STEPS OF CLOUD AND API DEPLOYMENET OF MACHINE LEARNING MODEL ON HEROKU

Name: Daniela Alvarez Zegarra

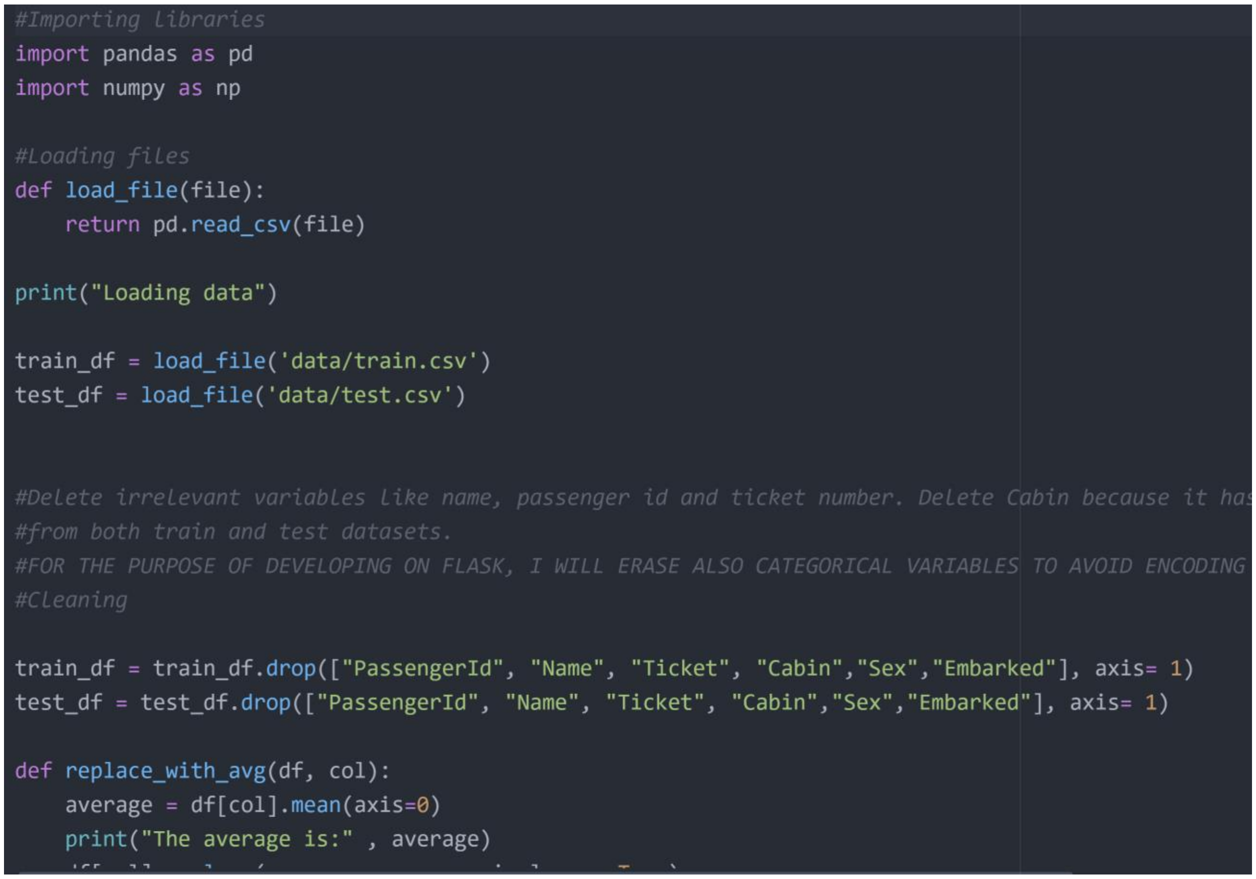
Batch Code: LISUM03

Submission Date: 03-10-2021

Submitted to: Github (https://github.com/danielaaz04/UsingHeroku)

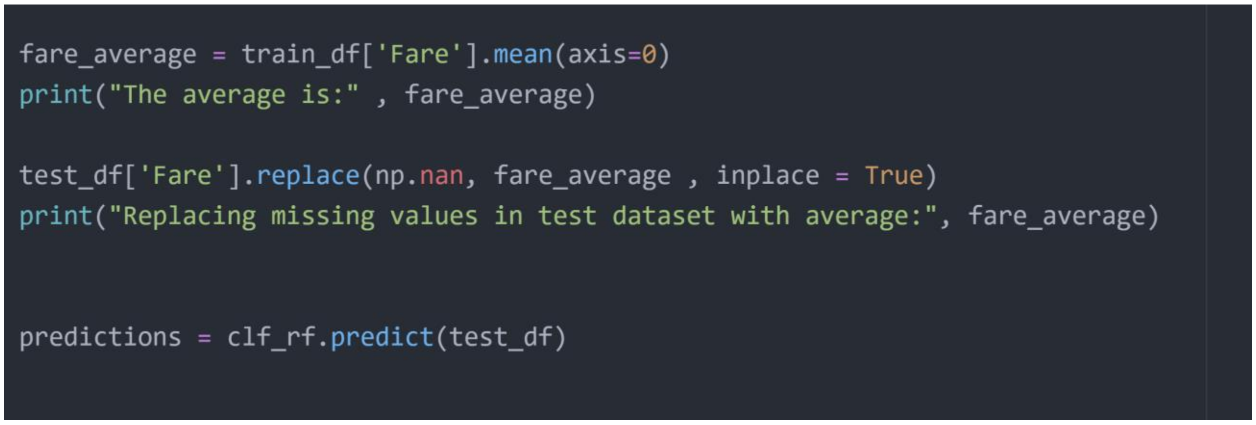
STEP 1

Using Titanic Dataset with only numerical variables



```
#Importing libraries
import pandas as pd
import numpy as np

#Loading files
def load_file(file):
    return pd.read_csv(file)

print("Loading data")

train_df = load_file('data/train.csv')
test_df = load_file('data/test.csv')


#Delete irrelevant variables like name, passenger id and ticket number. Delete Cabin because it has
#from both train and test datasets.
#FOR THE PURPOSE OF DEVELOPING ON FLASK, I WILL ERASE ALSO CATEGORICAL VARIABLES TO AVOID ENCODING
#Cleaning

train_df = train_df.drop(["PassengerId", "Name", "Ticket", "Cabin","Sex","Embarked"], axis= 1)
test_df = test_df.drop(["PassengerId", "Name", "Ticket", "Cabin","Sex","Embarked"], axis= 1)

def replace_with_avg(df, col):
    average = df[col].mean(axis=0)
    print("The average is:" , average)
```

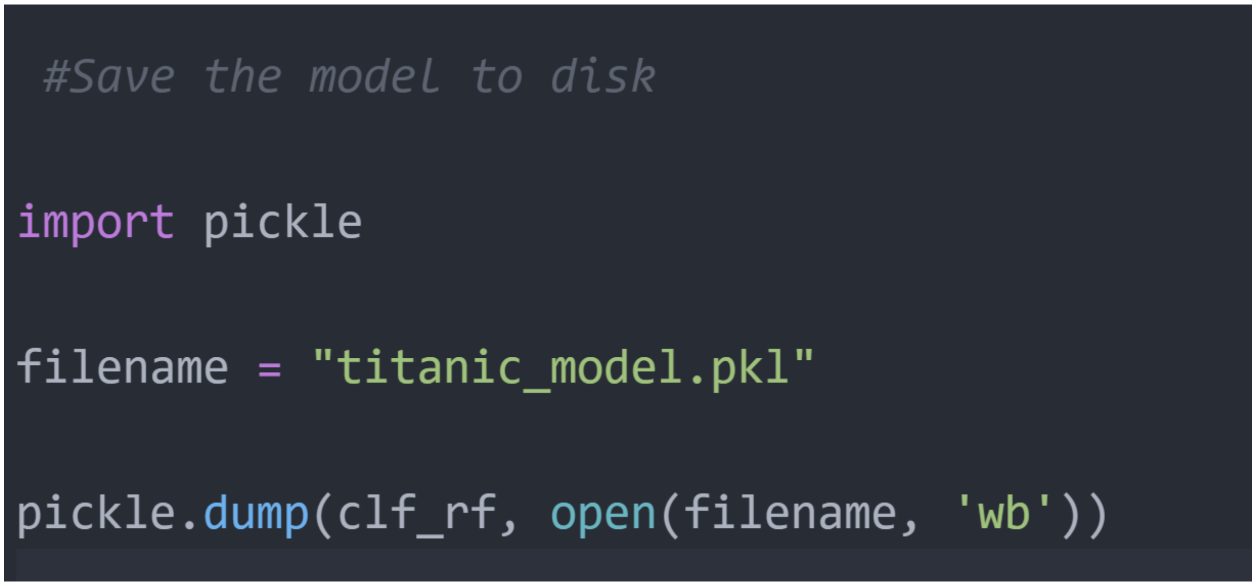
```python
def replace_with_avg(df, col):
    average = df[col].mean(axis=0)
    print("The average is:" , average)
    df[col].replace(np.nan, average , inplace = True)
    print("Replacing missing values with average:", average)

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)
```



```python
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

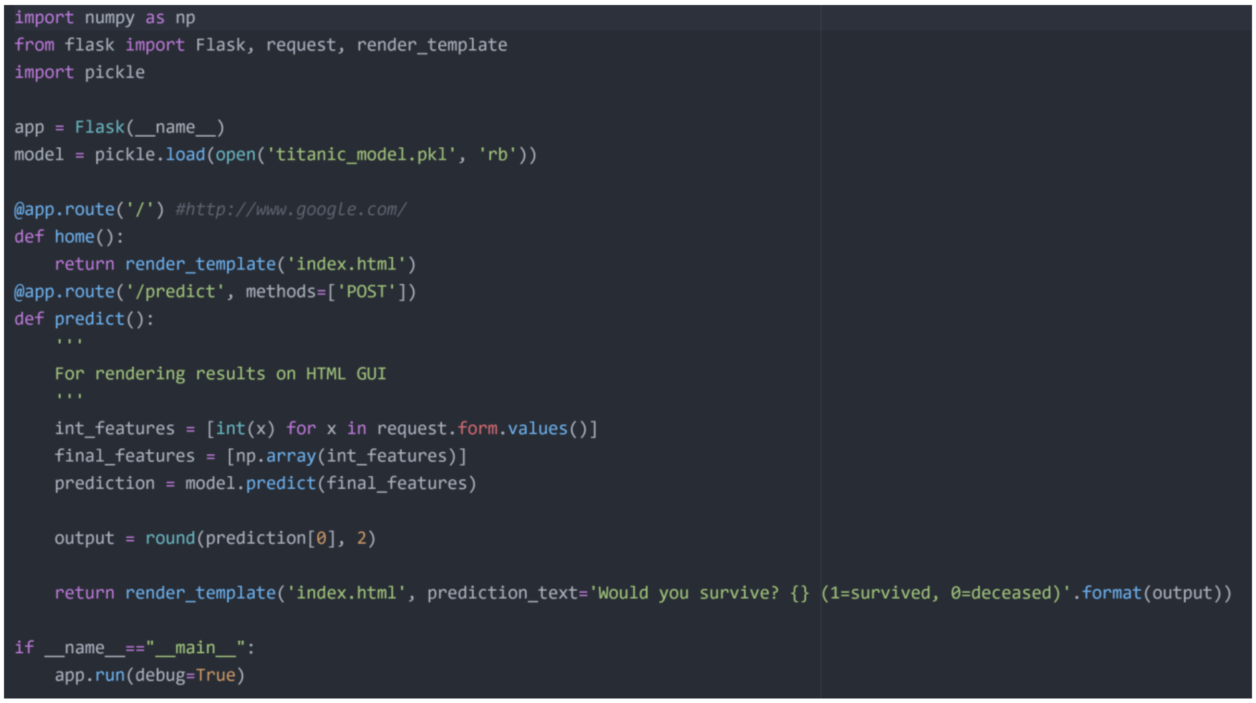
Saving the model to disk



```
 #Save the model to disk

import pickle

filename = "titanic_model.pkl"

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

STEP 3

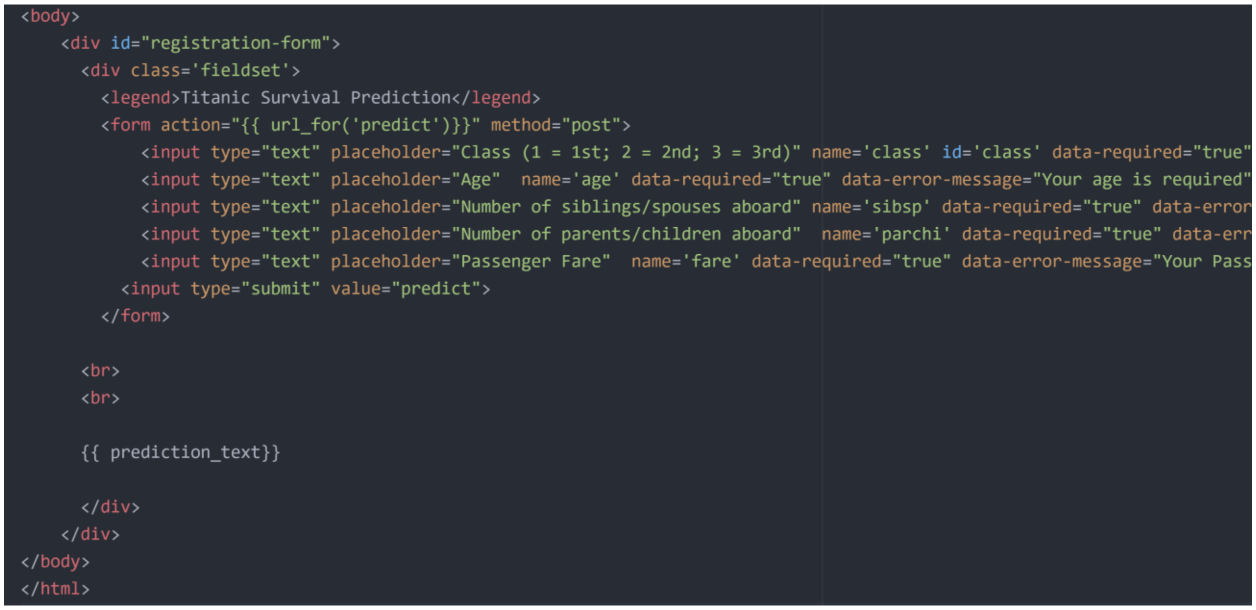
Using Flask to make a web API for our machine learning model



```
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(debug=True)
```

STEP 4

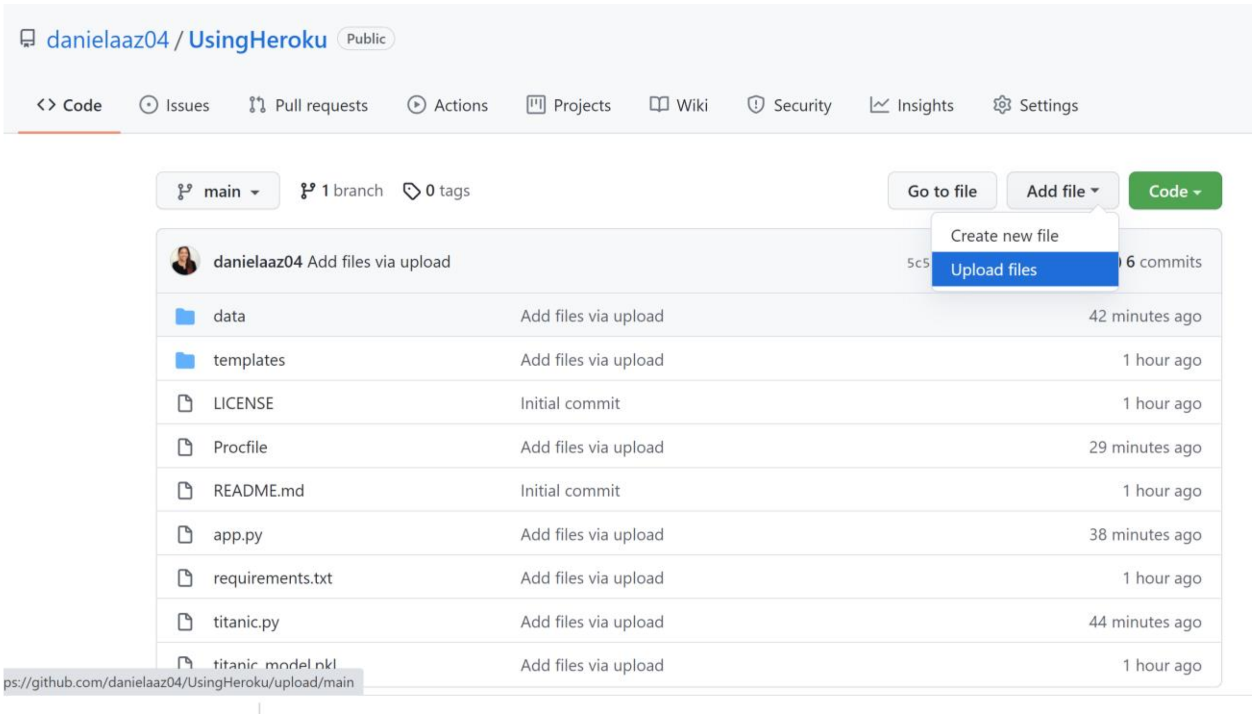
Using a simple form with HTML and CSS to gather values and make prediction.



```
<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-error
            <input type="text" placeholder="Number of parents/children aboard"  name='parchi' data-required="true" data-err
            <input type="text" placeholder="Passenger Fare"  name='fare' data-required="true" data-error-message="Your Pass
          <input type="submit" value="predict">
        </form>

      <br>
      <br>

      {{ prediction_text}}

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

STEP 5

Creating a repo on Github and uploading all my files.



danielaaz04 / UsingHeroku Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

Create new file
Upload files

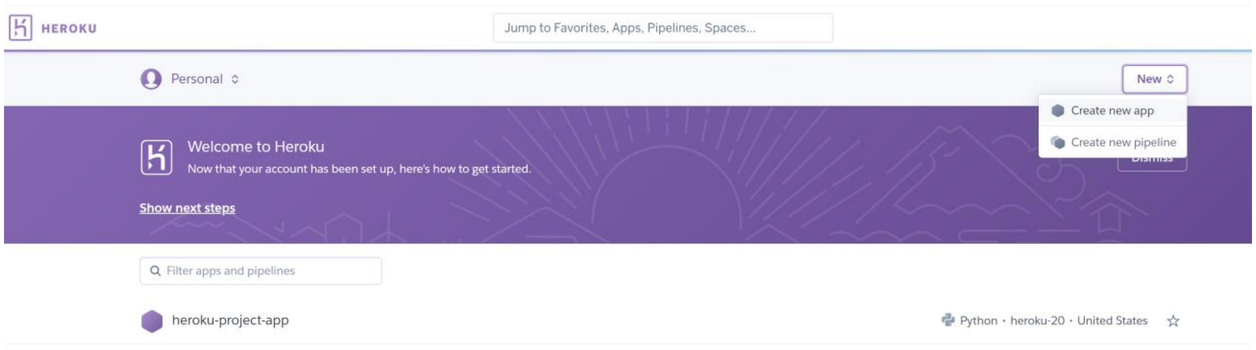
danielaaz04 Add files via upload 5c5 6 commits

data	Add files via upload	42 minutes ago
templates	Add files via upload	1 hour ago
LICENSE	Initial commit	1 hour ago
Procfile	Add files via upload	29 minutes ago
README.md	Initial commit	1 hour ago
app.py	Add files via upload	38 minutes ago
requirements.txt	Add files via upload	1 hour ago
titanic.py	Add files via upload	44 minutes ago
titanic_model.pkl	Add files via upload	1 hour ago

ps://github.com/danielaaz04/UsingHeroku/upload/main

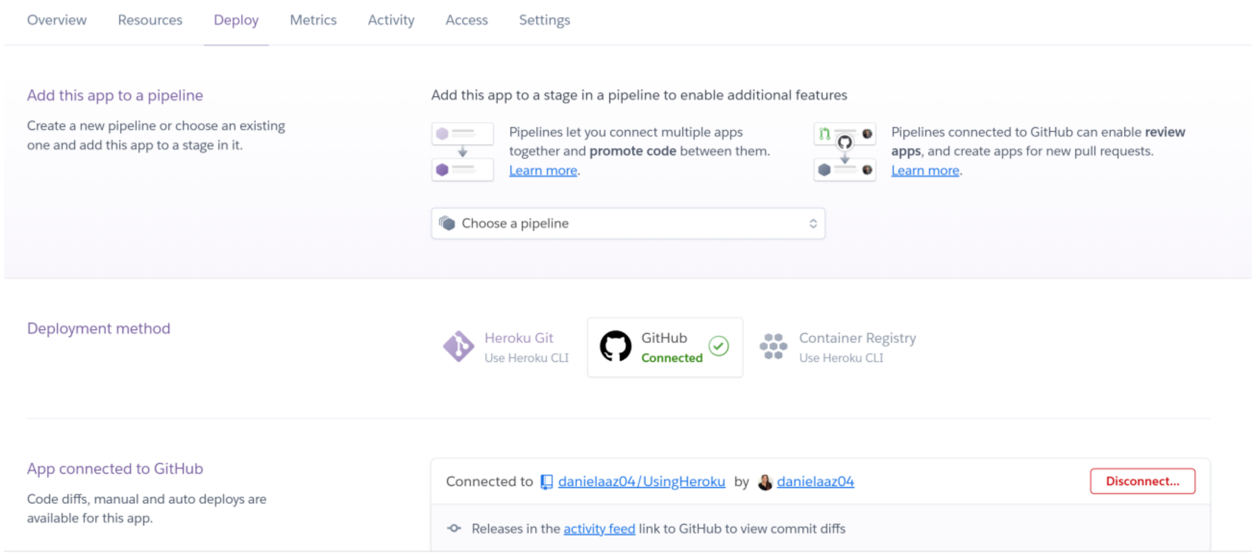
STEP 6

Creating an account on Heroku and clic on “create a new app”


HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Personal
New
Create new app
Create new pipeline
Welcome to Heroku
Now that your account has been set up, here's how to get started.
Show next steps
Filter apps and pipelines
heroku-project-app
Python • heroku-20 • United States

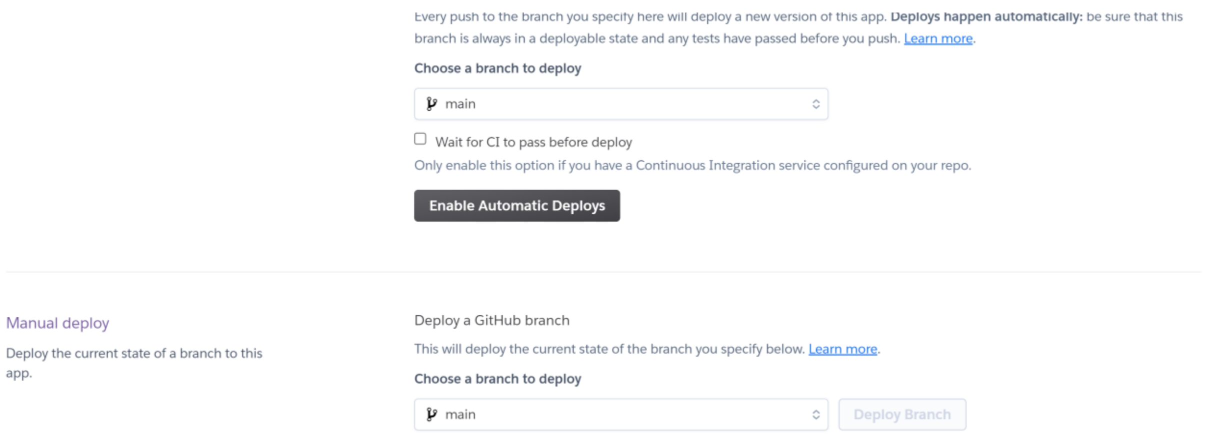
STEP 7

On “deploy” tab: Link Heroku app to my Github account and select repo to connect to.


Overview
Resources
Deploy
Metrics
Activity
Access
Settings
Add this app to a pipeline
Create a new pipeline or choose an existing one and add this app to a stage in it.
Add this app to a stage in a pipeline to enable additional features
Pipelines let you connect multiple apps together and promote code between them. Learn more.
Pipelines connected to GitHub can enable review apps, and create apps for new pull requests. Learn more.
Choose a pipeline
Deployment method
Heroku Git
Use Heroku CLI
GitHub
Connected
Container Registry
Use Heroku CLI
App connected to GitHub
Code diffs, manual and auto deploys are available for this app.
Connected to danielaaz04/UsingHeroku by danielaaz04
Disconnect...
Releases in the activity feed link to GitHub to view commit diffs

STEP 8

Scroll down to choose Manual Deploy and after making sure you are on the branch you want to deploy (in this case: main) then clic on “Deploy Branch”.



Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. Learn more.

Choose a branch to deploy

main

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Manual deploy

Deploy the current state of a branch to this app.

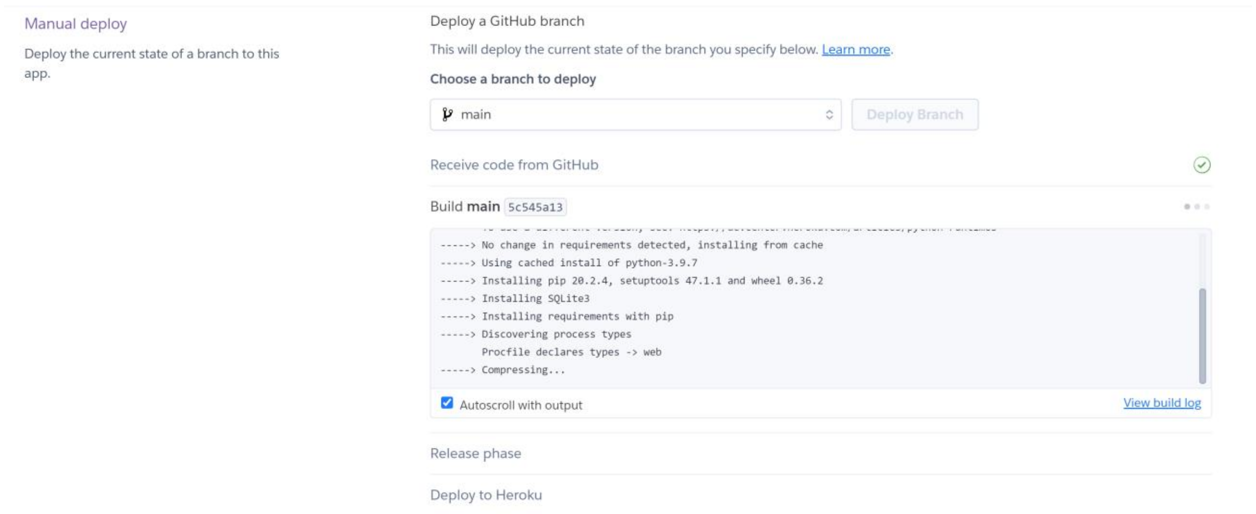
Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

main

You will see all the required packages been installed like the following screenshot:



Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

main

Receive code from GitHub ✓

Build **main** 5c545a13

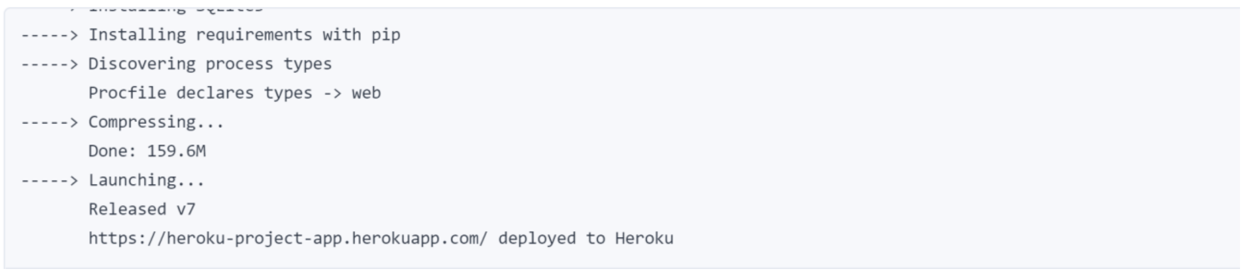
```
-----> No change in requirements detected, installing from cache
-----> Using cached install of python-3.9.7
-----> Installing pip 20.2.4, setuptools 47.1.1 and wheel 0.36.2
-----> Installing SQLite3
-----> Installing requirements with pip
-----> Discovering process types
       Procfile declares types -> web
-----> Compressing...
```

☑ Autoscroll with output

Release phase

Deploy to Heroku

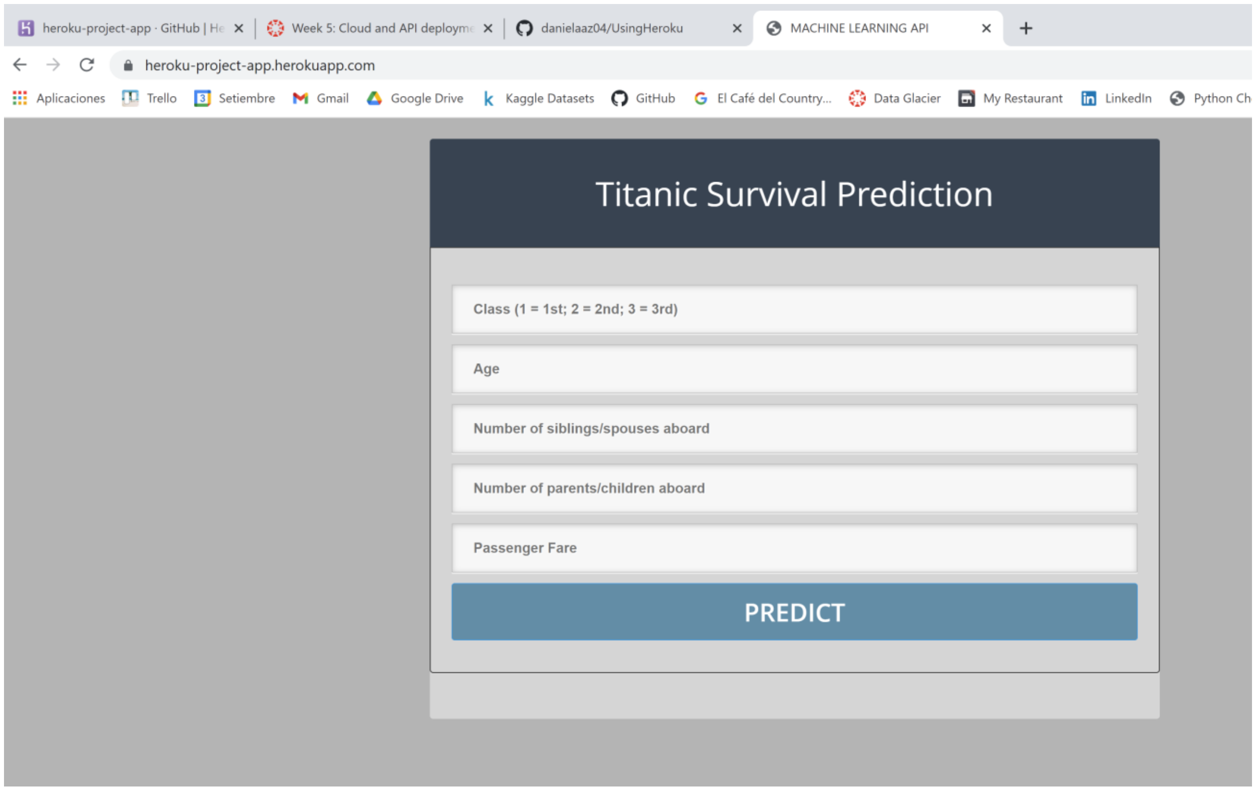
When it finishes, it should look like the following screenshot and you copy the app link on the browser to test the prediction web app.



```
-----> Installing requirements with pip
-----> Discovering process types
       Procfile declares types -> web
-----> Compressing...
       Done: 159.6M
-----> Launching...
       Released v7
       https://heroku-project-app.herokuapp.com/ deployed to Heroku
```

STEP 9

Paste the copied link on your browser and test your app 😊


heroku-project-app · GitHub | He
Week 5: Cloud and API deployme
danielaaz04/UsingHeroku
MACHINE LEARNING API
heroku-project-app.herokuapp.com
Aplicaciones
Trello
Setiembre
Gmail
Google Drive
Kaggle Datasets
GitHub
El Café del Country...
Data Glacier
My Restaurant
LinkedIn
Python Ch
Titanic Survival Prediction
Class (1 = 1st; 2 = 2nd; 3 = 3rd)
Age
Number of siblings/spouses aboard
Number of parents/children aboard
Passenger Fare
PREDICT

(You can find my app at : https://heroku-project-app.herokuapp.com/)