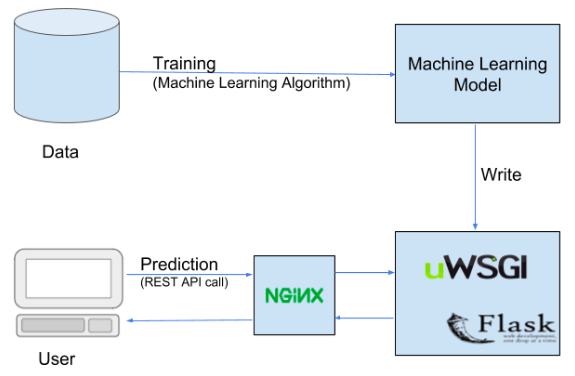
TUTORIAL FLASK



1. Instalar o Python <https://www.python.org/downloads/>

Verifique se instalou corretamente, no terminal: py -3 --version

1. Instalar a extensão no VSCode <https://marketplace.visualstudio.com/items?itemName=ms-python.python>
2. Selecione um interpretador Python: No VS Code, selecione um interpretador Python 3 abrindo a Paleta de Comandos ( Ctrl + Shift + P ), comece a digitar o Python: Selecione o comando Interpreter
3. Crie um arquivo de código-fonte do Python Hello World
   1. Criar uma pasta e dentro da mesma um arquivo hello.py
   2. É simples de executar hello.py com Python. Basta clicar no botão Executar arquivo Python no Terminal no lado superior direito do editor.

msg = "Hello World"

print(msg)

VAMOS PARA O FLASK AGORA

1. Instalando o Flask

py -3 -m pip install --upgrade pip

py -3 -m pip install flask

1. Crie e execute um aplicativo Flask mínimo
   1. No VS Code, crie um novo arquivo na pasta do seu projeto nomeado app.py

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route("/")

def home():

return "Hello, Flask!"

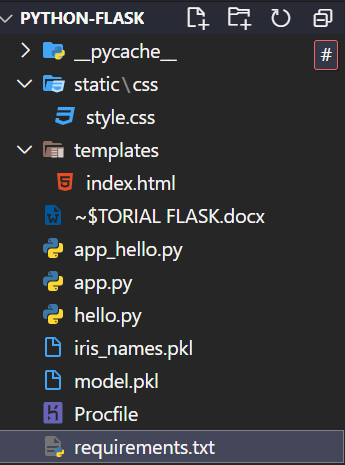
1. Salve o app.py ( Ctrl + S ).
2. No Terminal (verique se está na pasta do projeto), execute o aplicativo digitando py -3 -m flask run.

**MACHINE LERNING DEPLOY COM FLASK**

1. Instalar numpy e sklearn

py -3 -m pip install numpy

**py -3 -m pip install sklearn**



**Index.html**

<!DOCTYPE *html*>

<html >

<head>

  <meta *charset*="UTF-8">

  <title>Deployment Tutorial 1</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" *type*= "text/css" *href*= "{{ url\_for('static',filename='css/style.css') }}">

</head>

<body *style*="background: #000;">

 <div *class*="login">

  <h1>Classificador de espécies IRIS</h1>

     <!-- Main Input For Receiving Query to our ML -->

    <form *action*="{{ url\_for('predict')}}" *method*="post">

    <input *type*="text" *name*="sl" *placeholder*="Sepal length (cm)" *required*="required" />

    <input *type*="text" *name*="sw" *placeholder*="Sepal width (cm)" *required*="required" />

    <input *type*="text" *name*="pl" *placeholder*="Petal length (cm)" *required*="required" />

    <input *type*="text" *name*="pw" *placeholder*="Petal length (cm)" *required*="required" />

        <button *type*="submit" *class*="btn btn-primary btn-block btn-large">Classificar IRIS</button>

    </form>

   <br>

   <br>

   {{ prediction\_text }}

 </div>

</body>

</html>

**Style.css**

@import url(*https://fonts.googleapis.com/css?family=Open+Sans*);

html {

  width: 100%;

  height: 100%;

  overflow: hidden;

}

h1 {

  color: #fff;

  text-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

  letter-spacing: 1px;

  text-align: center;

}

body {

  width: 100%;

  height: 100%;

  font-family: "Helvetica";

  background: blue;

  color: #fff;

  font-size: 24px;

  text-align: center;

  letter-spacing: 1.4px;

}

*.login* {

  position: absolute;

  top: 40%;

  left: 50%;

  margin: -150px 0 0 -150px;

  width: 400px;

  height: 400px;

}

input {

  width: 100%;

  margin-bottom: 10px;

  background: rgba(0, 0, 0, 0.3);

  border: none;

  outline: none;

  padding: 10px;

  font-size: 13px;

  color: #fff;

  text-shadow: 1px 1px 1px rgba(0, 0, 0, 0.3);

  border: 1px solid rgba(0, 0, 0, 0.3);

  border-radius: 4px;

  box-shadow: inset 0 -5px 45px rgba(100, 100, 100, 0.2),

    0 1px 1px rgba(255, 255, 255, 0.2);

*-webkit-transition*: box-shadow 0.5s ease;

*-moz-transition*: box-shadow 0.5s ease;

*-o-transition*: box-shadow 0.5s ease;

*-ms-transition*: box-shadow 0.5s ease;

  transition: box-shadow 0.5s ease;

}

**App.py**

import numpy as np

from flask import Flask, request, render\_template

import pickle

app = Flask(\_\_name\_\_)

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

names = pickle.load(open("iris\_names.pkl", "rb"))

@app.route("/")

def home():

    return render\_template("index.html")

@app.route("/predict", *methods*=["POST"])

def predict():

    features = [float(x) for x in request.form.values()]

    final\_features = [np.array(features)]

    pred = model.predict(final\_features)

    output = names[pred[0]]

    return render\_template("index.html", *prediction\_text*=output)

Para executar

py -3 -m flask run

**DEPLOY ON HEROKU**

1. Subir o projeto no GIT, se cadastrar no Heroku

Heroku é uma das primeiras plataformas de nuvem como serviço (PaaS) e oferece suporte a várias linguagens - Ruby, Java, Node.js, Scala, Clojure, Python, PHP e Go.

A primeira coisa que precisamos fazer é definir quais bibliotecas nosso aplicativo usa. Dessa forma, o Heroku sabe quais fornecer para nós, semelhante à forma como os instalamos localmente ao desenvolver o aplicativo.

Para isso, precisamos criar um requirements.txt arquivo com todos os módulos:

Flask==2.0.0

numpy==1.20.3

scikit-learn==0.24.2

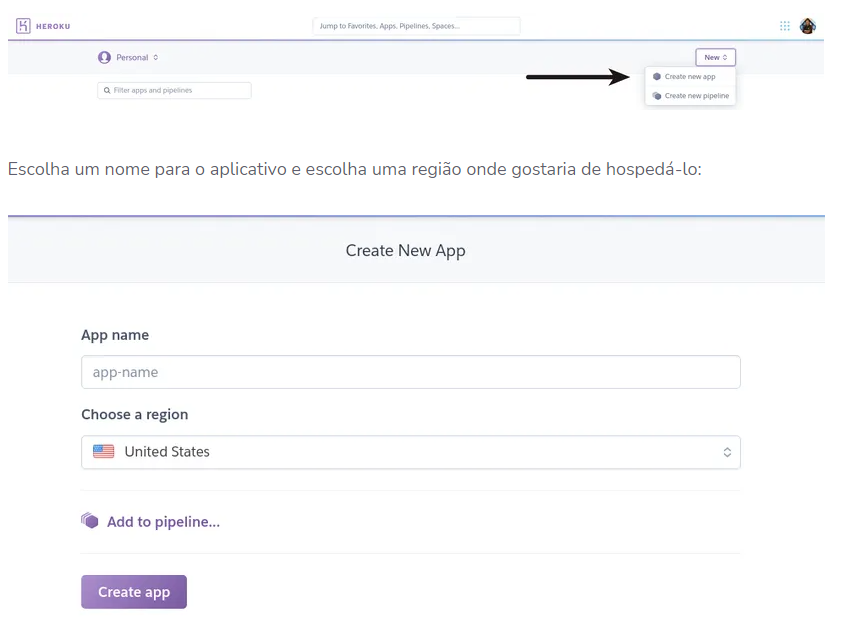
gunicorn==19.0.0

Para que o Heroku possa rodar nosso aplicativo como deveria, precisamos definir um conjunto de processos / comandos que ele deve rodar de antemão. Esses comandos estão localizados em **Procfile**:

web: gunicorn app:app

O comando **web** diz ao Heroku para iniciar um servidor web para o aplicativo, usando gunicorn. Como nosso aplicativo é chamado app.py, definimos o app nome para ser app também.

**Com a conta já criada devemos criar um novo app**



Após criar o app, escolha a opção de fazer o deploy usando um repositório git, selecione-o e deixe habilitado a opção de deploy automático.

Se tudo der certo o app estará online em segundos.

**Referências**

<https://code.visualstudio.com/docs/python/python-tutorial#_prerequisites>

<https://code.visualstudio.com/docs/python/tutorial-flask>

<https://towardsdatascience.com/how-to-easily-deploy-machine-learning-models-using-flask-b95af8fe34d4>

<https://stackabuse.com/deploying-a-flask-application-to-heroku/>