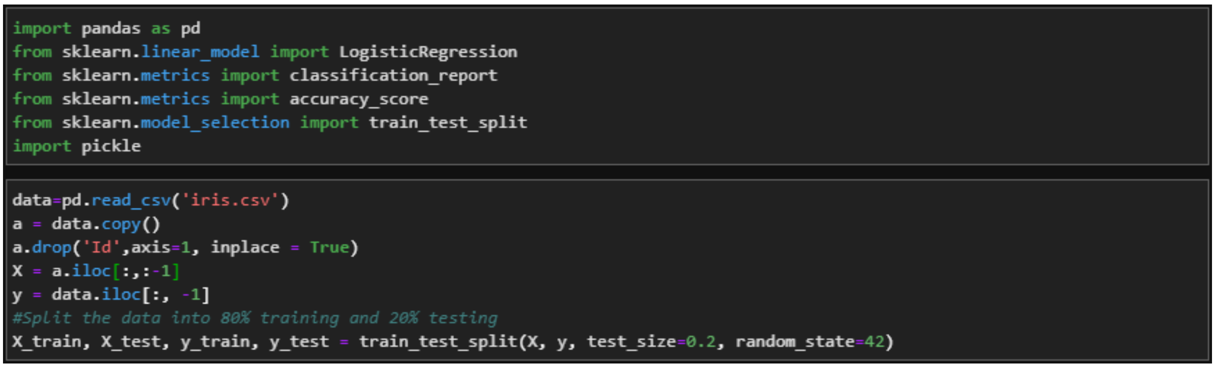
NAME : BURAK DEMİR

BATCH CODE : LISP01

SUBMISSION DATE : 21/03/2021

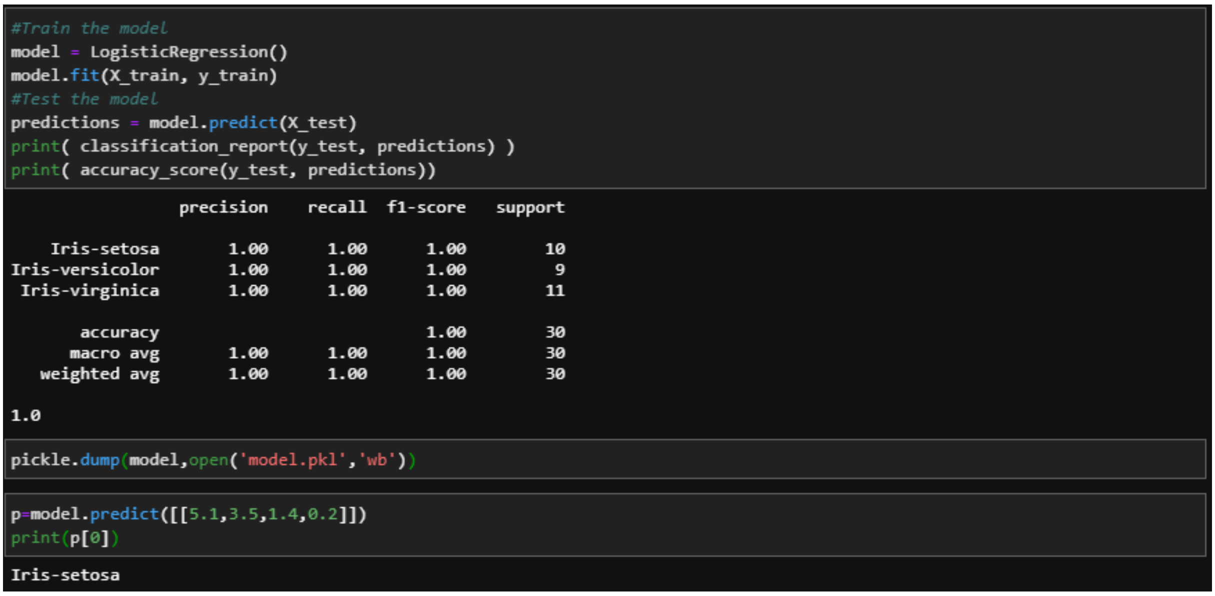
SUBMITTED TO : DATA GLACIER

Creating The Model



```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import pickle
```

```
data=pd.read_csv('iris.csv')
a = data.copy()
a.drop('Id',axis=1, inplace = True)
X = a.iloc[:,:-1]
y = data.iloc[:, -1]
#Split the data into 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```



```
#Train the model
model = LogisticRegression()
model.fit(X_train, y_train)
#Test the model
predictions = model.predict(X_test)
print( classification_report(y_test, predictions) )
print( accuracy_score(y_test, predictions))
```

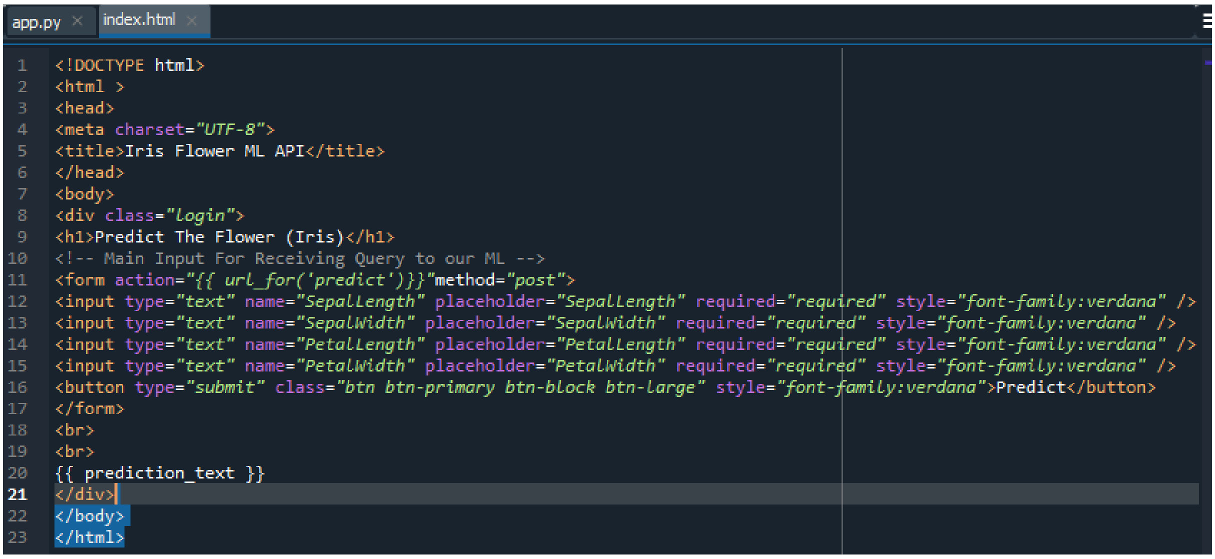
```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30

1.0
```

```
pickle.dump(model,open('model.pkl','wb'))
```

```
p=model.predict([[5.1,3.5,1.4,0.2]])
print(p[0])
```

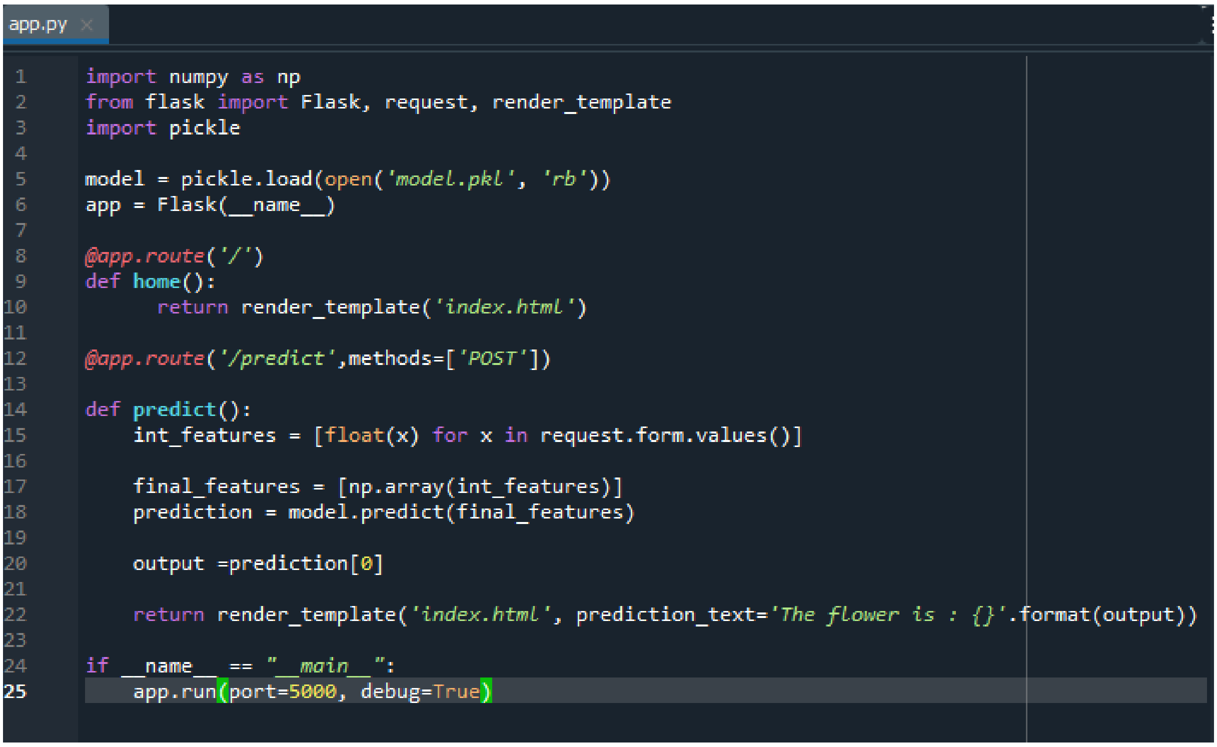
```
Iris-setosa
```

index.html



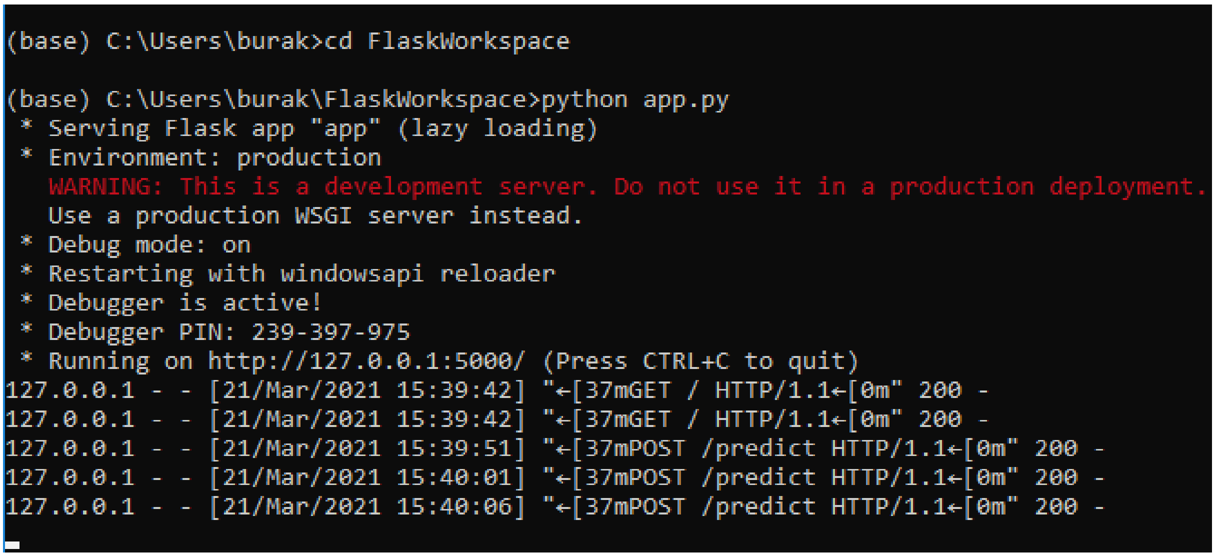
```html
<!DOCTYPE html>
<html >
<head>
<meta charset="UTF-8">
<title>Iris Flower ML API</title>
</head>
<body>
<div class="login">
<h1>Predict The Flower (Iris)</h1>
<!-- Main Input For Receiving Query to our ML -->
<form action="{{ url_for('predict')}}"method="post">
<input type="text" name="SepalLength" placeholder="SepalLength" required="required" style="font-family:verdana" />
<input type="text" name="SepalWidth" placeholder="SepalWidth" required="required" style="font-family:verdana" />
<input type="text" name="PetalLength" placeholder="PetalLength" required="required" style="font-family:verdana" />
<input type="text" name="PetalWidth" placeholder="PetalWidth" required="required" style="font-family:verdana" />
<button type="submit" class="btn btn-primary btn-block btn-large" style="font-family:verdana">Predict</button>
</form>
<br>
<br>
{{ prediction_text }}
</div>
</body>
</html>
```

Flask App

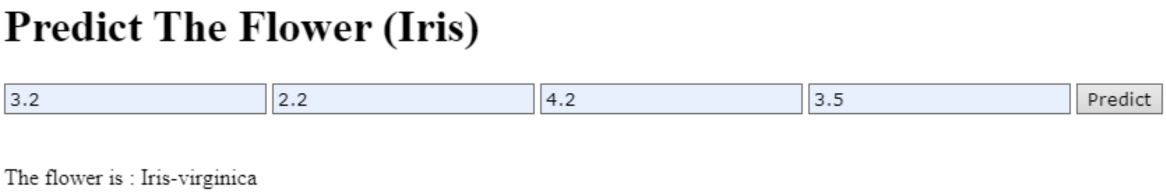


```python
import numpy as np
from flask import Flask, request, render_template
import pickle

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

@app.route('/')
def home():
      return render_template('index.html')

@app.route('/predict',methods=['POST'])

def predict():
    int_features = [float(x) for x in request.form.values()]

    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output =prediction[0]

    return render_template('index.html', prediction_text='The flower is : {}'.format(output))

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

python app.py



```
(base) C:\Users\burak>cd FlaskWorkspace

(base) C:\Users\burak\FlaskWorkspace>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 windowsapi reloader
 * Debugger is active!
 * Debugger PIN: 239-397-975
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [21/Mar/2021 15:39:42] "←[37mGET / HTTP/1.1←[0m" 200 -
127.0.0.1 - - [21/Mar/2021 15:39:42] "←[37mGET / HTTP/1.1←[0m" 200 -
127.0.0.1 - - [21/Mar/2021 15:39:51] "←[37mPOST /predict HTTP/1.1←[0m" 200 -
127.0.0.1 - - [21/Mar/2021 15:40:01] "←[37mPOST /predict HTTP/1.1←[0m" 200 -
127.0.0.1 - - [21/Mar/2021 15:40:06] "←[37mPOST /predict HTTP/1.1←[0m" 200 -
```



Predict The Flower (Iris)

3.2 | 2.2 | 4.2 | 3.5

The flower is : Iris-virginica