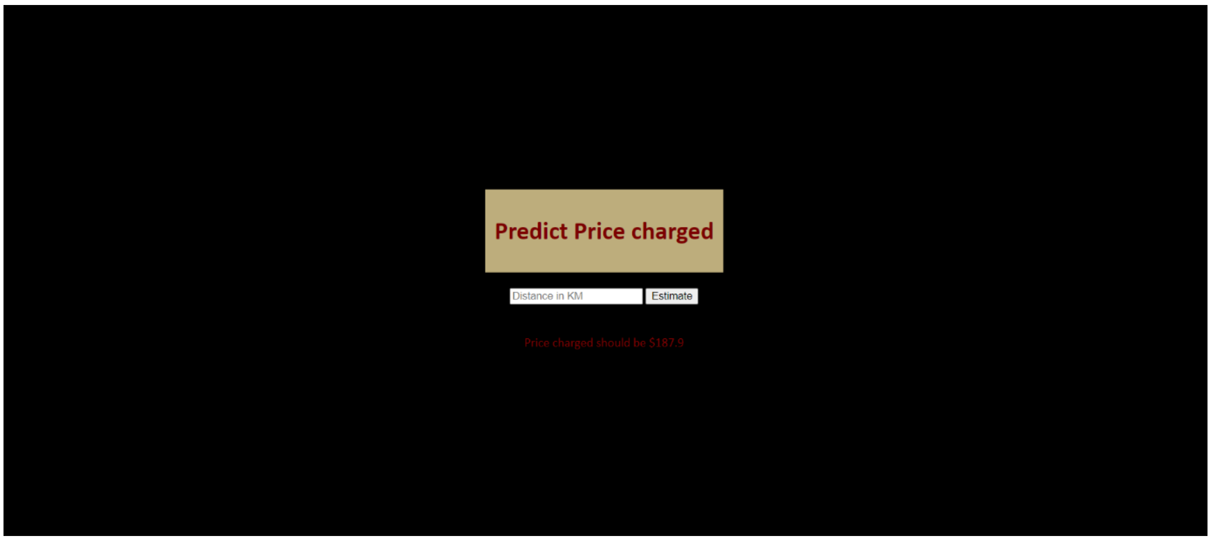
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CAB PRICE ESTIMATION</title>
     <style>
        body {
            font-family: Calibri;
            color:#800000;
            background-color: Black;
            display: flex;
            align-items: center;
            justify-content: center;
            height: 100vh;
            margin: 0;
        }
        .login {
            text-align: center;
        }
        .title-frame {
            border: 2px solid #c2b280 ;  /* Replace #000000 with the border color you want */
            padding: 10px;
            display: inline-block;
            animation: blink 2s infinite;
            background-color: #c2b280 ;
        }
         @keyframes blink {
            10% {
                opacity: 0; /* Make the element disappear at 50% of the animation */
            }
        }
        form {
            margin-top: 20px;
        }

    </style>
    <link href="//fonts.googleapis.com/css?family=Pacifico" rel="stylesheet" type="text/css"/>
<link href="//fonts.googleapis.com/css?family=Pacifico" rel="stylesheet" type="text/css"/>
<link href="//fonts.googleapis.com/css?family=Pacifico" rel="stylesheet" type="text/css"/>
<link href="//fonts.googleapis.com/css?family=Pacifico" rel="stylesheet" type="text/css"/>

</head>
<body>
<div class="login">
    <div class="title-frame">
    <h1>Predict Price charged</h1>
    </div>
  <form action="{{url_for('predict')}}" method="post">
    <input type="text" name="KM travelled" placeholder="Distance in KM" required="required"/>
    <button type="submit" class="btn btn-primary btn-block btn-large">Estimate</button>
  </form>
    <br>
    <br>
    {{prediction_text}}
</div>

</body>
</html>
```

Figure 4.html code for deployment



```
D:\Flask\venv\Scripts\python.exe D:\Flask\app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:100
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 571-534-734
```

Figure 5.Output obtained for the model deployment



Predict Price charged

Distance in KM

Estimate

Price charged should be $187.9

Figure 6.Web page