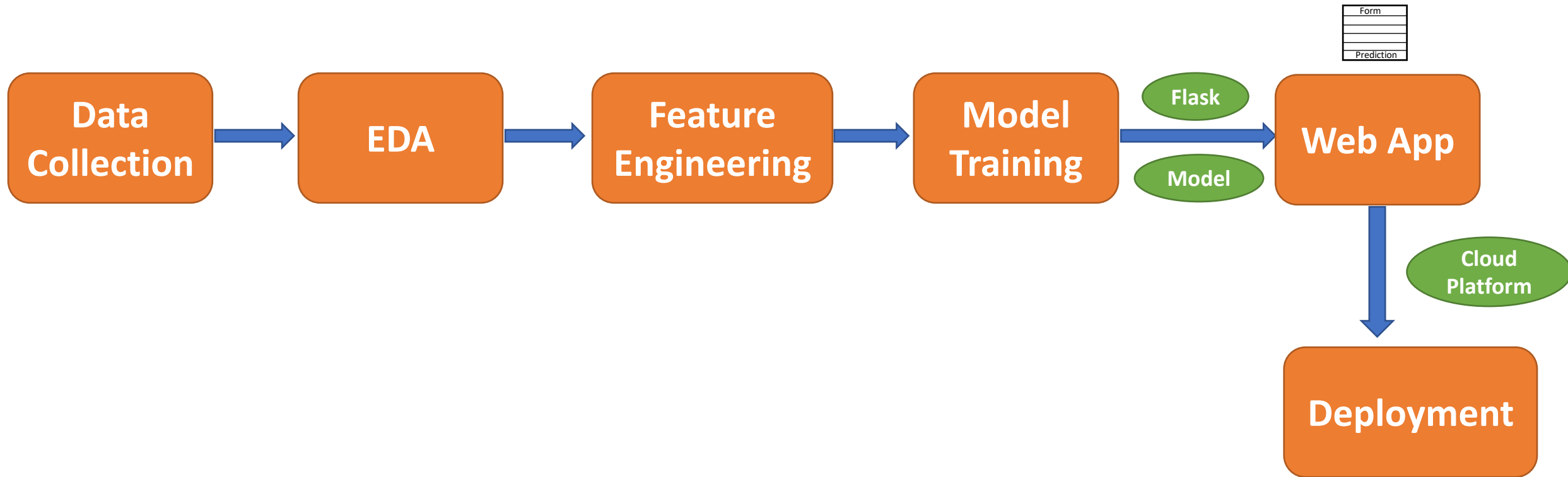


Data Science Masters

Machine Learning End-to-End Project

Linear Regression Project

Project Life Cycle



Folder and File

- Step 1.
 - * Create **dataset** folder for *data*
 - * Create **models** folder for *.pkl models*
 - * Create **notebooks** folder for *.ipynb files*
- Step 2.
 - * Create **application.py** file for *flask app*
 - * Create **requirements.txt** file for require libraries
 - * Create **templates** folder for *index.html, home.html*

Coding

- index.html

```
<h1> MOHAMMAD WASIQ Welcome to the Home Page </h1>
```

- requirements.txt

```
Flask  
numpy  
pandas  
scikit-learn
```

```
pip install -r requirements.txt
```

- home.html

```
<html>  
<body>  
  <div class="login">  
    <h1>FWI Prediction by MOHAMMAD WASIQ</h1>  
  
    <!-- Main Input For Receiving Query to our ML -->  
    <form action="{{ url_for('predict_datapoint')}}" method="post">  
      <input type="text" name="Temperature" placeholder="Temperature" required="required"  
/><br>  
      <input type="text" name="RH" placeholder="RH" required="required" /><br>  
      <input type="text" name="Ws" placeholder="Ws" required="required" /><br>  
      <input type="text" name="Rain" placeholder="Rain" required="required" /><br>  
      <input type="text" name="FFMC" placeholder="FFMC" required="required" /><br>  
      <input type="text" name="DMC" placeholder="DMC" required="required" /><br>  
      <input type="text" name="ISI" placeholder="ISI" required="required" /><br>  
      <input type="text" name="Classes" placeholder="Classes" required="required" /><br>  
      <input type="text" name="Region" placeholder="Region" required="required" /><br>  
  
      <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>  
    </form>  
    <h2>  
      THE FWI prediction is {{result}}  
    </h2>  
  </body>  
</html>
```

Coding

- application.py

```
import pickle
from flask import Flask, request, jsonify, render_template
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler

application = Flask(__name__)
app=application

# import ridge regresor model and standard scaler pickle
ridge_model=pickle.load(open('models/ridge.pkl','rb'))
standard_scaler=pickle.load(open('models/scaler.pkl','rb'))

# Route for home page
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predictdata',methods=['GET','POST'])
def predict_datapoint():
    if request.method=='POST':
        Temperature=float(request.form.get('Temperature'))
        RH = float(request.form.get('RH'))
        Ws = float(request.form.get('Ws'))
        Rain = float(request.form.get('Rain'))
        FFMC = float(request.form.get('FFMC'))
        DMC = float(request.form.get('DMC'))
        ISI = float(request.form.get('ISI'))
        Classes = float(request.form.get('Classes'))
        Region = float(request.form.get('Region'))

        new_data_scaled=standard_scaler.transform([[Temperature,RH,Ws,Rain,FFMC,DMC,ISI,Classes,Region]])
        result=ridge_model.predict(new_data_scaled)

        return render_template('home.html',result=result[0])

    else:
        return render_template('home.html')

if __name__=="__main__":
    app.run(host="0.0.0.0")
```

Git Commands

- `ls -a`
- `git remote -v`
- `git remote rm origin`
- `git init`
- `git add README.md`
- `git add .`
- `git status`
- `git commit -m "first commit"`
- `git config --global user.email "mohammadwasiq0786@gmail.com"`
- `git config --global user.name "mohamamdwasiq0"`
- `git commit -m "first commit"`
- `git branch -M main`
- `git branch`
- `git remote add origin < GitHub Repo Link >`
- `git push -u origin main`

- Now it requires to permission to sign in the GitHub.

AWS Deployment

- Create **.ebextensions** folder
- In this folder create **python.config** file and write the following code

```
option_settings:  
  "aws:elasticbeanstalk:container:python":  
    WSGIPath: application:application
```

Go to Console Home

- Click on Elastic Beanstalk
- On Left Top Click on **Applications**
- After Clicking On Right Side Click on **Create Application**
- In **Application name** give any *name* < algeriaforestfire >
- On **Platform** choose *Platform* as **Python**
- In **Sample code** select *Sample application*
- Click on **Create Application** . It takes some times to get ready.

Code Pipeline

Search CodePipeline

- Click on **CodePipeline**
- On Left Top Click on **CodePipeline** **Create Pipeline**
- In **Pipeline name** give any *name* < algeriaforestfire >
- Set everything default
- On Right Bottom Click on **Next**
- After that in **Source Provider** choose **GitHub** (Version 1)
- Click on **Connect to GitHub**
- **Confirm** the **Processing OAuth request**
- In **Repository** choose < Repository Name >
- In **Branch** choose **main** and Click on **Next**

Code Pipeline

- After clicking **Next**
- On Add build stage Click on **Skip**
- After Skipping there is an another window names **Add deploy stage** select **deploy provider** as **AWS Elastic Beanstalk**
- In **Region** select **Asia Pacific (Mumbai)**
- In **Application** name select **< algeriaforestfire >**
- In **Environment** name select **< algeriaforestfire-env >**
- Click on **Next**
- After that there is an another window named **Review**
- Go to **Elastic Beanstalk** and click on **algeriaforestfire-env** and check **health** when it is **Ok** then.
- Go to again **Review** and on Right bottom Click on **Create Pipeline**