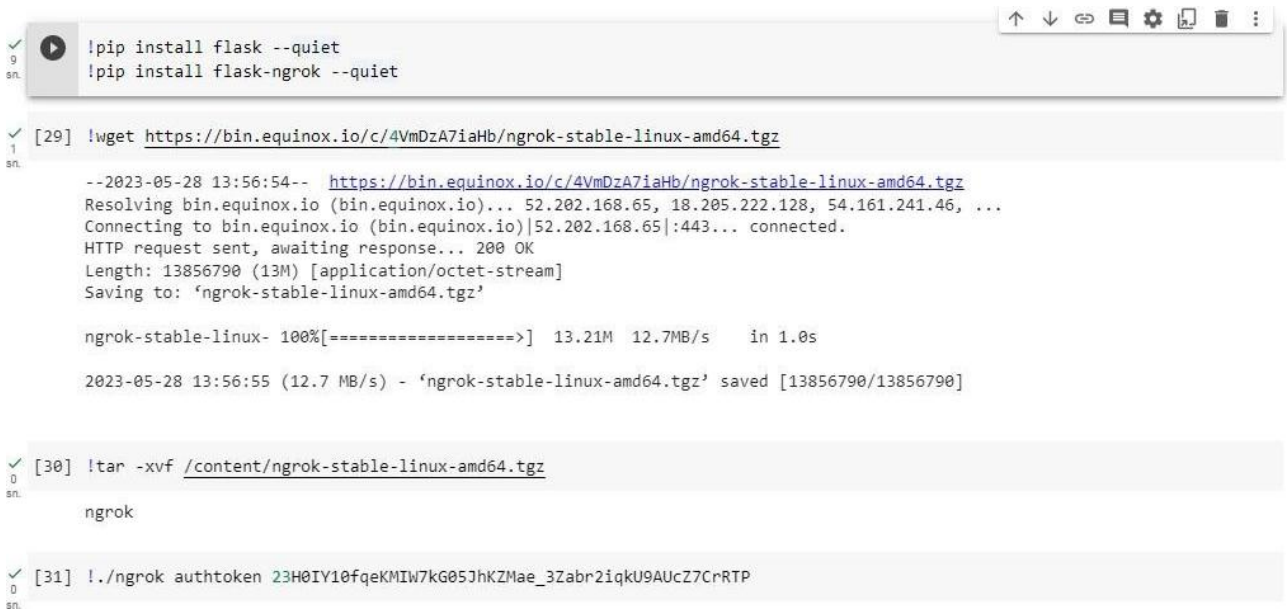
Week 5: Cloud and API Deployment

Name: Cloud and API deployment
Report date: 04-June-2023
Internship Batch: LISUM21
Version:1.0
Data intake by: Ece Yavuzyılmaz
Data intake reviewer : Data Glacier
Data storage location:
https://github.com/eceyy/Data_Glacier_Intership_2023/tree/main/Week%205/workflows

Tabular data details:

Total number of observations	5000
Total number of files	1
Total number of features	7
Base format of the file	csv
Size of the data	709 KB

Step 1: Flask API



```
!pip install flask --quiet
!pip install flask-ngrok --quiet
```

```
[29] !wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.tgz

--2023-05-28 13:56:54--  https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.tgz
Resolving bin.equinox.io (bin.equinox.io)... 52.202.168.65, 18.205.222.128, 54.161.241.46, ...
Connecting to bin.equinox.io (bin.equinox.io)|52.202.168.65|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13856790 (13M) [application/octet-stream]
Saving to: 'ngrok-stable-linux-amd64.tgz'

ngrok-stable-linux- 100%[===================>]  13.21M  12.7MB/s    in 1.0s

2023-05-28 13:56:55 (12.7 MB/s) - 'ngrok-stable-linux-amd64.tgz' saved [13856790/13856790]
```

```
[30] !tar -xvf /content/ngrok-stable-linux-amd64.tgz

ngrok
```

```
[31] !./ngrok authtoken 23H0IY10fqeKMIW7kG05JhKZMae_3Zabr2iqkU9AUcZ7CrRTP
```

```
# import Flask from flask module
from flask import Flask

# import run_with_ngrok from flask_ngrok to run the app using ngrok
from flask_ngrok import run_with_ngrok
from flask import Flask, request, render_template
app = Flask(__name__) #app name
run_with_ngrok(app)

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

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

#Set a post method to yield predictions on page
@app.route('/', methods = ['POST'])
def predict():

    #obtain all form values and place them in an array, convert into integers
    int_features = [int(x) for x in request.form.values()]
    #Combine them all into a final numpy array
    final_features = [np.array(int_features)]
    #predict the price given the values inputted by user
    prediction = model.predict(final_features)
```

```
[37]
    #Round the output to 2 decimal places
    output = round(prediction[0], 2)

    #If the output is negative, the values entered are unreasonable to the context of the application
    #If the output is greater than 0, return prediction
    if output < 0:
        return render_template('index.html', prediction_text = "Predicted Price is negative, values entered not reasonable
    elif output >= 0:
        return render_template('index.html', prediction_text = 'Predicted Price of the house is: ${}'.format(output))

#Run app
if __name__ == "__main__":
    app.run()
```

```
 * Serving Flask app '__main__'
 * Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
 * Running on http://ce5d-34-147-48-184.ngrok-free.app
 * Traffic stats available on http://127.0.0.1:4040
INFO:werkzeug:127.0.0.1 - - [28/May/2023 13:57:33] "GET / HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [28/May/2023 13:57:37] "GET /favicon.ico HTTP/1.1" 404 -
```

Step 2. Requirement.txt

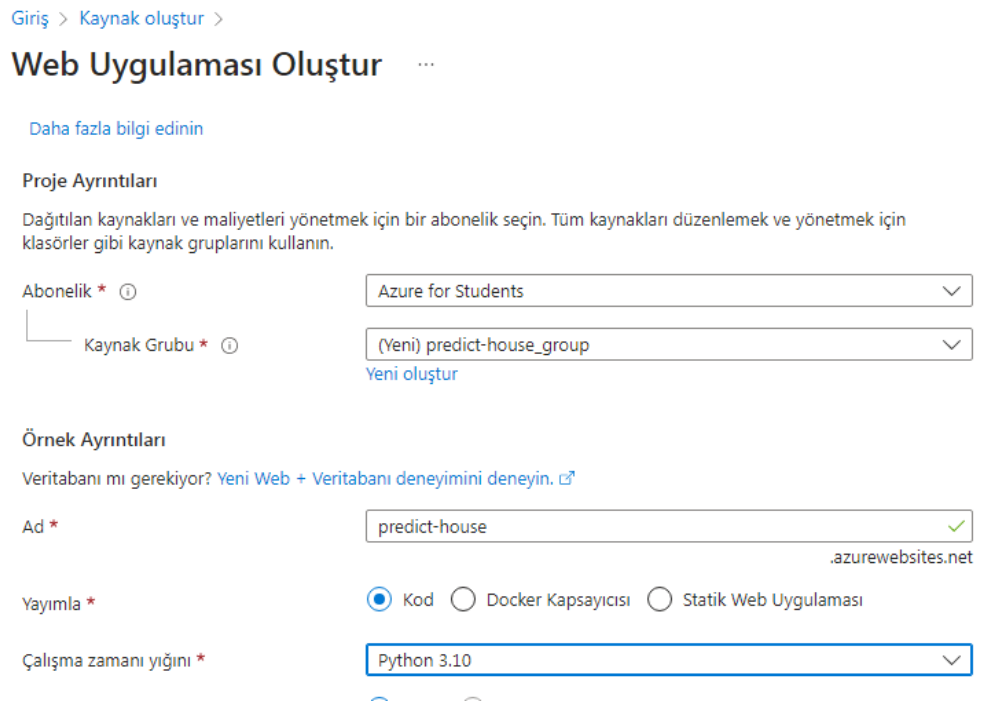
```
[ ] !pip install pipreqs
```

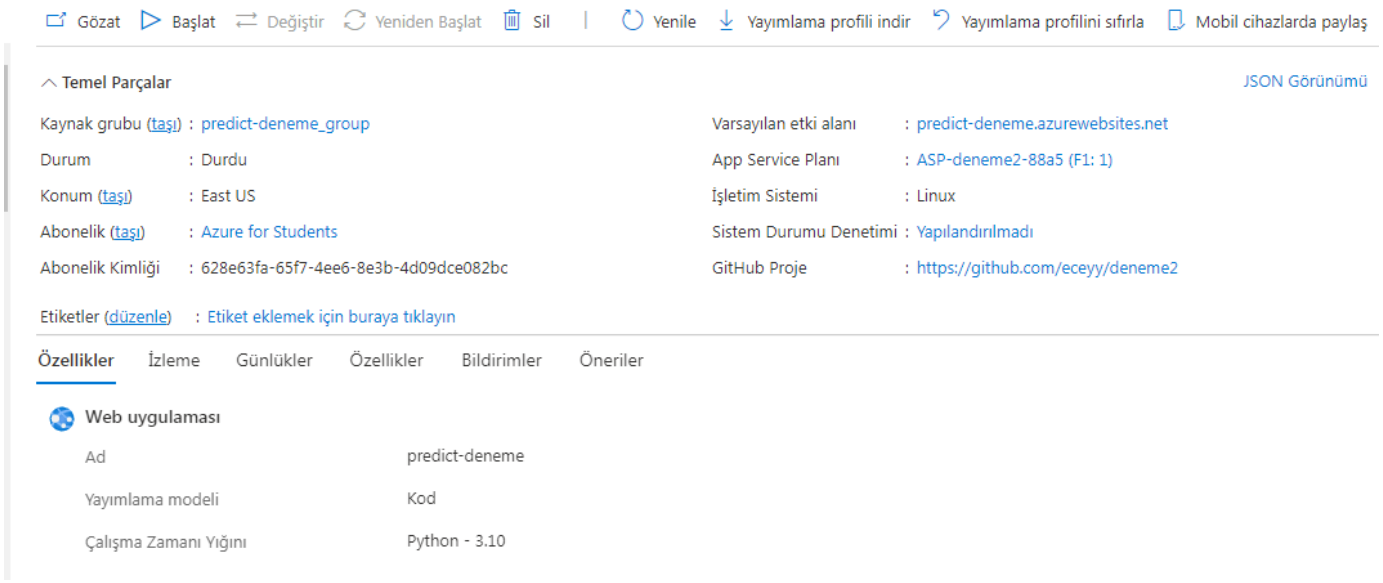
```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pipreqs
  Downloading pipreqs-0.4.13-py2.py3-none-any.whl (33 kB)
Collecting docopt (from pipreqs)
  Downloading docopt-0.6.2.tar.gz (25 kB)
  Preparing metadata (setup.py) ... done
Collecting yarg (from pipreqs)
  Downloading yarg-0.1.9-py2.py3-none-any.whl (19 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from yarg->pipreqs) (2.27.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->yarg->pipreqs) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->yarg->pipreqs) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->yarg->pipreqs) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->yarg->pipreqs) (3.4)
Building wheels for collected packages: docopt
  Building wheel for docopt (setup.py) ... done
  Created wheel for docopt: filename=docopt-0.6.2-py2.py3-none-any.whl size=13707 sha256=60d77673781f9dd856ab0e05517f4c9114b674c04ec28610917dbf023a4dd349
  Stored in directory: /root/.cache/pip/wheels/fc/ab/d4/5da2067ac95b36618c629a5f93f809425700506f72c9732fac
Successfully built docopt
Installing collected packages: docopt, yarg, pipreqs
Successfully installed docopt-0.6.2 pipreqs-0.4.13 yarg-0.1.9
```

```
[ ] !pipreqs .
```

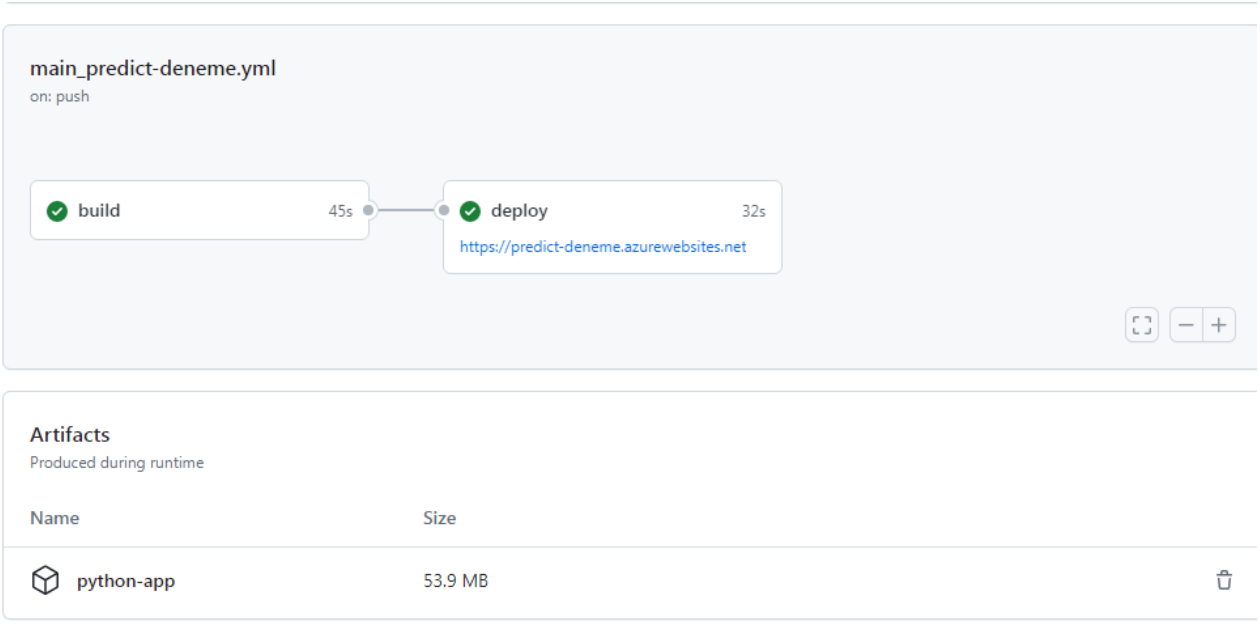
```
INFO: Successfully saved requirements file in ./requirements.txt
```

Step 3. Create a new app in Azure





Step 4. Finish the build and deploy it



Code | Blame | 63 lines (50 loc) · 1.71 KB | Raw

```yaml
on:
  push:
    branches:
      - main
  workflow_dispatch:

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v3

      - name: Set up Python version
        uses: actions/setup-python@v3
        with:
          python-version: '3.10'

      - name: Create and start virtual environment
        run: |
          python -m venv venv
          source venv/bin/activate

      - name: Install dependencies
        run: pip install -r requirements.txt


      - name: Upload artifact for deployment jobs
        uses: actions/upload-artifact@v3
        with:
          name: python-app
          path: |
            .
            !venv/

  deploy:
    runs-on: ubuntu-latest
    needs: build
    environment:
      name: 'Production'
      url: ${{ steps.deploy-to-webapp.outputs.webapp-url }}

    steps:
      - name: Download artifact from build job
        uses: actions/download-artifact@v3
        with:
          name: python-app
          path: .

      - name: 'Deploy to Azure Web App'
        uses: azure/webapps-deploy@v2
        id: deploy-to-webapp
        with:
```

Step 5. Testing the model