STEPS OF DEPLOYMENT OF A SIMPLE MACHINE LEARNING MODEL ON FLASK

Name: Daniela Alvarez Zegarra

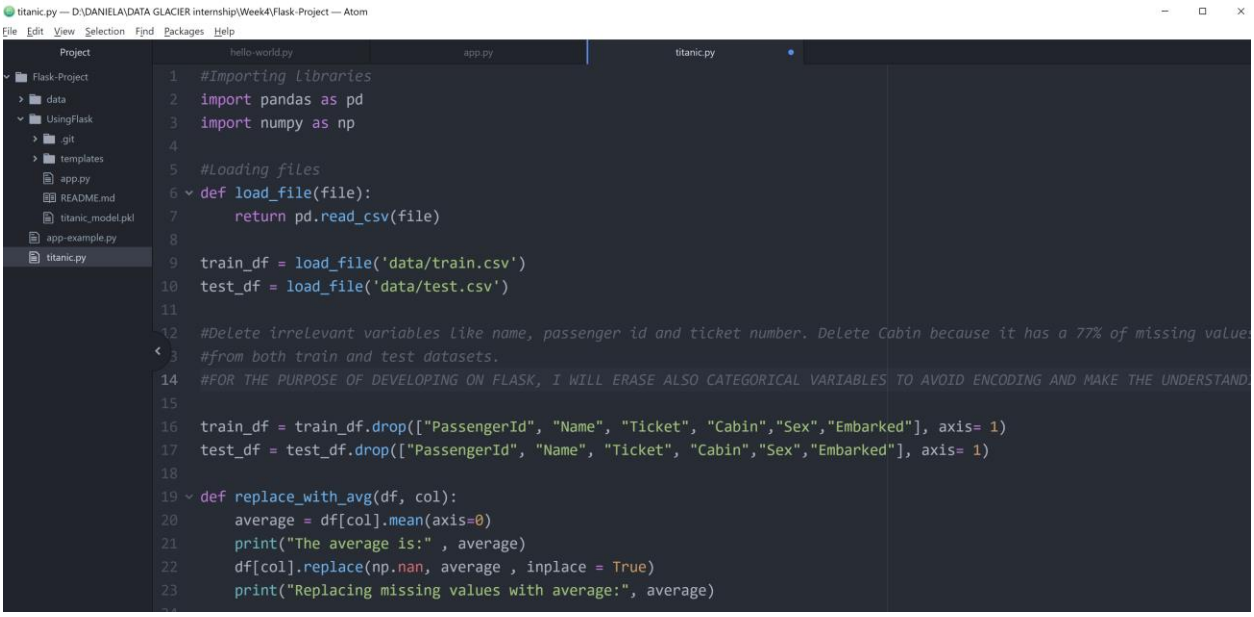
Batch code : LISUM03

Submission date: 20-09-2021

Submitted to: Github

Step 1

Using Titanic Dataset with only numerical variables.



titanic.py — D:\DANIELA\DATA GLACIER internship\Week4\Flask-Project — Atom

File Edit View Selection Find Packages Help

Project

- Flask-Project
  - data
  - UsingFlask
    - .git
    - templates
    - app.py
    - README.md
    - titanic_model.pkl
  - app-example.py
  - titanic.py

hello-world.py | app.py | titanic.py

```python
replace_with_avg(train_df, "Age")

train_target = train_df["Survived"]
train_features = train_df.drop(["Survived"], axis = 1)

#Select algorithm
from sklearn.ensemble import RandomForestClassifier
clf_rf = RandomForestClassifier()
#Training
clf_rf.fit(train_features, train_target)

#Transformations to my test set
age_average = train_df['Age'].mean(axis=0)
print("The average is:" , age_average)

test_df['Age'].replace(np.nan, age_average , inplace = True)
print("Replacing missing values in test dataset with average:", age_average)

fare_average = train_df['Fare'].mean(axis=0)
print("The average is:" , fare_average)

test_df['Fare'].replace(np.nan, fare_average , inplace = True)
print("Replacing missing values in test dataset with average:", fare_average)

predictions = clf_rf.predict(test_df)
```

Step 2

Saving the model on disk

```python
 #Save the model to disk

import pickle

filename = "titanic_model.pkl"

pickle.dump(clf_rf, open(filename, 'wb'))
```

Step 3

Using Flask to make a web API for our ML model

app.py — D:\DANIELA\DATA GLACIER internship\Week4\Flask-Project — Atom

File Edit View Selection Find Packages Help

Project: Flask-Project / data / UsingFlask / .git / templates / app.py / README.md / titanic_model.pkl / app-example.py / titanic.py

hello-world.py | app.py

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

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

@app.route('/') #http://www.google.com/
def home():
    return render_template('index.html')
@app.route('/predict', methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    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='Would you survive? {} (1=survived, 0=deceased)'.format(output))

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

Step 4

Using a simple form with HTML and CSS to gather values and make a prediction.

index.html — D:\DANIELA\DATA GLACIER internship\Week4\Flask-Project — Atom

File Edit View Selection Find Packages Help

Project: Flask-Project / data / UsingFlask / .git / templates / index.html / app.py / README.md / titanic_model.pkl / app-example.py / titanic.py

hello-world.py | app.py | titanic.py | index.html

```html
    </style>
</head>
<body>
    <div id="registration-form">
      <div class='fieldset'>
        <legend>Titanic Survival Prediction</legend>
        <form action="{{ url_for('predict')}}" method="post">
            <input type="text" placeholder="Class (1 = 1st; 2 = 2nd; 3 = 3rd)" name='class' id='class' data-required="true
            <input type="text" placeholder="Age"  name='age' data-required="true" data-error-message="Your age is required
            <input type="text" placeholder="Number of siblings/spouses aboard" name='sibsp' data-required="true" data-erro
            <input type="text" placeholder="Number of parents/children aboard"  name='parchi' data-required="true" data-er
            <input type="text" placeholder="Passenger Fare"  name='fare' data-required="true" data-error-message="Your Pas
          <input type="submit" value="predict">
        </form>

      <br>
      <br>

      {{ prediction_text}}

      </div>
    </div>
</body>
</html>
```

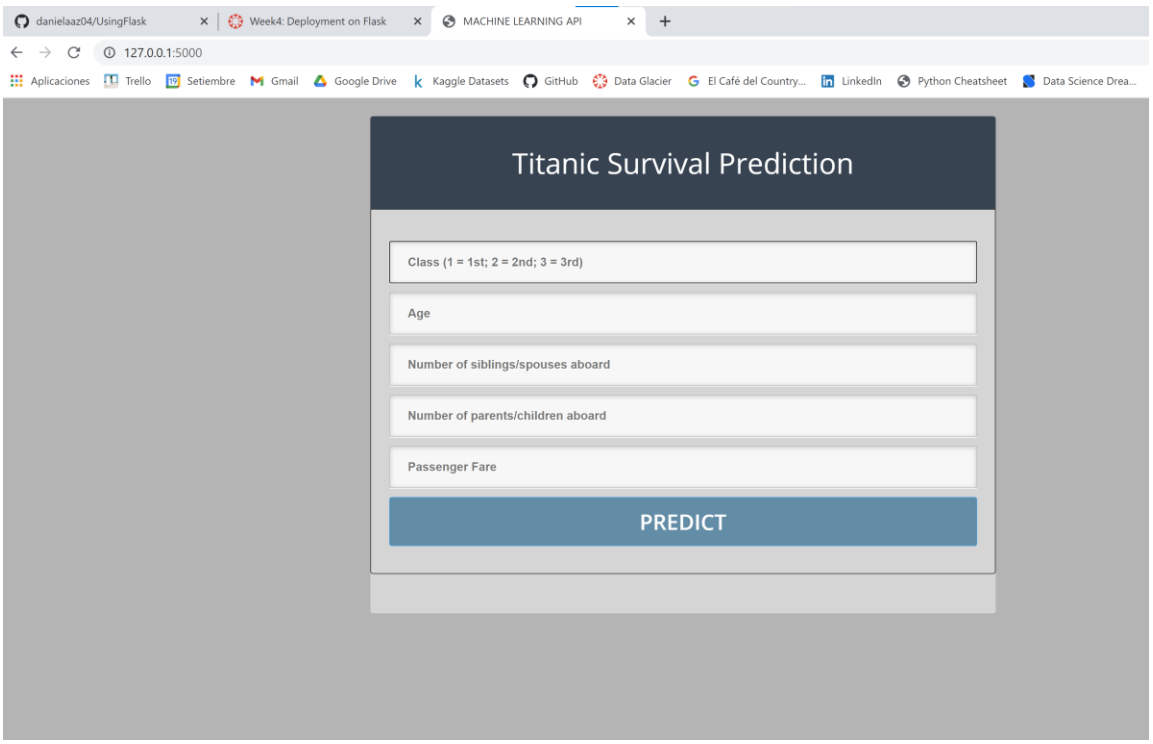
Step 5

After creating a repo on Github and cloning it to my computer, I move the necessary files to the repository and run my app.py file

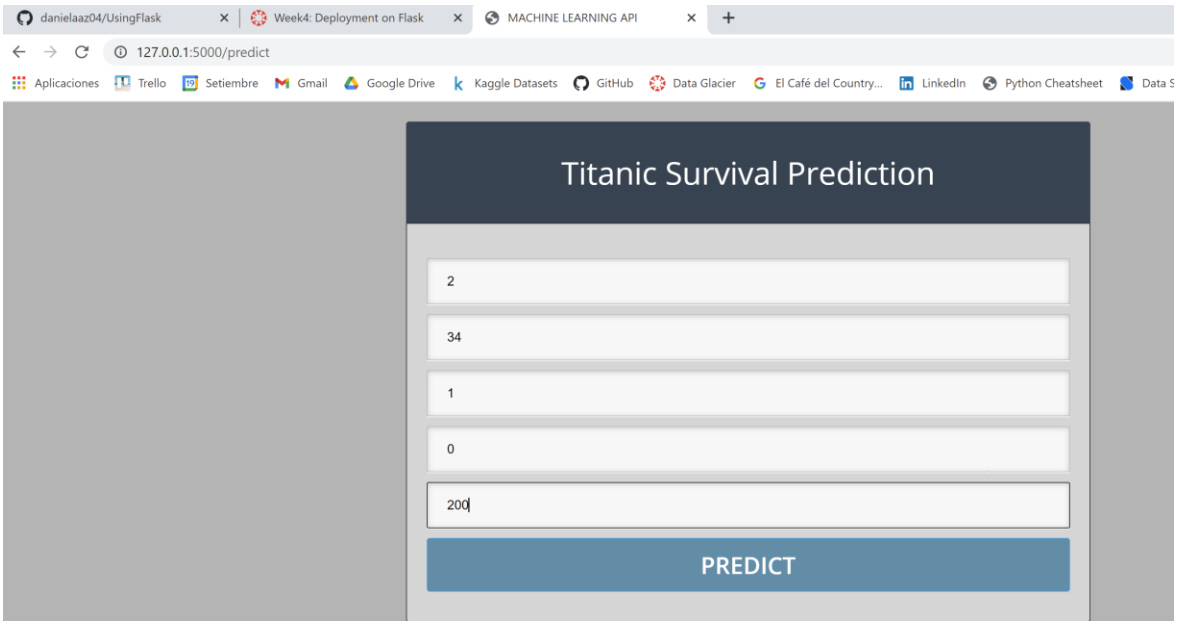
```
danielaaz04@DESKTOP-DANIELA: /mnt/d/DANIELA/DATA GLACIER internship/Week4/Flask-Project/UsingFlask
(base) danielaaz04@DESKTOP-DANIELA:/mnt/c/Windows/System32$ cd
(base) danielaaz04@DESKTOP-DANIELA:~$ cd ..
(base) danielaaz04@DESKTOP-DANIELA:/home$ cd ..
(base) danielaaz04@DESKTOP-DANIELA:/$ cd mnt
(base) danielaaz04@DESKTOP-DANIELA:/mnt$ cd d
(base) danielaaz04@DESKTOP-DANIELA:/mnt/d$ cd DANIELA/
(base) danielaaz04@DESKTOP-DANIELA:/mnt/d/DANIELA$ cd DATA\ GLACIER\ internship/
(base) danielaaz04@DESKTOP-DANIELA:/mnt/d/DANIELA/DATA GLACIER internship$ cd Week4/
(base) danielaaz04@DESKTOP-DANIELA:/mnt/d/DANIELA/DATA GLACIER internship/Week4$ ls
lask-HelloWorldApp  Flask-Project
(base) danielaaz04@DESKTOP-DANIELA:/mnt/d/DANIELA/DATA GLACIER internship/Week4$ cd Flask-Project/
(base) danielaaz04@DESKTOP-DANIELA:/mnt/d/DANIELA/DATA GLACIER internship/Week4/Flask-Project$ ls
singFlask  app-example.py  data  titanic.py
(base) danielaaz04@DESKTOP-DANIELA:/mnt/d/DANIELA/DATA GLACIER internship/Week4/Flask-Project$ cd UsingFlask/
(base) danielaaz04@DESKTOP-DANIELA:/mnt/d/DANIELA/DATA GLACIER internship/Week4/Flask-Project/UsingFlask$ ls
EADME.md  app.py  templates  titanic_model.pkl
(base) danielaaz04@DESKTOP-DANIELA:/mnt/d/DANIELA/DATA GLACIER internship/Week4/Flask-Project/UsingFlask$ 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 inotify reloader
 * Debugger is active!
 * Debugger PIN: 974-093-224
```

Step 6

In my browser I put http://127.0.0.1:5000/ and get the form I created



Now I enter the values according to the question and press Predict button.



And get my prediction in the bottom.