

Name: **Cloud and API deployment**


Submission date: **4-July-2022**

Internship Batch: **LISUM10**

Data intake and model creation by: **Efe KARASIL**

Submitted to: **<https://github.com/ekarasil/DataGlacier-Week5>**

The model was already set and ready, which was week 4's Machine Learning model. In terms of python codes, the only part that was needed to change was the "flask_app" variable as "app" in the app.py file.



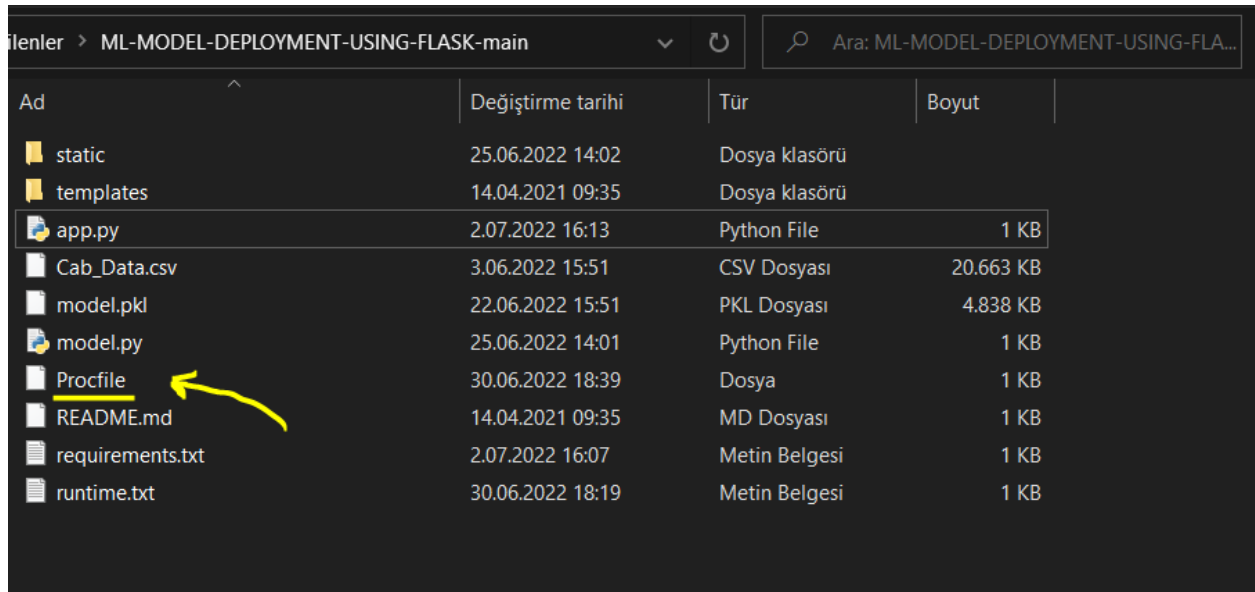
```
app.py X
app.py > ...
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4
5 # Create flask app
6 flask_app = Flask(__name__)
7 model = pickle.load(open("model.pkl", "rb"))
8
9 @flask_app.route("/")
10 def Home():
11     return render_template("index.html")
12
13 @flask_app.route("/predict", methods = ["POST"])
14 def predict():
15     float_features = [float(x) for x in request.form.values()]
16     features = [np.array(float_features)]
17     prediction = model.predict(features)
18     return render_template("index.html", prediction_text = "The price charge for cab usage is {}".format(prediction))
19
20 if __name__ == "__main__":
21     flask_app.run(debug=True)
```

Before



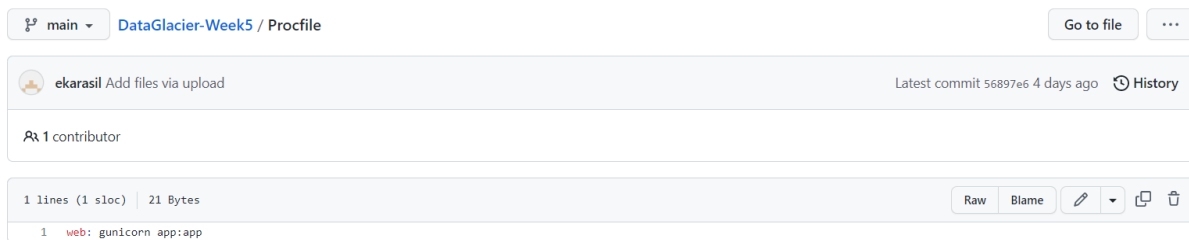
```
app.py X
app.py > app
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4
5 # Create flask app
6 app = Flask(__name__)
7 model = pickle.load(open("model.pkl", "rb"))
8
9 @app.route("/")
10 def Home():
11     return render_template("index.html")
12
13 @app.route("/predict", methods = ["POST"])
14 def predict():
15     float_features = [float(x) for x in request.form.values()]
16     features = [np.array(float_features)]
17     prediction = model.predict(features)
18     return render_template("index.html", prediction_text = "The price charge for cab usage is {}".format(prediction))
19
20 if __name__ == "__main__":
21     app.run(debug=True)
```

After



Ad	Değiştirme tarihi	Tür	Boyut
static	25.06.2022 14:02	Dosya klasörü	
templates	14.04.2021 09:35	Dosya klasörü	
app.py	2.07.2022 16:13	Python File	1 KB
Cab_Data.csv	3.06.2022 15:51	CSV Dosyası	20.663 KB
model.pkl	22.06.2022 15:51	PKL Dosyası	4.838 KB
model.py	25.06.2022 14:01	Python File	1 KB
Procfile	30.06.2022 18:39	Dosya	1 KB
README.md	14.04.2021 09:35	MD Dosyası	1 KB
requirements.txt	2.07.2022 16:07	Metin Belgesi	1 KB
runtime.txt	30.06.2022 18:19	Metin Belgesi	1 KB

“Procfile” named file needed to be created with no extension. You can edit it with a notepad, or other text editor app, you need to write “web: gunicorn app:app” inside of it.



```
pip freeze > requirements.txt
```

This is the command that gives us our requirements.txt file. You should be in the same file directory with your project and application files in your command tab. This is a very important file that not only guides other developers on the version of the python packages you have used in your app , but it is also a file that Heroku uses to run your app during deployment. Your requirements file should look something like the image below. A list of your packages and the version. Heroku might bring an error with versions during deployment and you can always downgrade your package versions.

Also, if it is not already on the “requirements.txt” file add “gunicorn” to that file as well.

27 lines (27 sloc) | 439 Bytes

```
1  atomicwrites==1.4.0
2  attrs==21.4.0
3  click==8.1.3
4  colorama==0.4.4
5  Flask==2.1.2
6  iniconfig==1.1.1
7  itsdangerous==2.1.2
8  Jinja2==3.1.2
9  joblib==1.1.0
10 MarkupSafe==2.1.1
11 numpy==1.22.4
12 packaging==21.3
13 pandas==1.4.2
14 pluggy==1.0.0
15 py==1.11.0
16 pyparsing==3.0.9
17 pytest==7.1.2
18 python-dateutil==2.8.2
19 pytz==2022.1
20 scikit-learn==1.1.1
21 scipy==1.8.1
22 six==1.16.0
23 sklearn==0.0
24 threadpoolctl==3.1.0
25 tomli==2.0.1
26 Werkzeug==2.1.2
27 gunicorn
```

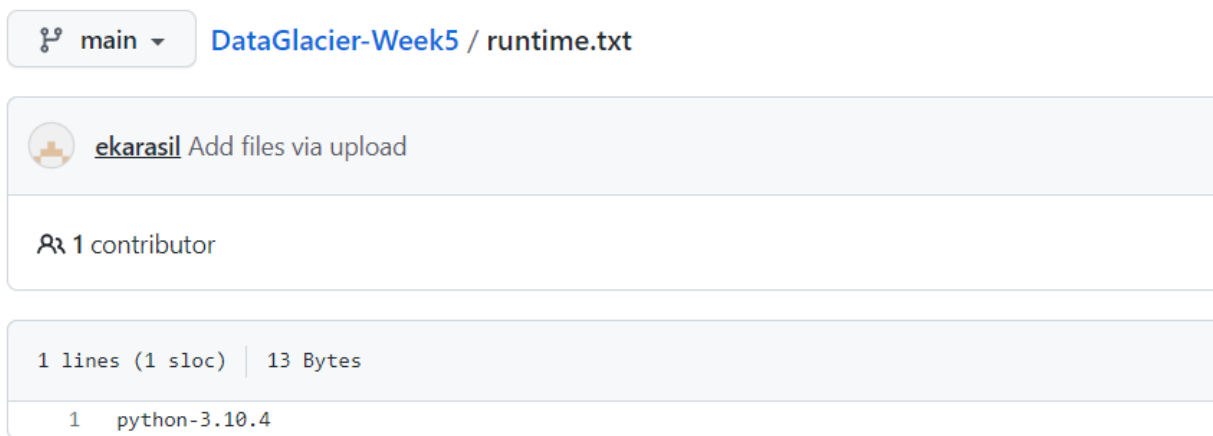
“requirements.txt”

The last file you need to create is again a txt file, which is named “runtime.txt”.

```
C:\Users\HP>python -V
Python 3.10.4

C:\Users\HP>
```

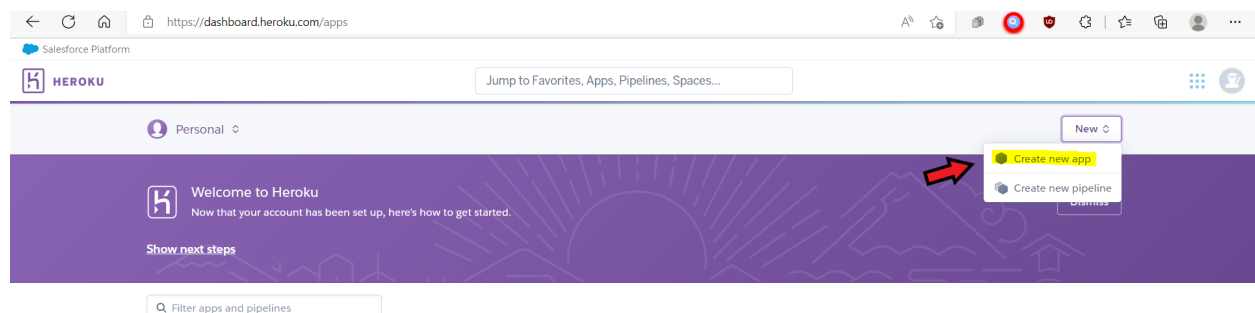
You need to type your python version as in the below screenshot, you can learn it from your command prompt with “python -V” command.



<https://github.com/ekarasil/DataGlacier-Week5>

All files can be seen from this link. And also the easiest way to use Heroku is using github connection with Heroku. So, I preferred linking my github repository as well when using Heroku.

After creation of your Heroku membership etc., you can follow the steps below;



← ↻ 🏠 🔒 https://dashboard.heroku.com/new-app

Salesforce Platform

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



Create New App

App name
ml-model-flask-ekarasil

Choose a region
Europe

Add to pipeline...

Create app

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

Personal > ml-model-flask-ekarasil

GitHub ekarasil/DataGlacier-Week5 main

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 ekarasil/DataGlacier-Week5 by ekarasil [Disconnect...](#)

Releases in the [activity feed](#) link to GitHub to view commit diffs

Automatically deploys from 

As it is mentioned above, choose github connection and verify your github profile etc. then choose your repository to connect as the second step.

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to [@ekarasil/DataGlacier-Week5](#) by [@ekarasil](#)

[Disconnect...](#)

- Releases in the [activity feed](#) link to GitHub to view commit diffs
- Automatically deploys from [main](#)

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

☒ Automatic deploys from [main](#) are enabled

Every push to [main](#) will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. [Learn more](#).

☐ Wait for CI to pass before deploy

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

[Disable Automatic Deploys](#)

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](#)



When you scroll down in the “Deploy” tab, you can deploy your chosen repository manually or you can select an automatic deployment action which deploys your application when it detects change/update in your repository. You can always do it manually as well.

Personal > ml-model-flask-ekarasil

GitHub ekarasil/DataGlacier-Week5 main

Overview

Resources

Deploy

Metrics

Activity

Access

Settings

Installed add-ons \$0.00/month

[Configure Add-ons](#)

There are no add-ons for this app

You can add add-ons to this app and they will show here. [Learn more](#)

Dyno formation \$0.00/month

[Configure Dynos](#)This app is using **free** dynos

web gunicorn app:app

ON

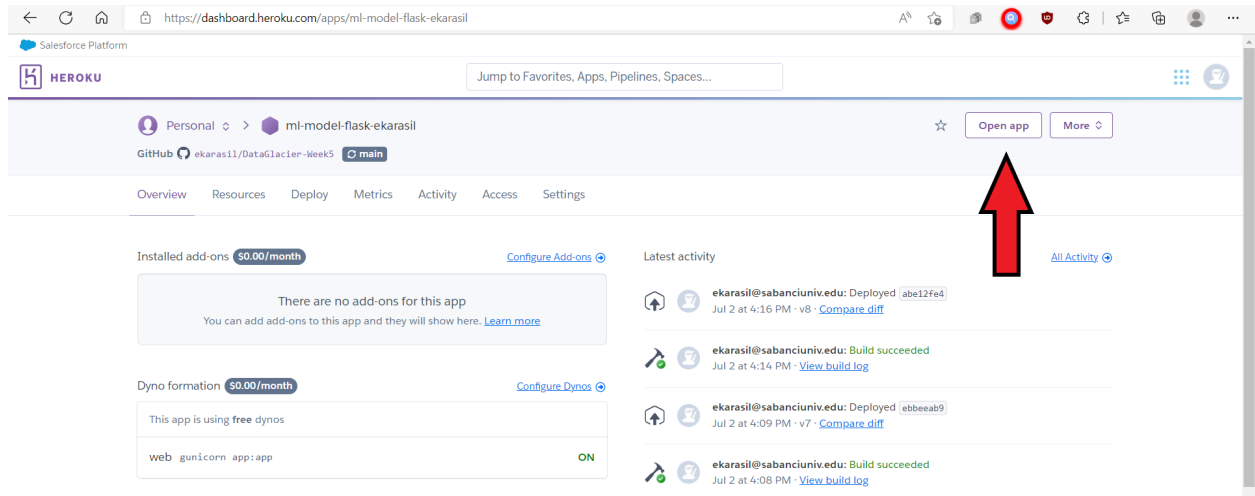
Collaborator activity

[Manage Access](#)

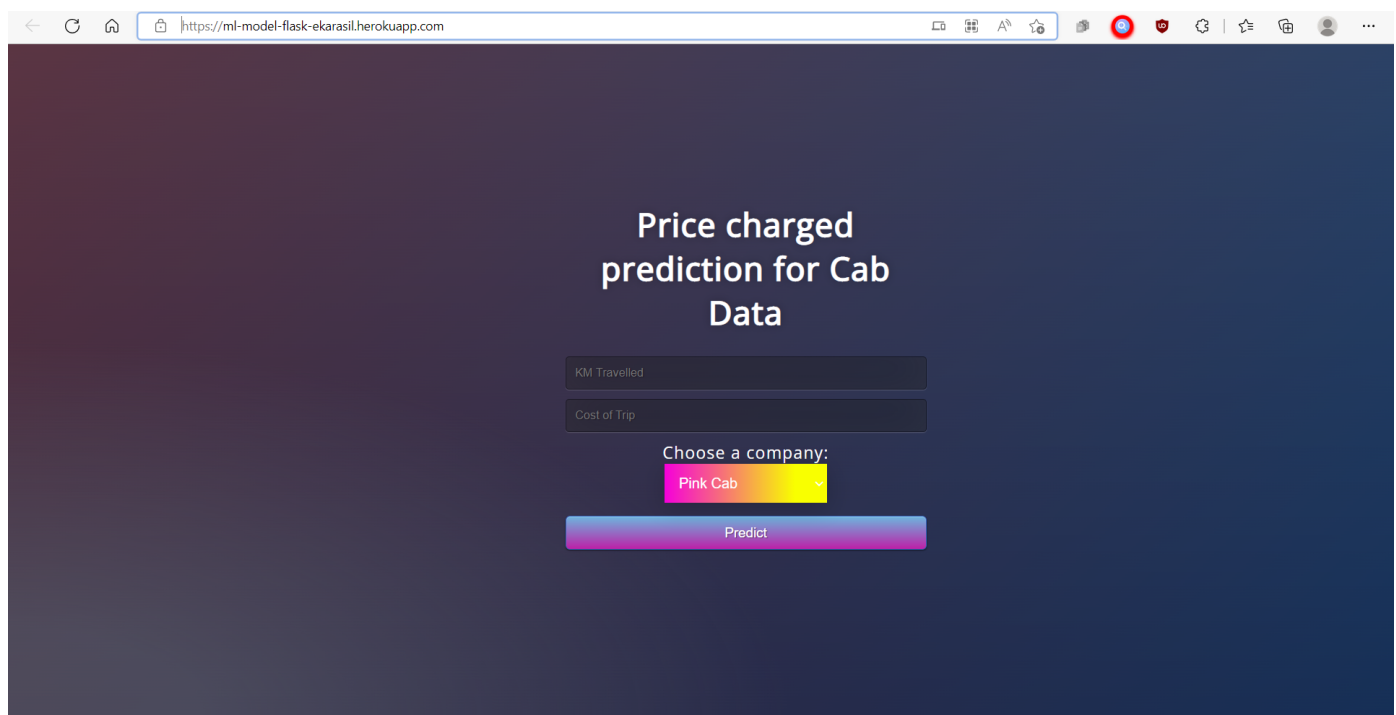
ekarasil@sabanciuniv.edu

6 deploys

After you deploy your application, you need to go to the “Overview” tab and see if Dyno formation says “ON” otherwise go to “Configure Dynos” and make it “ON”.



You can access your API application through the “Open app” button on the upper right side of the page.



<https://ml-model-flask-ekarasil.herokuapp.com/>

It gives a domain with your application name(mine is the above link/url) and “herokuapp.com” you can even use this url link for your access to your API application whenever you want.