



**Data Glacier**

Your Deep Learning Partner

# Credit Risk Prediction Cloud and API deployment

## LISUM10

## By: Mohsen Bahremani

July 1, 2022

[https://github.com/MohsenBah/Credit\\_risk\\_prediction](https://github.com/MohsenBah/Credit_risk_prediction)

<https://credit-risk-webapp.herokuapp.com/>

# Agenda

- ✓ **Goal:** Deploying ML, tested by Postman, on Cloud,
- ✓ **Data:** it has 9 features and 1 response variable: **Age, Sex, Job, Housing, Saving accounts, Checking accounts, Credit amount , Duration, Purpose, Risk** (target value)
- ✓ **Model:** Logistic Regression.
- ✓ **Deployment:** Deploying web app written by Flask API on Heroku.

# Data



Data Glacier

Real Deep Learning Partner

```
In [2]: data=pd.read_csv("/Users/mohsen/Documents/GitHub/Credit_risk_prediction/german_credit_data.csv", index_col=0)  
data
```

Out [2]:

	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose	Risk
0	67	male	2	own	NaN	little	1169	6	radio/TV	good
1	22	female	2	own	little	moderate	5951	48	radio/TV	bad
2	49	male	1	own	little	NaN	2096	12	education	good
3	45	male	2	free	little	little	7882	42	furniture/equipment	good
4	53	male	2	free	little	little	4870	24	car	bad
...	...	...	...	...	...	...	...	...	...	...
995	31	female	1	own	little	NaN	1736	12	furniture/equipment	good
996	40	male	3	own	little	little	3857	30	car	good
997	38	male	2	own	little	NaN	804	12	radio/TV	good
998	23	male	2	free	little	little	1845	45	radio/TV	bad
999	27	male	2	own	moderate	moderate	4576	45	car	good

1000 rows x 10 columns

# Saving (serialization) the final Model

```
params = {
    "C": np.logspace(-3, 3, 7),
    "penalty": ["l1", "l2"]
} # l1 lasso l2 ridge
model = LogisticRegression()
kfold=RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

logreg_cv = GridSearchCV(model, params, cv=kfold)
logreg_cv.fit(X, y)
print("tuned hpyerparameters :(best parameters) ", logreg_cv.best_params_)
print("accuracy :", logreg_cv.best_score_)
```

Logistic Regression has been selected as the final model. The grid search chose l2 penalty.

The model was saved as a **Pickle file**.

```
pickle.dump(logreg_cv, open('model.pkl', 'wb'))
```

# Web App Script



Web app via Flask API was written as shown.

The new test observation is needed to some preprocessing to become ready to predict, including changing the categorical variables to dummies.

```
import numpy as np
import pandas as pd
from flask import Flask, request, render_template
import pickle
import os

app = Flask(__name__)

clf = pickle.load(open('/Users/mohsen/Documents/GitHub/Credit_risk_prediction/model.pkl', 'rb'))

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

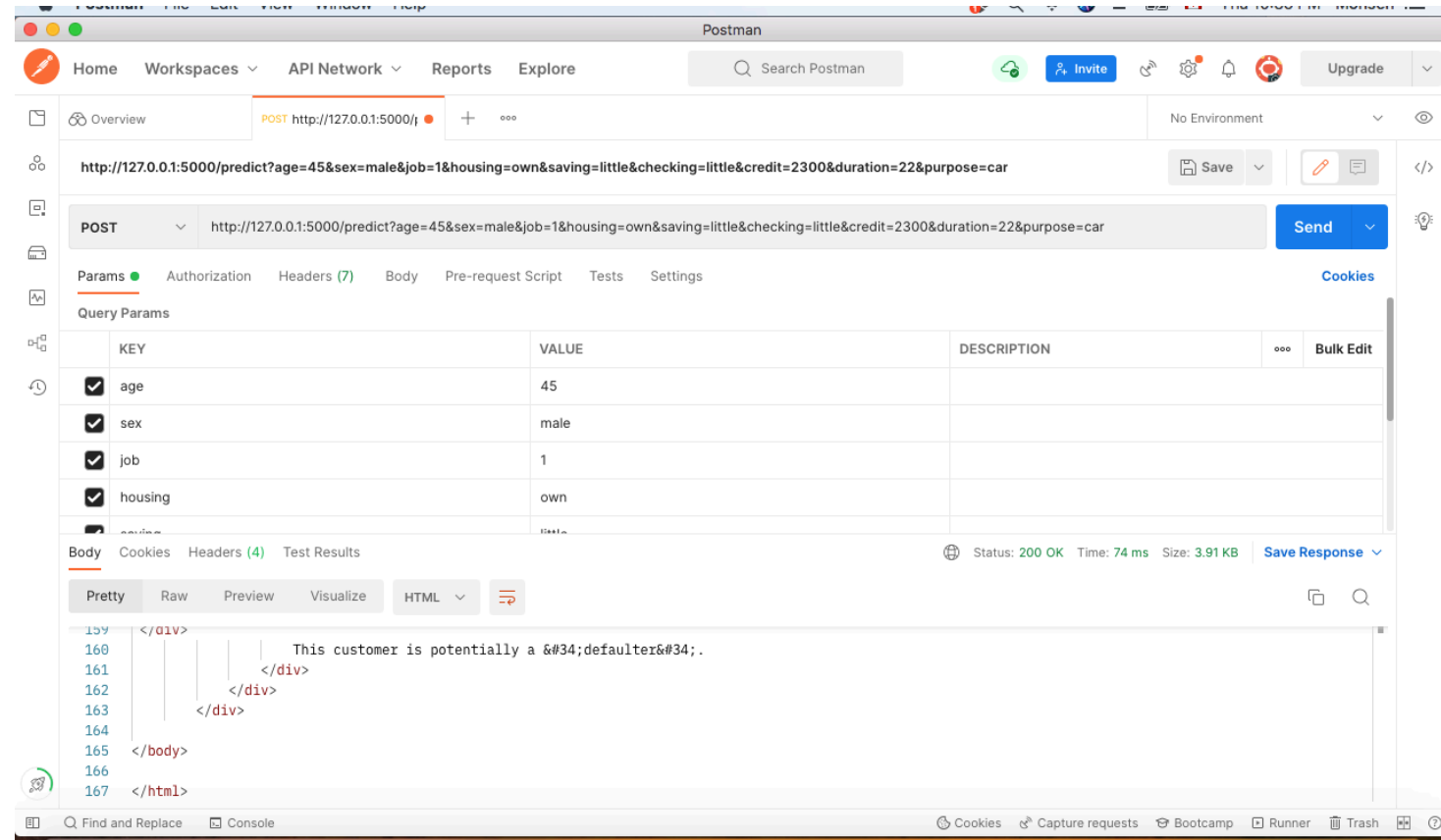
@app.route('/predict', methods=['POST'])
def predict():
    features = [x for x in request.form.values()]
    a = np.array(features)
    b = pd.DataFrame(a, columns=['Age', 'Sex', 'Job', 'Housing', 'Saving accounts',
                                'Checking account', 'Credit amount', 'Duration',
                                'Purpose'])
    c = pd.get_dummies(b, columns=[
        'Sex', 'Housing', 'Saving accounts', 'Checking account',
        'Purpose'
    ])
    new_x = pd.DataFrame(np.zeros((1, 19)), columns=['Age', 'Job', 'Credit amount', 'Duration', 'Sex_male', 'Housing_own',
                                                    'Housing_rent', 'Saving accounts_moderate',
                                                    'Saving accounts_quite rich', 'Saving accounts_rich',
                                                    'Checking account_moderate', 'Checking account_rich', 'Purpose_car',
                                                    'Purpose_domestic appliances', 'Purpose_education',
                                                    'Purpose_furniture/equipment', 'Purpose_radio/TV', 'Purpose_repairs',
                                                    'Purpose_vacation/others'])
    for i in c.columns:
        if i in new_x.columns:
            new_x[i] = 1
    pred = clf.predict(new_x)[0]
    if pred == 1:
        output = "non defaulter"
    else:
        output = "defaulter"

    return render_template('index.html', prediction_text='This customer is potentially a {}'.format(output))

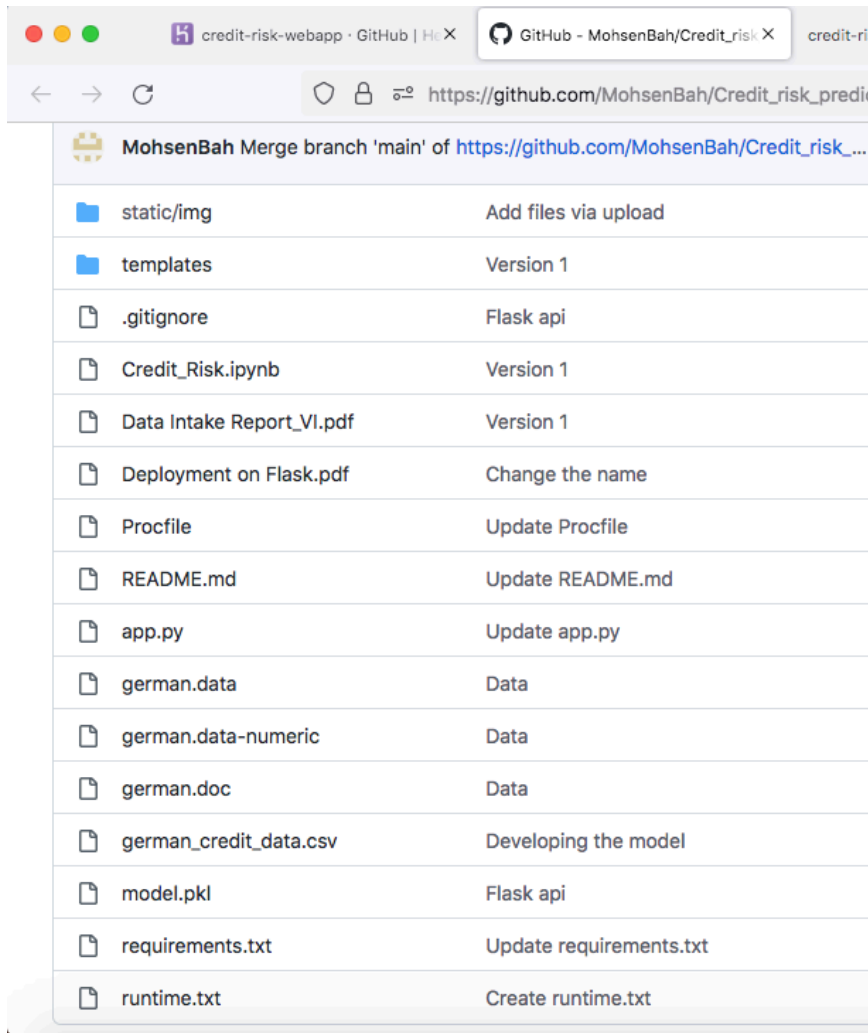
if __name__ == "__main__":
    app.run(debug=True)
```

# Testing the model on Postman API

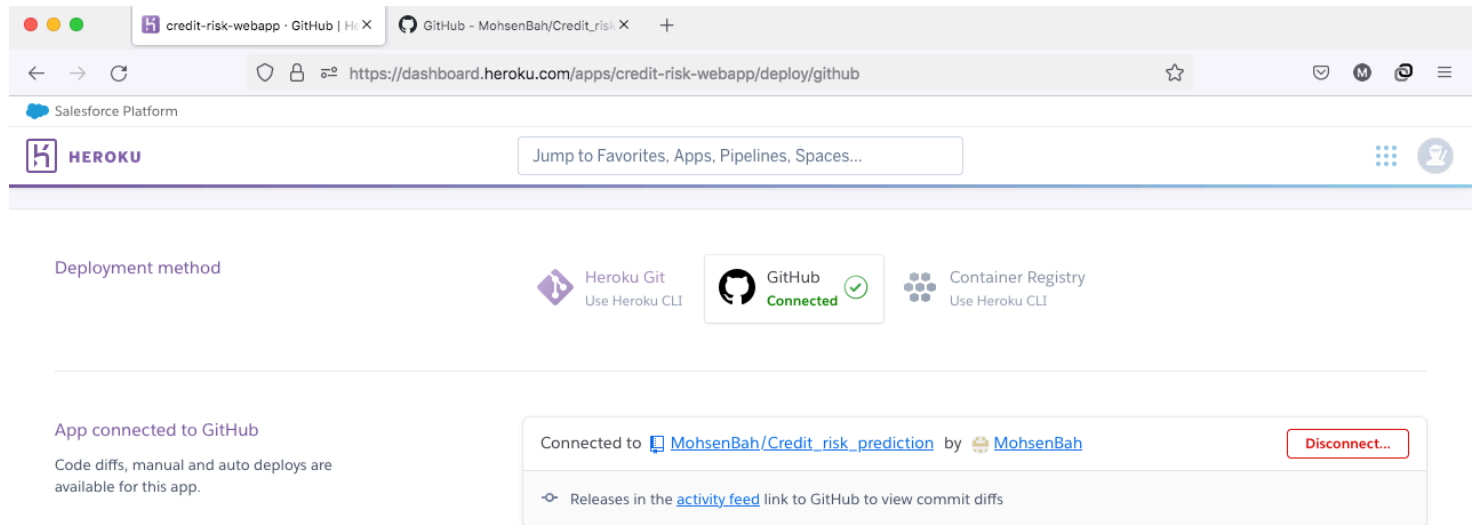
❖ The model was tested successfully on Postman and it is ready to deploy on Heroku cloud.



# Deploying on Heroku



MohsenBah Merge branch 'main' of https://github.com/MohsenBah/Credit_risk_prediction	
static/img	Add files via upload
templates	Version 1
.gitignore	Flask api
Credit_Risk.ipynb	Version 1
Data Intake Report_VI.pdf	Version 1
Deployment on Flask.pdf	Change the name
Procfile	Update Procfile
README.md	Update README.md
app.py	Update app.py
german.data	Data
german.data-numeric	Data
german.doc	Data
german_credit_data.csv	Developing the model
model.pkl	Flask api
requirements.txt	Update requirements.txt
runtime.txt	Create runtime.txt



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 [MohsenBah/Credit\\_risk\\_prediction](#) by [MohsenBah](#) [Disconnect...](#)

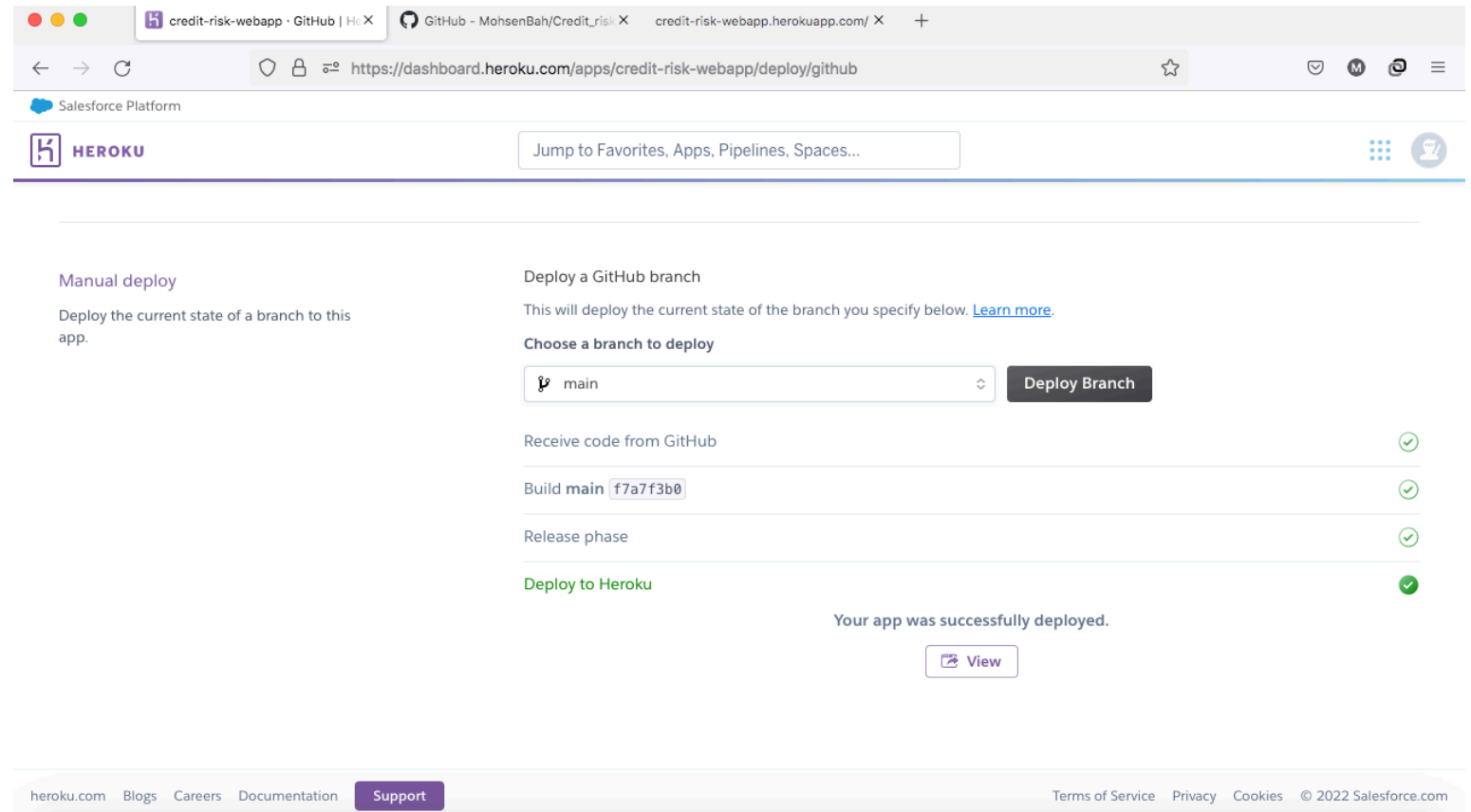
Releases in the [activity feed](#) link to GitHub to view commit diffs



1. All files (requirements.txt and Procfile) are ready to be deployed on the repo.
2. The repo connected with Heroku.

# Deploying on Heroku – cont'd

❖ App successfully deployed on Heroku cloud.



The screenshot shows the Heroku dashboard for the application 'credit-risk-webapp'. The browser tabs include 'credit-risk-webapp · GitHub', 'GitHub - MohsenBah/Credit\_risk', and 'credit-risk-webapp.herokuapp.com/'. The address bar shows 'https://dashboard.heroku.com/apps/credit-risk-webapp/deploy/github'. The Heroku logo and a search bar are at the top. The main content area is titled 'Manual deploy' and explains that it will deploy the current state of a branch to the app. A section titled 'Deploy a GitHub branch' shows a dropdown menu with 'main' selected and a 'Deploy Branch' button. Below this, a progress bar shows four steps: 'Receive code from GitHub', 'Build main f7a7f3b0', 'Release phase', and 'Deploy to Heroku', all of which are marked with green checkmarks. A message at the bottom states 'Your app was successfully deployed.' with a 'View' button. The footer contains links for 'heroku.com', 'Blogs', 'Careers', 'Documentation', 'Support', 'Terms of Service', 'Privacy', 'Cookies', and '© 2022 Salesforce.com'.

credit-risk-webapp · GitHub | Heroku | GitHub - MohsenBah/Credit\_risk | credit-risk-webapp.herokuapp.com/

https://dashboard.heroku.com/apps/credit-risk-webapp/deploy/github

Salesforce Platform

HEROKU

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

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 f7a7f3b0

Release phase

Deploy to Heroku

Your app was successfully deployed.

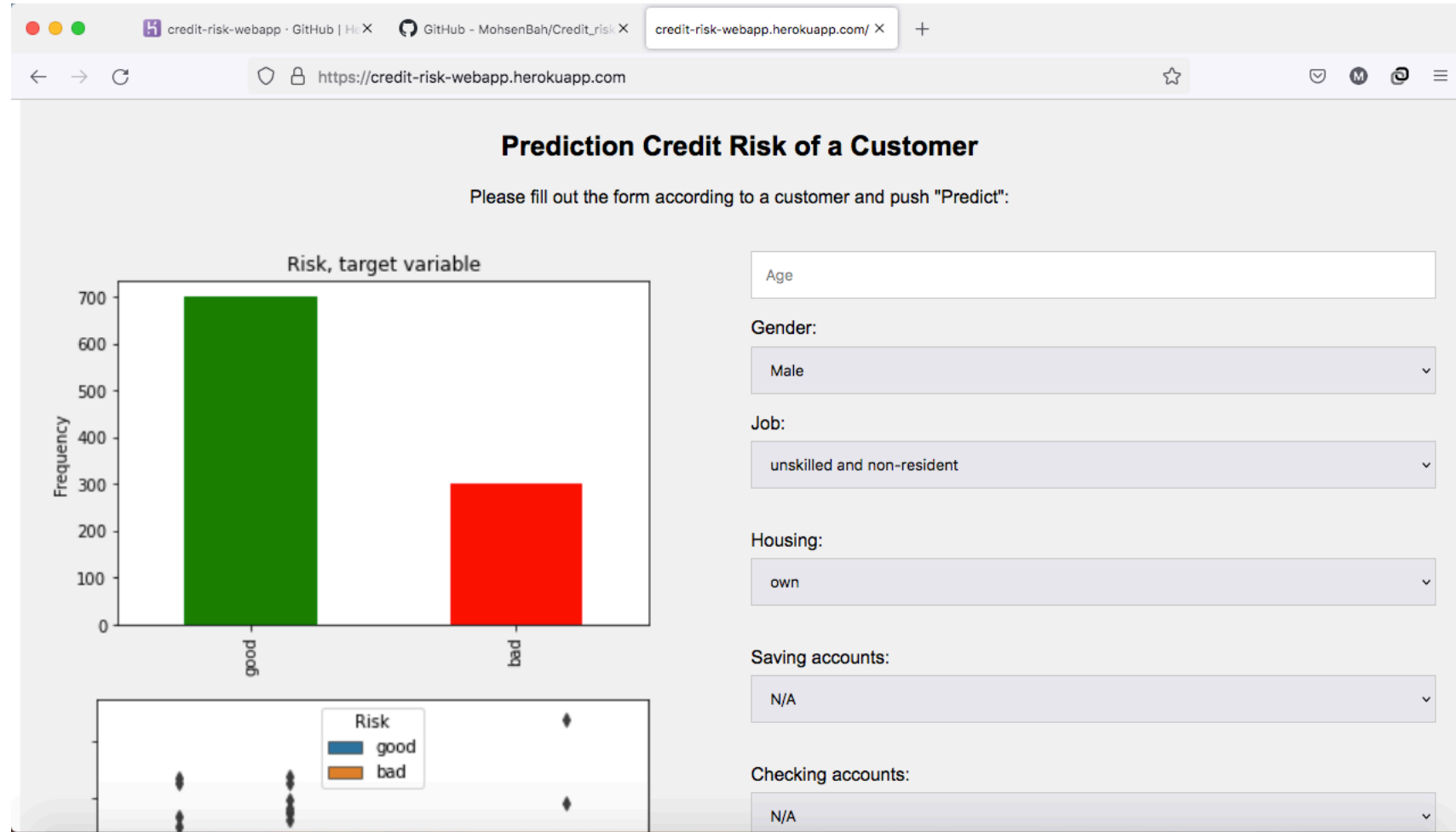
View

heroku.com Blogs Careers Documentation Support Terms of Service Privacy Cookies © 2022 Salesforce.com



# Deploying on Heroku – cont'd

- ❖ <https://credit-risk-webapp.herokuapp.com>
- ❖ It is the successfully deployed app.



# Thank You