Name: Roger Pineda

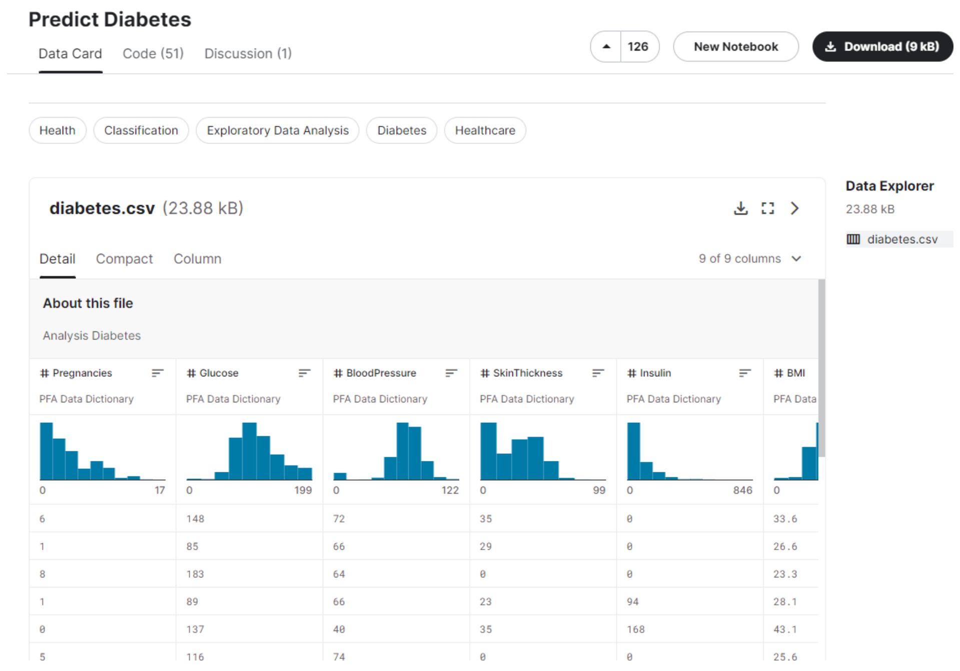
Batch Code: LISUM16

Submission Date: January 5th, 2023

Submitted To: Data Glacier

Deployment on Heroku

Task 1: Go out and find simple data for modeling



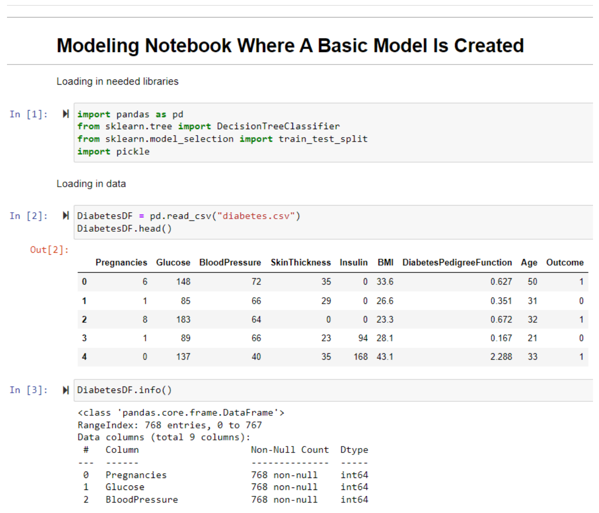
diabetes.csv (23.88 kB)

About this file

Analysis Diabetes

# Pregnancies	# Glucose	# BloodPressure	# SkinThickness	# Insulin	# BMI
PFA Data Dictionary	PFA Data Dictionary	PFA Data Dictionary	PFA Data Dictionary	PFA Data Dictionary	PFA Data
0 – 17	0 – 199	0 – 122	0 – 99	0 – 846	0
6	148	72	35	0	33.6
1	85	66	29	0	26.6
8	183	64	0	0	23.3
1	89	66	23	94	28.1
0	137	40	35	168	43.1
5	116	74	0	0	25.6

Task 2: Creating a predictive model using a machine learning algorithm. Data at hand is predictive whether someone is diabetic or not. A Decision Tree Classifier Algorithm will be used. Save model in pickle file



Modeling Notebook Where A Basic Model Is Created

Loading in needed libraries

```
In [1]:  import pandas as pd
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.model_selection import train_test_split
         import pickle
```

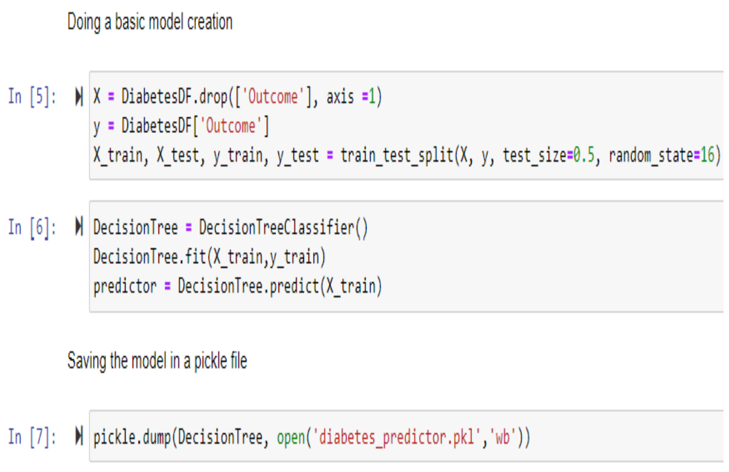
Loading in data

```
In [2]:  DiabetesDF = pd.read_csv("diabetes.csv")
         DiabetesDF.head()
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
In [3]:  DiabetesDF.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
```



Doing a basic model creation

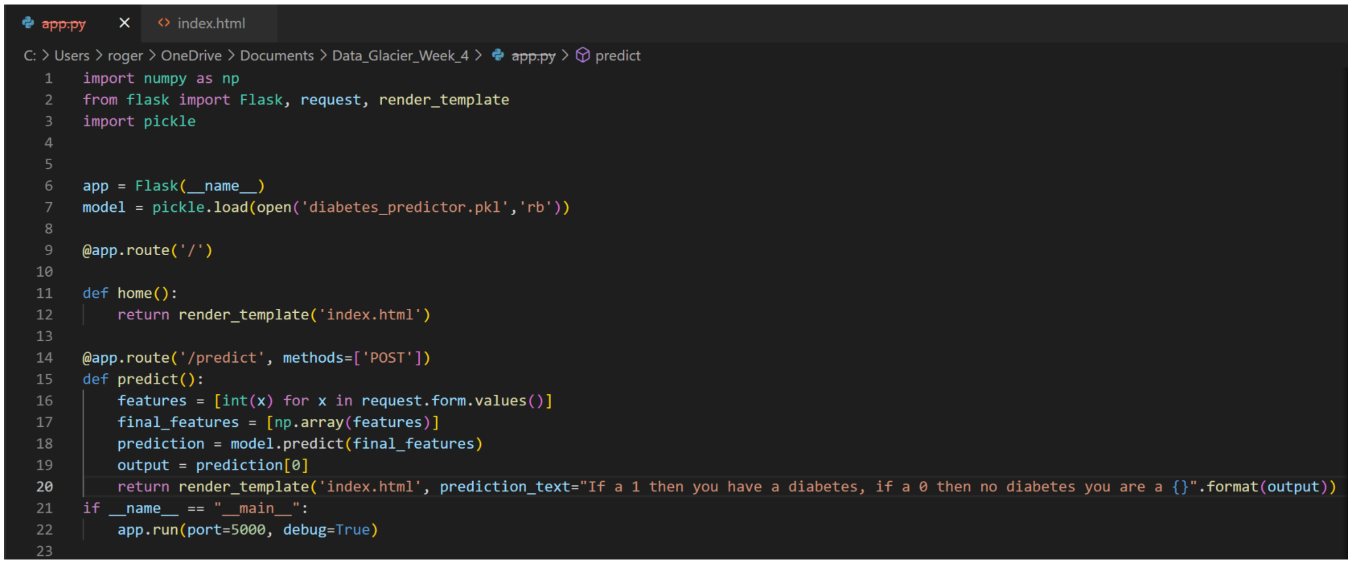
```
In [5]:  X = DiabetesDF.drop(['Outcome'], axis =1)
         y = DiabetesDF['Outcome']
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=16)
```

```
In [6]:  DecisionTree = DecisionTreeClassifier()
         DecisionTree.fit(X_train,y_train)
         predictor = DecisionTree.predict(X_train)
```

Saving the model in a pickle file

```
In [7]:  pickle.dump(DecisionTree, open('diabetes_predictor.pkl','wb'))
```

Task 3: Deploying the model using Flask(Using VS Code). It is routed using a local host at port number 5000.

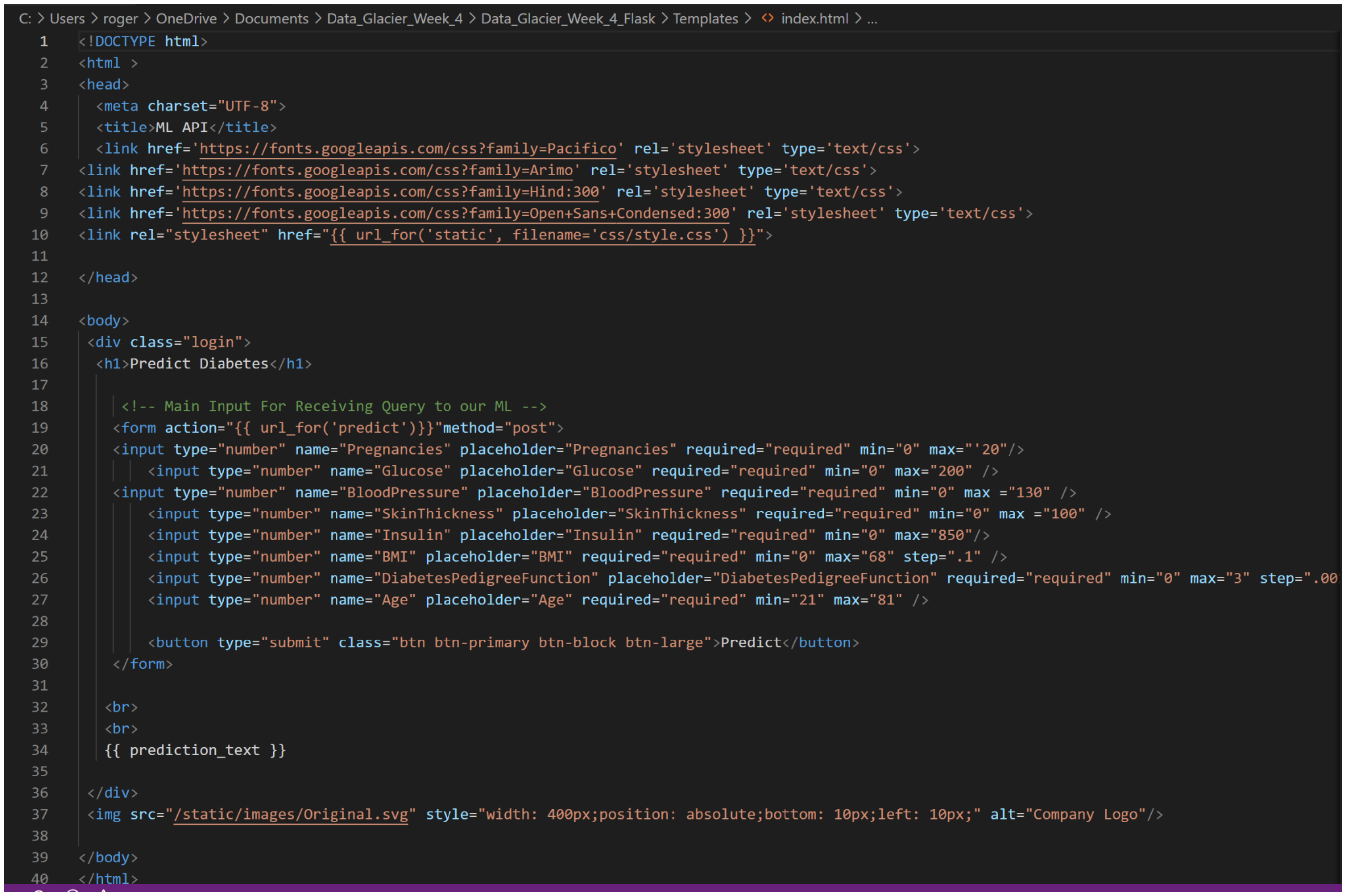


app.py × index.html

C: > Users > roger > OneDrive > Documents > Data_Glacier_Week_4 > app.py > predict

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


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

@app.route('/')

def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    features = [int(x) for x in request.form.values()]
    final_features = [np.array(features)]
    prediction = model.predict(final_features)
    output = prediction[0]
    return render_template('index.html', prediction_text="If a 1 then you have a diabetes, if a 0 then no diabetes you are a {}".format(output))
if __name__ == "__main__":
    app.run(port=5000, debug=True)

```

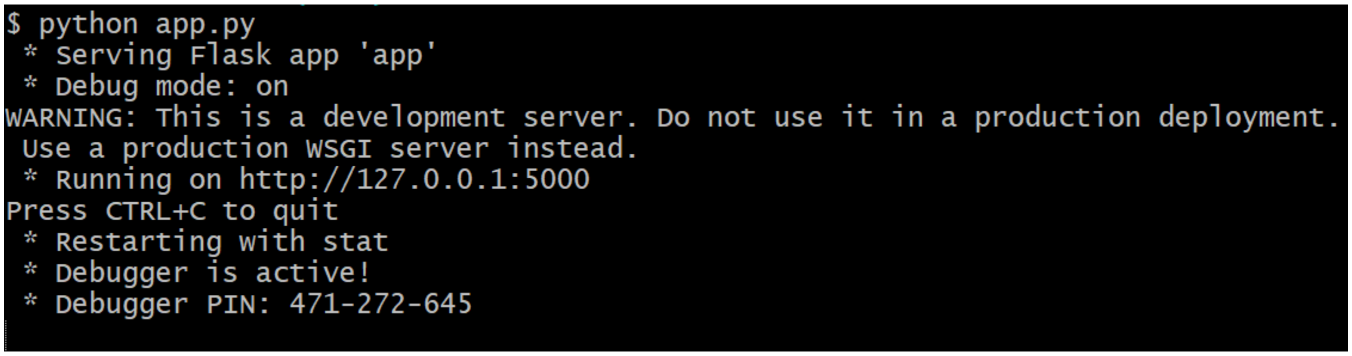
Task 3.1: Generating an html template for Web App. The index.html file is implemented in the app.py file in the home function.



C: > Users > roger > OneDrive > Documents > Data_Glacier_Week_4 > Data_Glacier_Week_4_Flask > Templates > index.html > ...

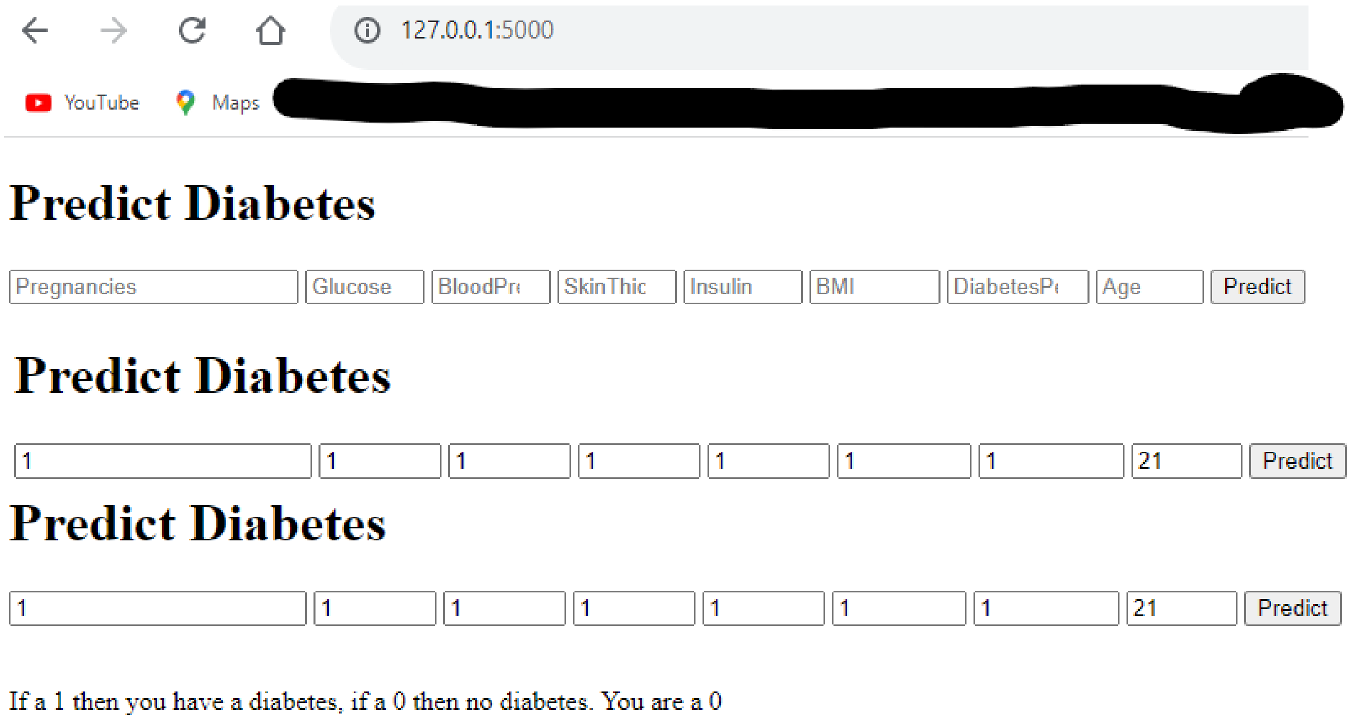
```html
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
  <h1>Predict Diabetes</h1>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
    <input type="number" name="Pregnancies" placeholder="Pregnancies" required="required" min="0" max="'20"/>
        <input type="number" name="Glucose" placeholder="Glucose" required="required" min="0" max="200" />
    <input type="number" name="BloodPressure" placeholder="BloodPressure" required="required" min="0" max ="130" />
        <input type="number" name="SkinThickness" placeholder="SkinThickness" required="required" min="0" max ="100" />
        <input type="number" name="Insulin" placeholder="Insulin" required="required" min="0" max="850"/>
        <input type="number" name="BMI" placeholder="BMI" required="required" min="0" max="68" step=".1" />
        <input type="number" name="DiabetesPedigreeFunction" placeholder="DiabetesPedigreeFunction" required="required" min="0" max="3" step=".00
        <input type="number" name="Age" placeholder="Age" required="required" min="21" max="81" />

        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>
 <img src="/static/images/Original.svg" style="width: 400px;position: absolute;bottom: 10px;left: 10px;" alt="Company Logo"/>

</body>
</html>
```

Task 4: Open the App



```
$ python 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://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 471-272-645
```

Task 5: Generating Predictions



127.0.0.1:5000

YouTube Maps [redacted]

Predict Diabetes

Pregnancies | Glucose | BloodPr | SkinThic | Insulin | BMI | DiabetesP | Age | Predict

Predict Diabetes

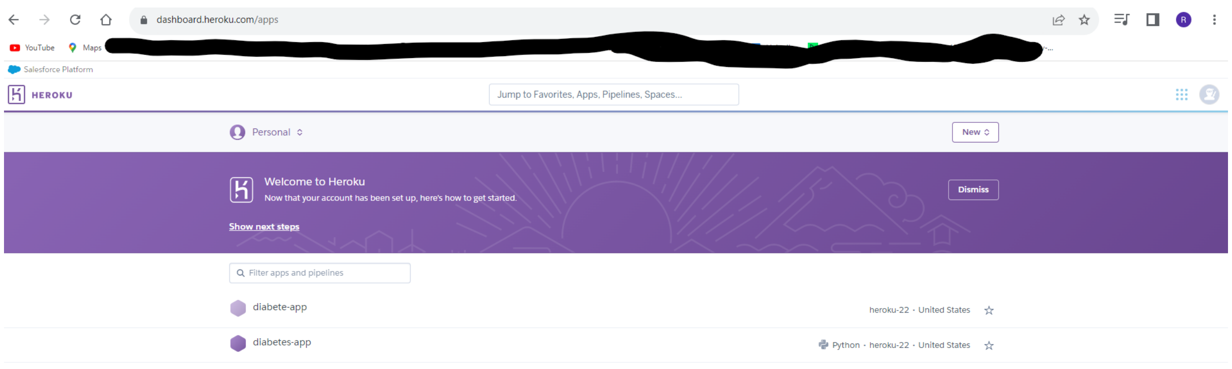
1 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | Predict

Predict Diabetes

1 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | Predict

If a 1 then you have a diabetes, if a 0 then no diabetes. You are a 0

Task 5: Create a Heroku Account



dashboard.heroku.com/apps

YouTube Maps [redacted]

Salesforce Platform

HEROKU Jump to Favorites, Apps, Pipelines, Spaces...

Personal New

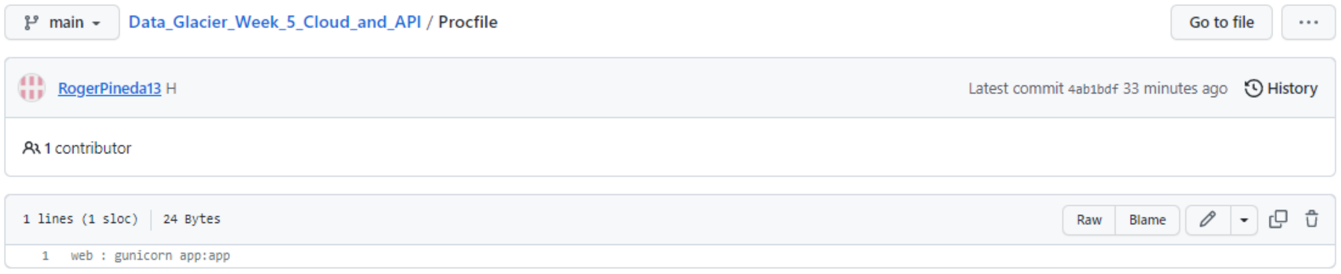
Welcome to Heroku
Now that your account has been set up, here's how to get started.
Show next steps
Dismiss

Filter apps and pipelines

diabete-app heroku-22 · United States

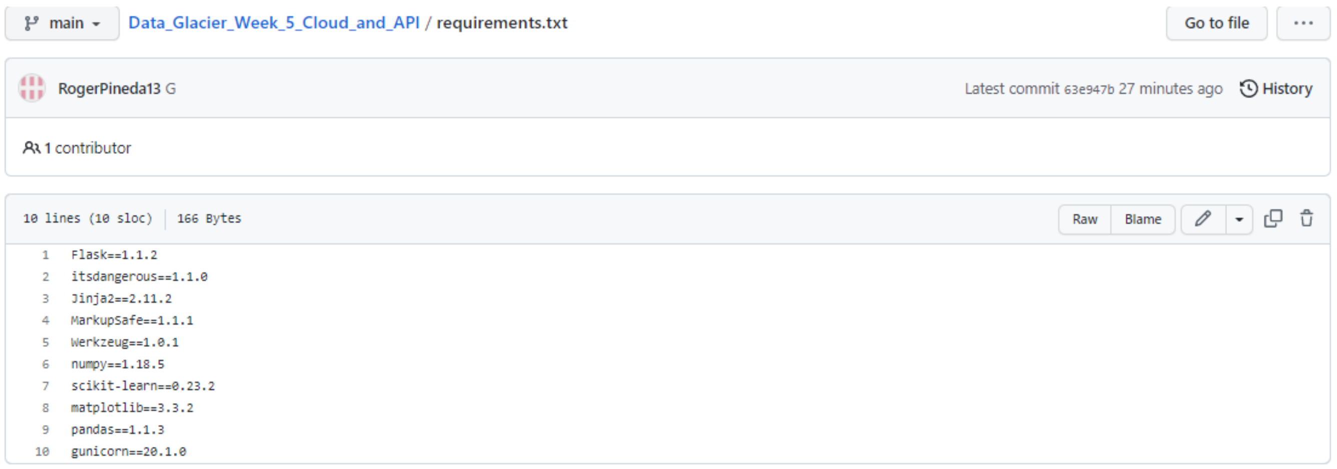
diabetes-app Python · heroku-22 · United States

Task 6: Create Required text files for Heroku Deployment. Files are the Procfile with start up **web: gunicorn app:app** and requirements that was generated using pip freeze and needing to create a runtime start file to use older version of python.


main
Data_Glacier_Week_5_Cloud_and_API / Procfile
Go to file
RogerPineda13 H
Latest commit 4ab1bdf 33 minutes ago
History
1 contributor
Raw
Blame

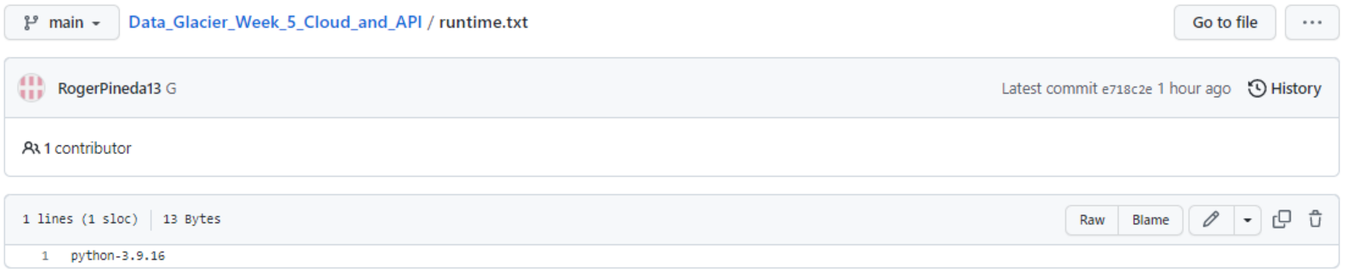
1 lines (1 sloc) | 24 Bytes

```
web : gunicorn app:app
```


main
Data_Glacier_Week_5_Cloud_and_API / requirements.txt
Go to file
RogerPineda13 G
Latest commit 63e947b 27 minutes ago
History
1 contributor
Raw
Blame

10 lines (10 sloc) | 166 Bytes

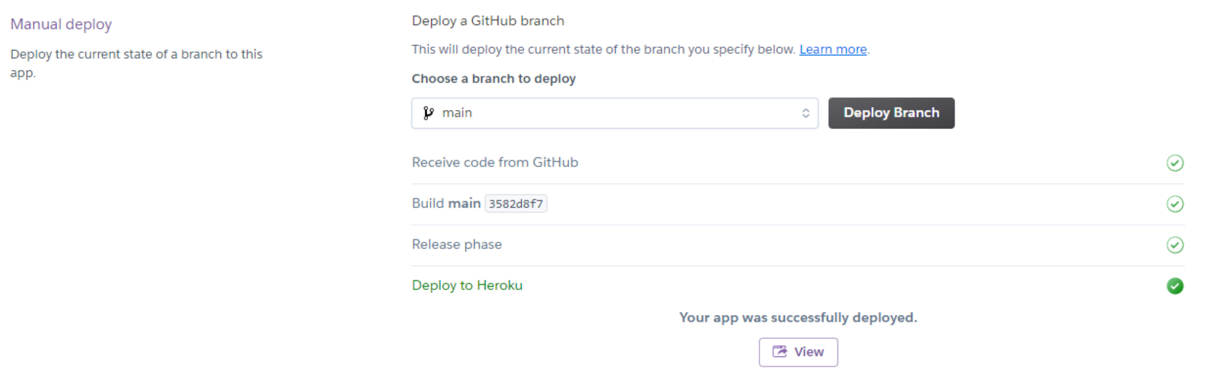
```
Flask==1.1.2
itsdangerous==1.1.0
Jinja2==2.11.2
MarkupSafe==1.1.1
Werkzeug==1.0.1
numpy==1.18.5
scikit-learn==0.23.2
matplotlib==3.3.2
pandas==1.1.3
gunicorn==20.1.0
```


main
Data_Glacier_Week_5_Cloud_and_API / runtime.txt
Go to file
RogerPineda13 G
Latest commit e718c2e 1 hour ago
History
1 contributor
Raw
Blame

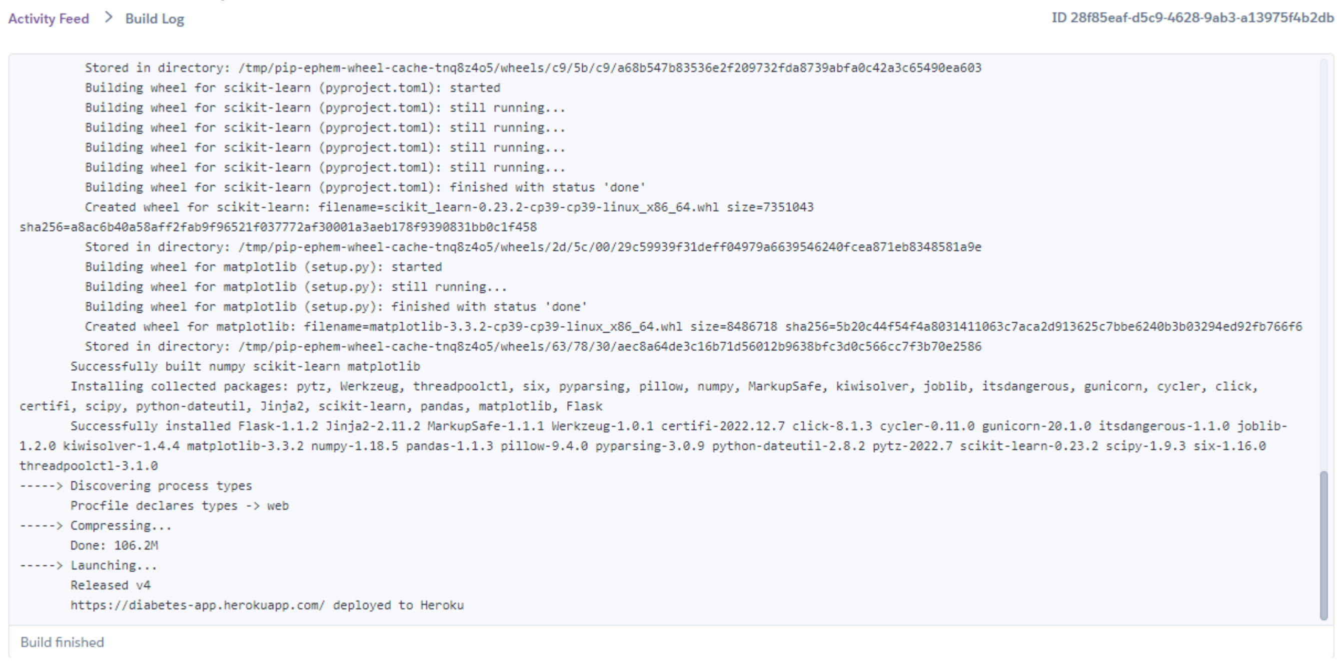
1 lines (1 sloc) | 13 Bytes

```
python-3.9.16
```

Task 7: Connect GitHub Repository to Heroku


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
Deploy Branch
Receive code from GitHub
Build main 3582d8f7
Release phase
Deploy to Heroku
Your app was successfully deployed.
View

Task 8: Run through Heroku



Activity Feed > Build Log

ID 28f85eaf-d5c9-4628-9ab3-a13975f4b2db

```
         Stored in directory: /tmp/pip-ephem-wheel-cache-tnq8z4o5/wheels/c9/5b/c9/a68b547b83536e2f209732fda8739abfa0c42a3c65490ea603
         Building wheel for scikit-learn (pyproject.toml): started
         Building wheel for scikit-learn (pyproject.toml): still running...
         Building wheel for scikit-learn (pyproject.toml): still running...
         Building wheel for scikit-learn (pyproject.toml): still running...
         Building wheel for scikit-learn (pyproject.toml): still running...
         Building wheel for scikit-learn (pyproject.toml): finished with status 'done'
         Created wheel for scikit-learn: filename=scikit_learn-0.23.2-cp39-cp39-linux_x86_64.whl size=7351043
sha256=a8ac6b40a58aff2fab9f96521f037772af30001a3aeb178f9390831bb0c1f458
         Stored in directory: /tmp/pip-ephem-wheel-cache-tnq8z4o5/wheels/2d/5c/00/29c59939f31deff04979a6639546240fcea871eb8348581a9e
         Building wheel for matplotlib (setup.py): started
         Building wheel for matplotlib (setup.py): still running...
         Building wheel for matplotlib (setup.py): finished with status 'done'
         Created wheel for matplotlib: filename=matplotlib-3.3.2-cp39-cp39-linux_x86_64.whl size=8486718 sha256=5b20c44f54f4a8031411063c7aca2d913625c7bbe6240b3b03294ed92fb766f6
         Stored in directory: /tmp/pip-ephem-wheel-cache-tnq8z4o5/wheels/63/78/30/aec8a64de3c16b71d56012b9638bfc3d0c566cc7f3b70e2586
       Successfully built numpy scikit-learn matplotlib
       Installing collected packages: pytz, Werkzeug, threadpoolctl, six, pyparsing, pillow, numpy, MarkupSafe, kiwisolver, joblib, itsdangerous, gunicorn, cycler, click,
certifi, scipy, python-dateutil, Jinja2, scikit-learn, pandas, matplotlib, Flask
       Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 certifi-2022.12.7 click-8.1.3 cycler-0.11.0 gunicorn-20.1.0 itsdangerous-1.1.0 joblib-
1.2.0 kiwisolver-1.4.4 matplotlib-3.3.2 numpy-1.18.5 pandas-1.1.3 pillow-9.4.0 pyparsing-3.0.9 python-dateutil-2.8.2 pytz-2022.7 scikit-learn-0.23.2 scipy-1.9.3 six-1.16.0
threadpoolctl-3.1.0
-----> Discovering process types
       Procfile declares types -> web
-----> Compressing...
       Done: 106.2M
-----> Launching...
       Released v4
       https://diabetes-app.herokuapp.com/ deployed to Heroku
```

Build finished

https://diabetes-app.herokuapp.com