

DEPLOYMENT ON FLASK

NAME: CATHERINE SANDA

BATCH CODE:LISUM01

SUBMISSION DATE:7/2/21

SUBMISSION TO:

This document contains the report of the steps carried out in order to deploy a model.

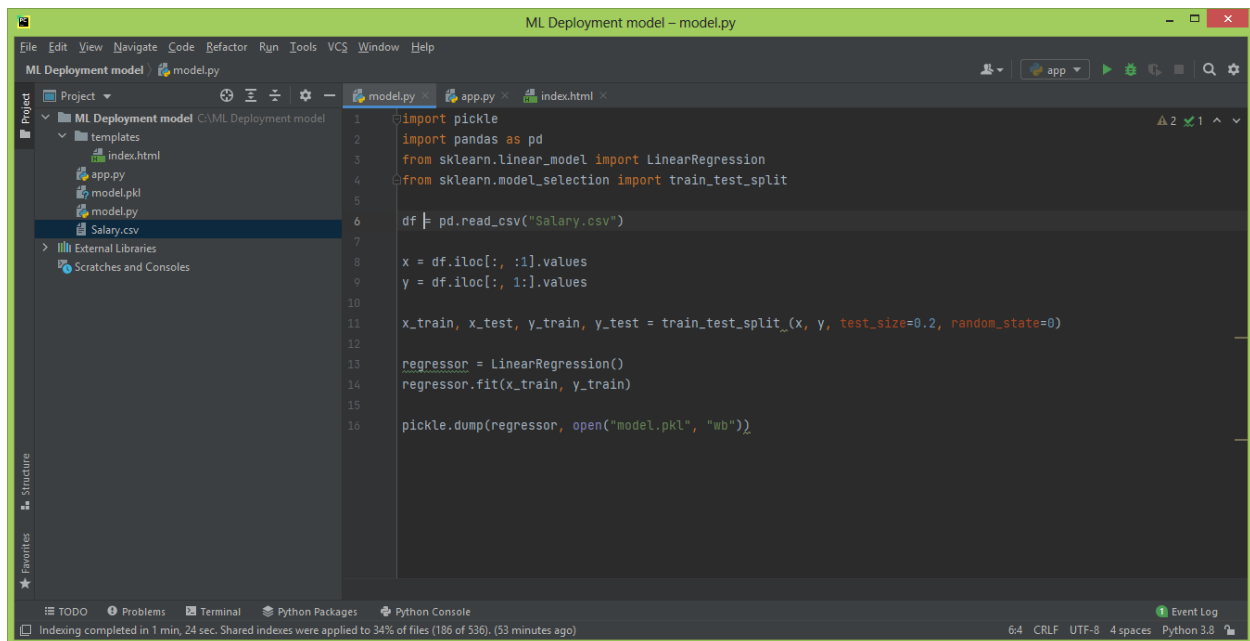
I used PyCharm, Mozilla Firefox and python programming language as well as different packages and libraries.

I created five files:

- Model.py (ML model)
- Model.pkl (Pickle file of the ML model)
- app.py (Flask application)
- index.html
- Salary.csv (data to build the ML model)

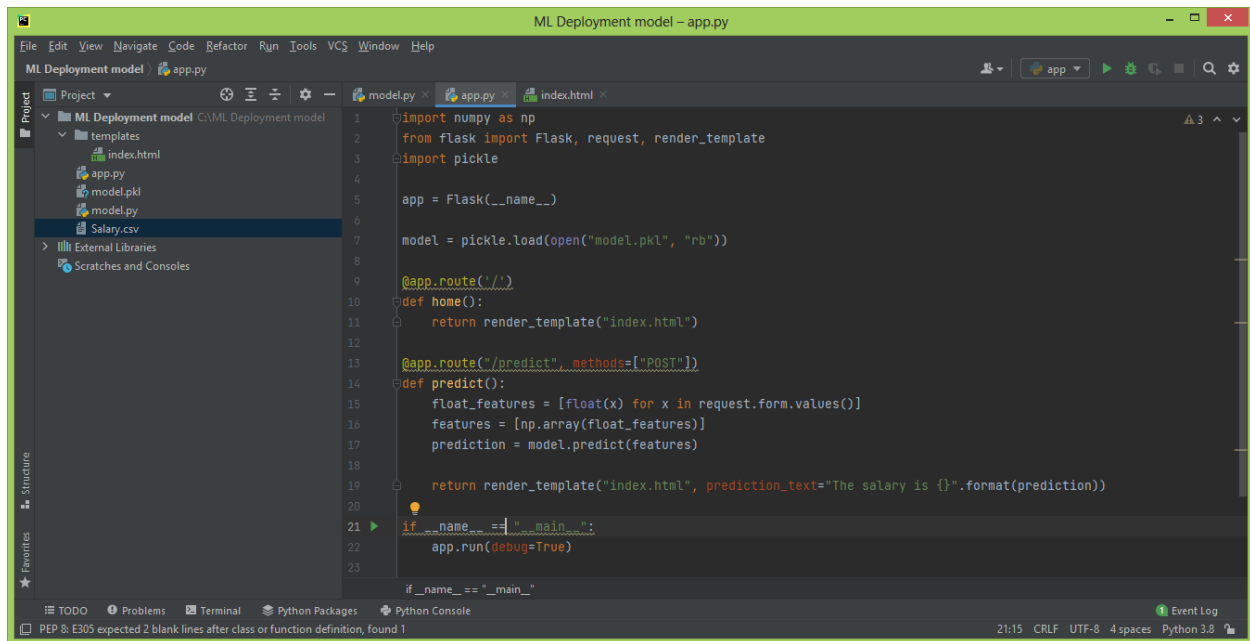
These files are on <https://github.com/cate6495/week4-redone>

First, I created the model:

A screenshot of the PyCharm IDE interface. The title bar reads "ML Deployment model - model.py". The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The left sidebar shows a project tree with folders "ML Deployment model" and "templates", and files "index.html", "app.py", "model.pkl", "model.py", and "Salary.csv". The main editor window displays the code for "model.py". The code imports pickle, pandas as pd, LinearRegression from sklearn.linear_model, and train_test_split from sklearn.model_selection. It reads "Salary.csv" into a DataFrame, splits it into x and y, then into training and testing sets. A LinearRegression model is trained on the training data and saved as "model.pkl". The status bar at the bottom shows "Indexing completed in 1 min, 24 sec. Shared indexes were applied to 34% of files (186 of 536). (53 minutes ago)", "64 CRLF UTF-8 4 spaces Python 3.8", and an "Event Log" icon.

```
1 import pickle
2 import pandas as pd
3 from sklearn.linear_model import LinearRegression
4 from sklearn.model_selection import train_test_split
5
6 df = pd.read_csv("Salary.csv")
7
8 x = df.iloc[:, :1].values
9 y = df.iloc[:, 1:].values
10
11 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
12
13 regressor = LinearRegression()
14 regressor.fit(x_train, y_train)
15
16 pickle.dump(regressor, open("model.pkl", "wb"))
```

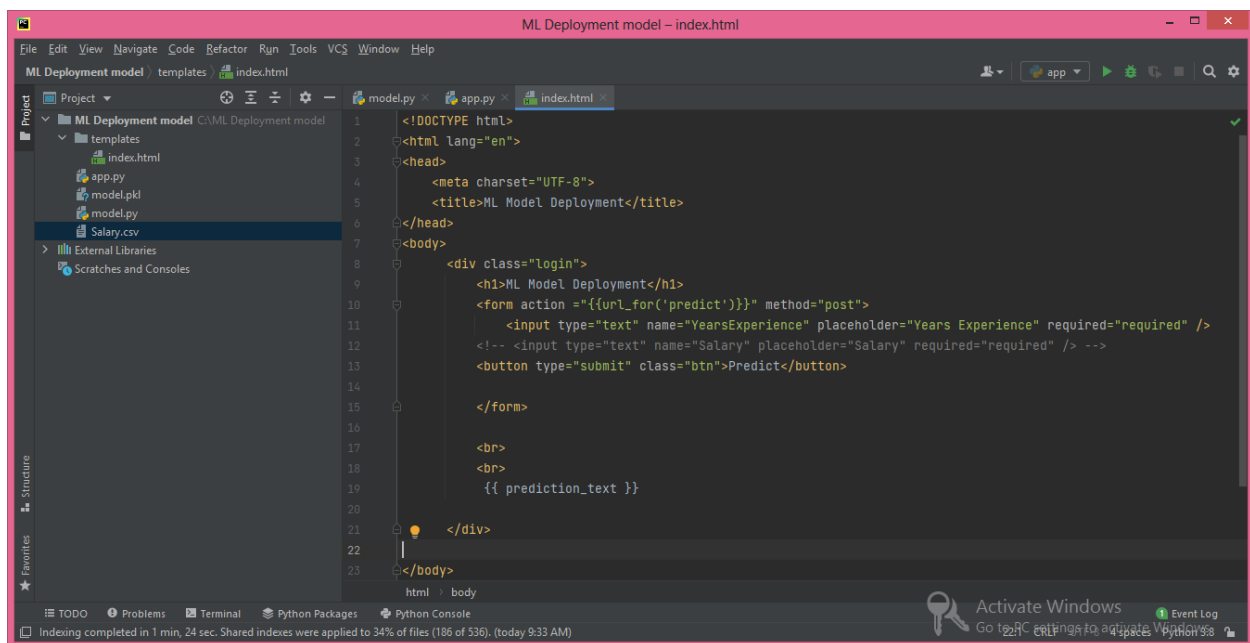
Then the flask app:



The screenshot shows the VS Code editor with the file 'app.py' open. The code is a Flask application that loads a pre-trained model and provides a web interface for predictions. The code is as follows:

```
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pickle
4
5 app = Flask(__name__)
6
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
19     return render_template("index.html", prediction_text="The salary is {}".format(prediction))
20
21 if __name__ == "__main__":
22     app.run(debug=True)
23
24 if __name__ == "__main__":
```

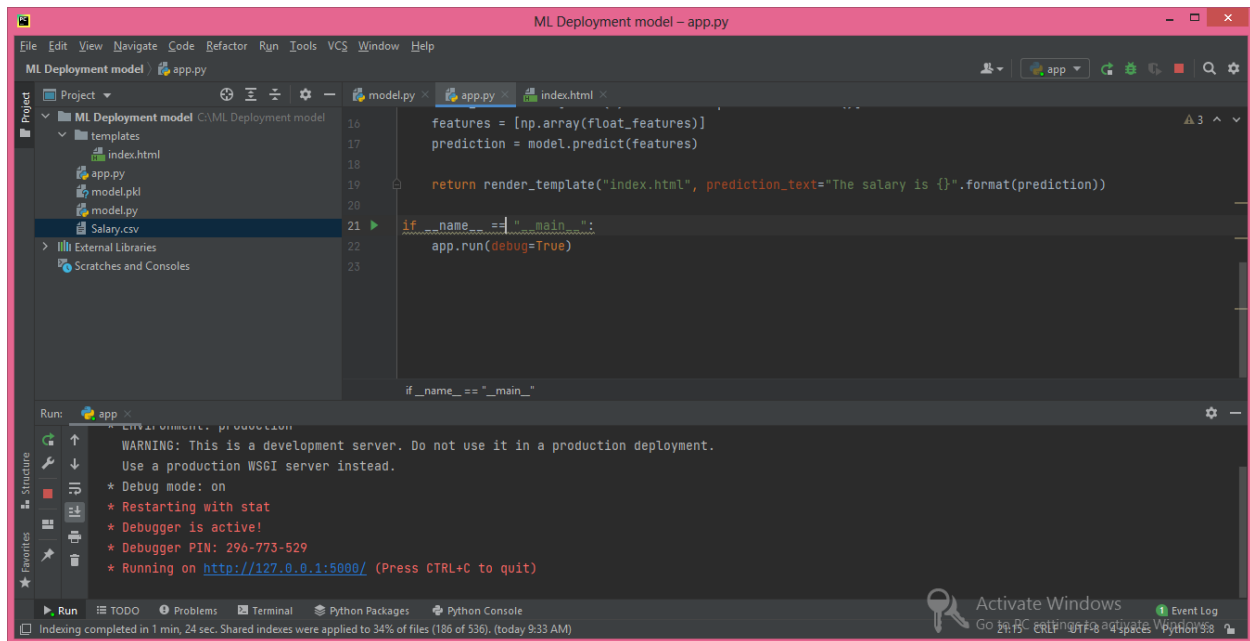
The index.html:



The screenshot shows the VS Code editor with the file 'index.html' open. The code is an HTML template for the web application. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>ML Model Deployment</title>
6 </head>
7 <body>
8     <div class="login">
9         <h1>ML Model Deployment</h1>
10        <form action="{{url_for('predict')}}" method="post">
11            <input type="text" name="YearsExperience" placeholder="Years Experience" required="required" />
12            <!-- <input type="text" name="Salary" placeholder="Salary" required="required" /> -->
13            <button type="submit" class="btn">Predict</button>
14        </form>
15
16        <br>
17        <br>
18        {{ prediction_text }}
19
20    </div>
21
22
23 </body>
```

On the running the app:



Checking how well our model functions:



ML Model Deployment

Years Experience



ML Model Deployment

Years Experience

The salary is [[115742.98052455]]