Week 4: Deployment on Flask

Name: Madhvik Bhalani

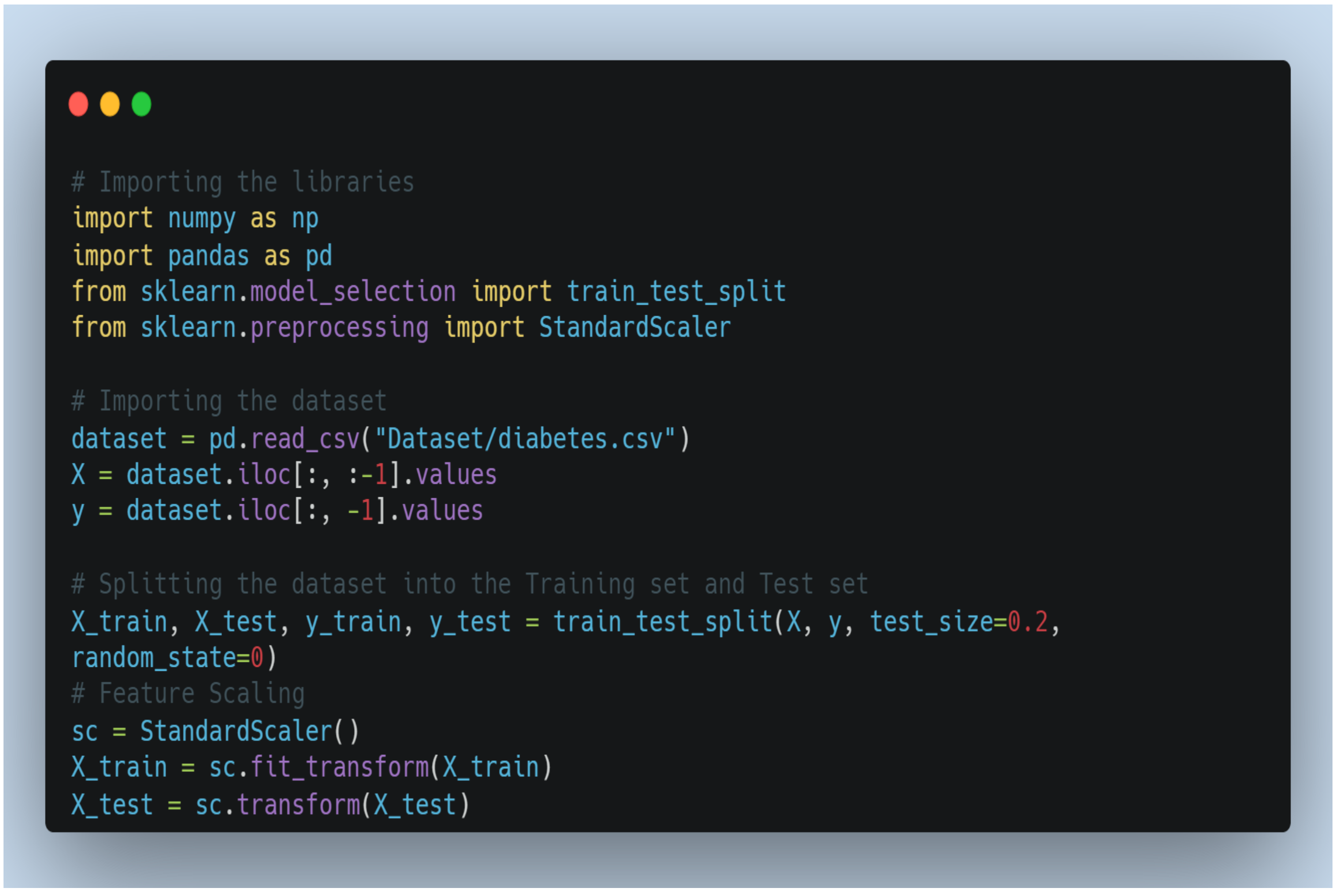
Batch code: LISUM32

Submission date: 09 May 2024

Submitted to: Data Glacier

1. Implementing Classification Model

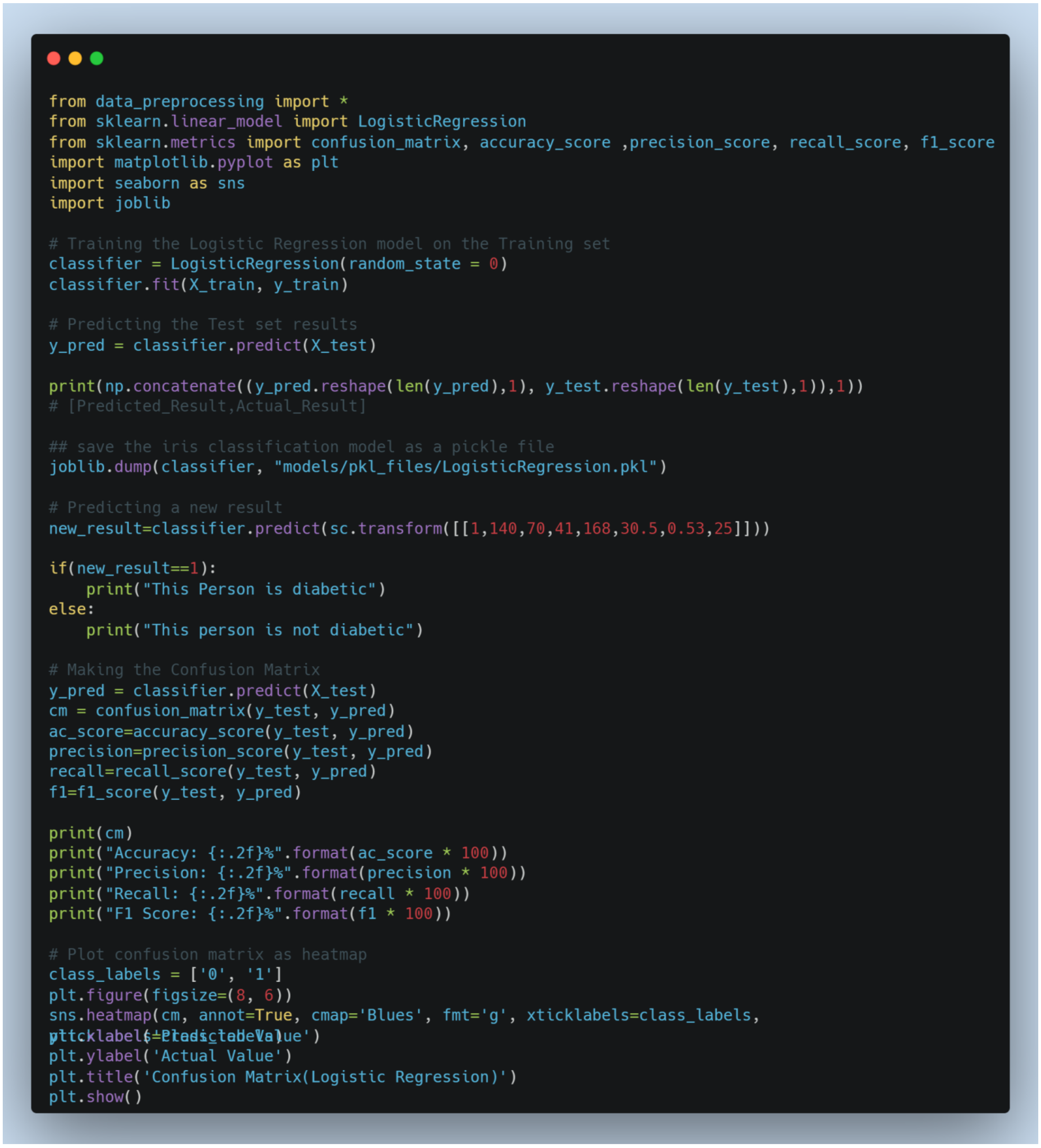
Step 1: Data Preprocessing



```python
# Importing the libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Importing the dataset
dataset = pd.read_csv("Dataset/diabetes.csv")
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

# Splitting the dataset into the Training set and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
# Feature Scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Step 2: Implementing Logistic Regression

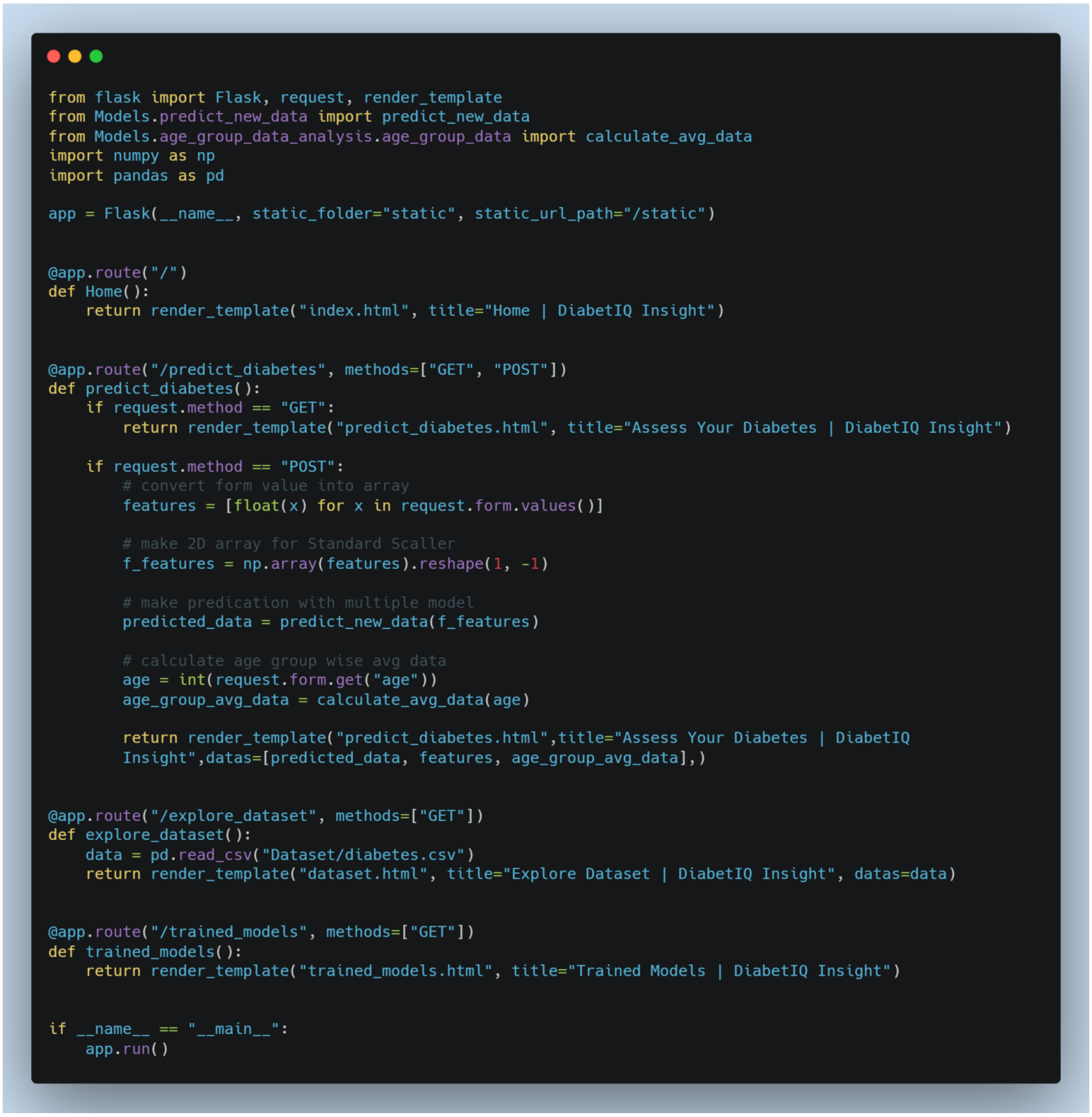
After thorough experimentation with various models, I've chosen to present the Logistic Regression Classifier here because of its highest accuracy.



```python
from data_preprocessing import *
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score ,precision_score, recall_score, f1_score
import matplotlib.pyplot as plt
import seaborn as sns
import joblib

# Training the Logistic Regression model on the Training set
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
# [Predicted_Result,Actual_Result]

## save the iris classification model as a pickle file
joblib.dump(classifier, "models/pkl_files/LogisticRegression.pkl")

# Predicting a new result
new_result=classifier.predict(sc.transform([[1,140,70,41,168,30.5,0.53,25]]))

if(new_result==1):
    print("This Person is diabetic")
else:
    print("This person is not diabetic")

# Making the Confusion Matrix
y_pred = classifier.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
ac_score=accuracy_score(y_test, y_pred)
precision=precision_score(y_test, y_pred)
recall=recall_score(y_test, y_pred)
f1=f1_score(y_test, y_pred)

print(cm)
print("Accuracy: {:.2f}%".format(ac_score * 100))
print("Precision: {:.2f}%".format(precision * 100))
print("Recall: {:.2f}%".format(recall * 100))
print("F1 Score: {:.2f}%".format(f1 * 100))

# Plot confusion matrix as heatmap
class_labels = ['0', '1']
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, cmap='Blues', fmt='g', xticklabels=class_labels,
yticklabels=class_labels)
plt.xlabel('Predicted Value')
plt.ylabel('Actual Value')
plt.title('Confusion Matrix(Logistic Regression)')
plt.show()
```

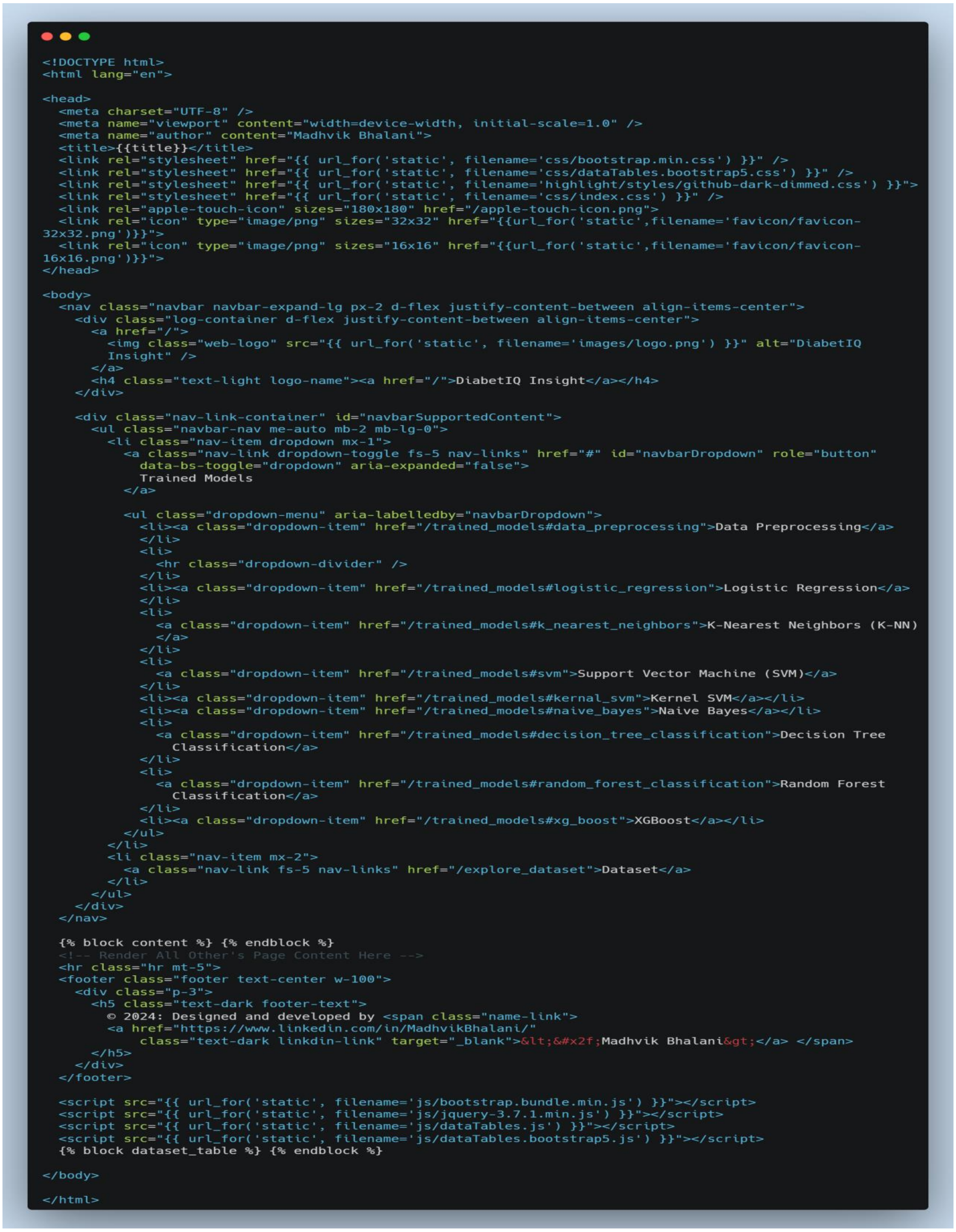
2. Flask App Development

Step 1: Create and Setup Flask App



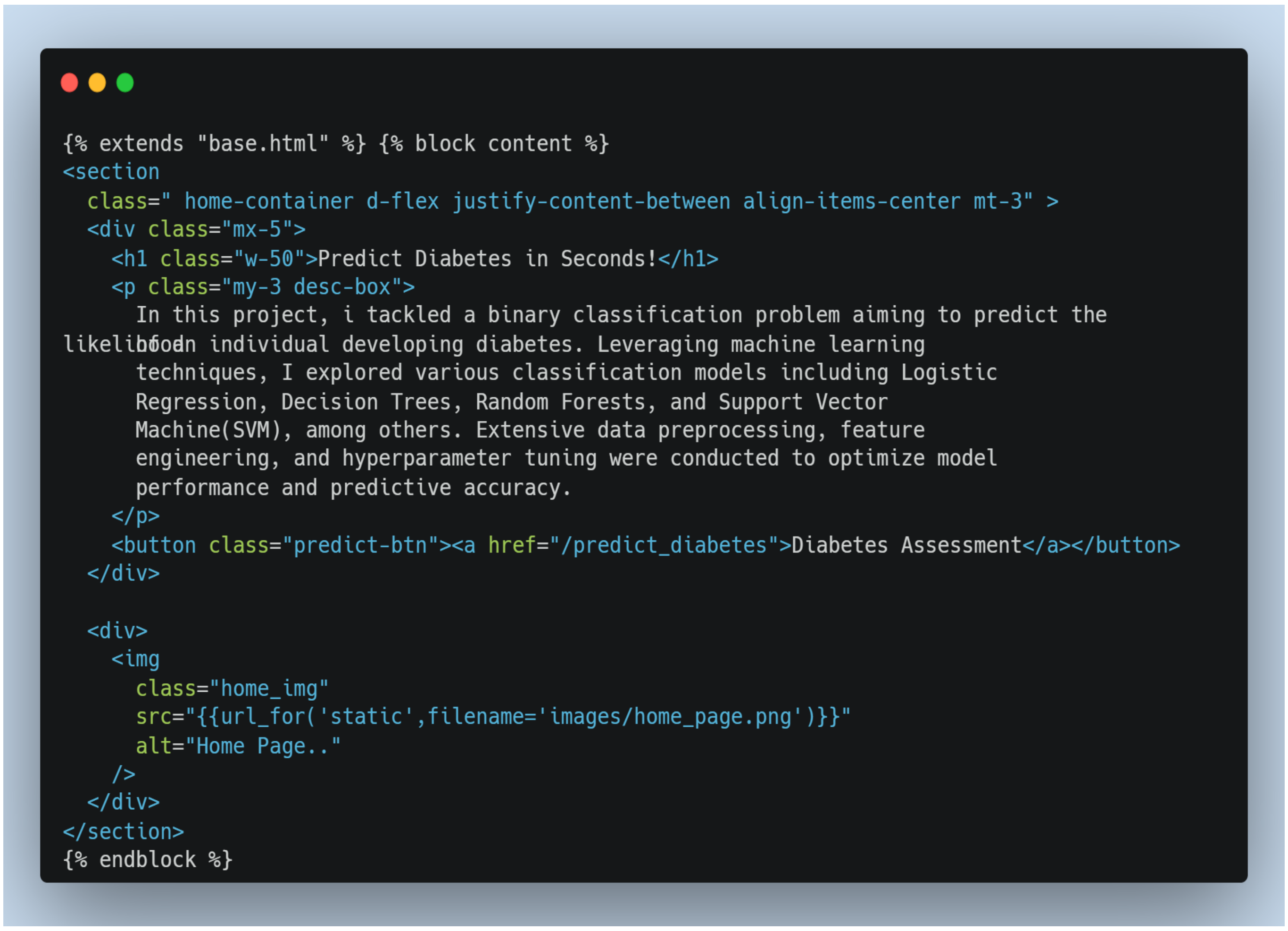
```python
from flask import Flask, request, render_template
from Models.predict_new_data import predict_new_data
from Models.age_group_data_analysis.age_group_data import calculate_avg_data
import numpy as np
import pandas as pd

app = Flask(__name__, static_folder="static", static_url_path="/static")


@app.route("/")
def Home():
    return render_template("index.html", title="Home | DiabetIQ Insight")


@app.route("/predict_diabetes", methods=["GET", "POST"])
def predict_diabetes():
    if request.method == "GET":
        return render_template("predict_diabetes.html", title="Assess Your Diabetes | DiabetIQ Insight")

    if request.method == "POST":
        # convert form value into array
        features = [float(x) for x in request.form.values()]

        # make 2D array for Standard Scaller
        f_features = np.array(features).reshape(1, -1)

        # make predication with multiple model
        predicted_data = predict_new_data(f_features)

        # calculate age group wise avg data
        age = int(request.form.get("age"))
        age_group_avg_data = calculate_avg_data(age)

        return render_template("predict_diabetes.html",title="Assess Your Diabetes | DiabetIQ
        Insight",datas=[predicted_data, features, age_group_avg_data],)


@app.route("/explore_dataset", methods=["GET"])
def explore_dataset():
    data = pd.read_csv("Dataset/diabetes.csv")
    return render_template("dataset.html", title="Explore Dataset | DiabetIQ Insight", datas=data)


@app.route("/trained_models", methods=["GET"])
def trained_models():
    return render_template("trained_models.html", title="Trained Models | DiabetIQ Insight")


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

Step 2: Create Base Template for Web App



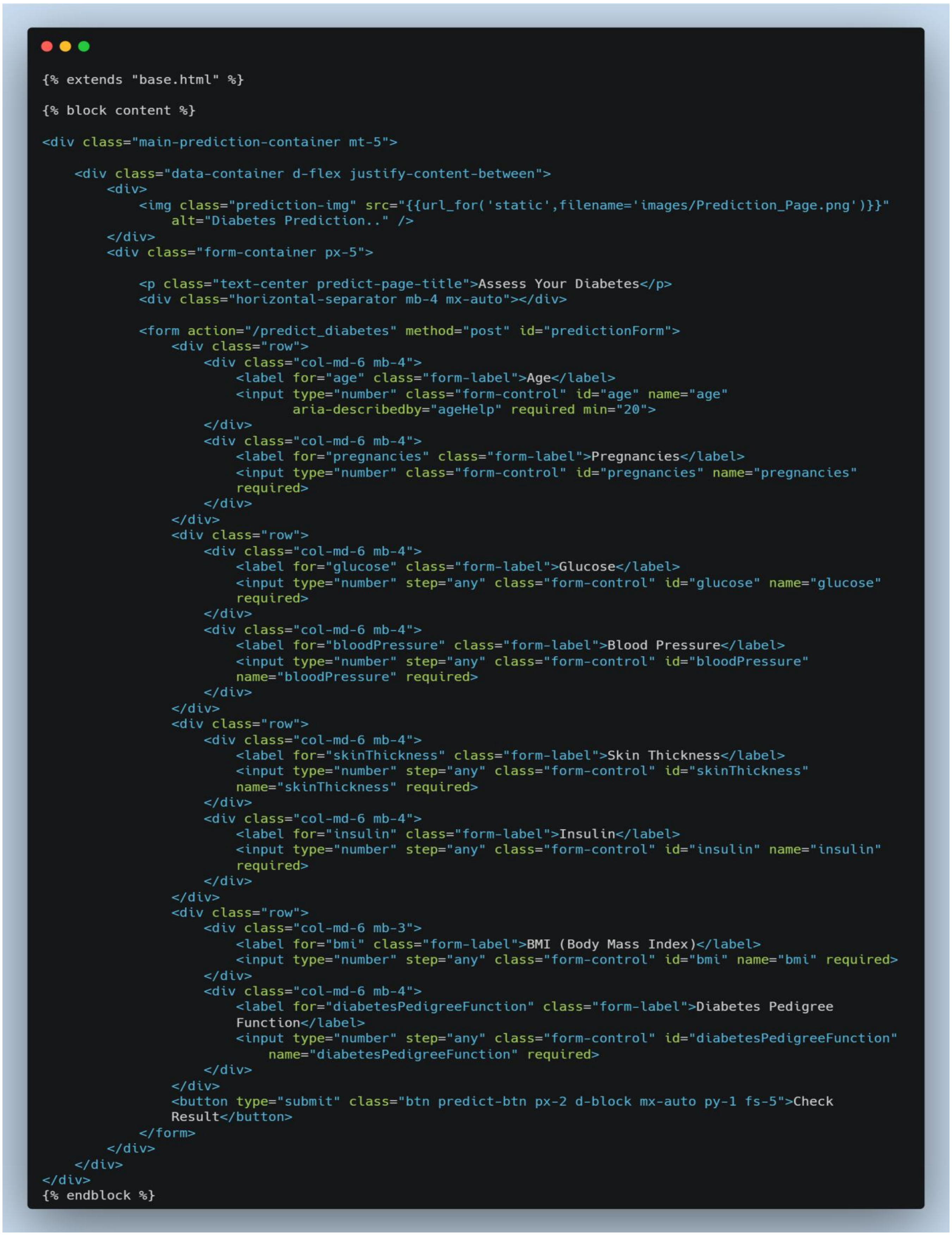
```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="author" content="Madhvik Bhalani">
  <title>{{title}}</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}" />
  <link rel="stylesheet" href="{{ url_for('static', filename='css/dataTables.bootstrap5.css') }}" />
  <link rel="stylesheet" href="{{ url_for('static', filename='highlight/styles/github-dark-dimmed.css') }}">
  <link rel="stylesheet" href="{{ url_for('static', filename='css/index.css') }}" />
  <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
  <link rel="icon" type="image/png" sizes="32x32" href="{{url_for('static',filename='favicon/favicon-
32x32.png')}}">
  <link rel="icon" type="image/png" sizes="16x16" href="{{url_for('static',filename='favicon/favicon-
16x16.png')}}">
</head>

<body>
  <nav class="navbar navbar-expand-lg px-2 d-flex justify-content-between align-items-center">
    <div class="log-container d-flex justify-content-between align-items-center">
      <a href="/">
        <img class="web-logo" src="{{ url_for('static', filename='images/logo.png') }}" alt="DiabetIQ
        Insight" />
      </a>
      <h4 class="text-light logo-name"><a href="/">DiabetIQ Insight</a></h4>
    </div>

    <div class="nav-link-container" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item dropdown mx-1">
          <a class="nav-link dropdown-toggle fs-5 nav-links" href="#" id="navbarDropdown" role="button"
            data-bs-toggle="dropdown" aria-expanded="false">
            Trained Models
          </a>

          <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="/trained_models#data_preprocessing">Data Preprocessing</a>
            </li>
            <li>
              <hr class="dropdown-divider" />
            </li>
            <li><a class="dropdown-item" href="/trained_models#logistic_regression">Logistic Regression</a>
            </li>
            <li>
              <a class="dropdown-item" href="/trained_models#k_nearest_neighbors">K-Nearest Neighbors (K-NN)
              </a>
            </li>
            <li>
              <a class="dropdown-item" href="/trained_models#svm">Support Vector Machine (SVM)</a>
            </li>
            <li><a class="dropdown-item" href="/trained_models#kernal_svm">Kernel SVM</a></li>
            <li><a class="dropdown-item" href="/trained_models#naive_bayes">Naive Bayes</a></li>
            <li>
              <a class="dropdown-item" href="/trained_models#decision_tree_classification">Decision Tree
                Classification</a>
            </li>
            <li>
              <a class="dropdown-item" href="/trained_models#random_forest_classification">Random Forest
                Classification</a>
            </li>
            <li><a class="dropdown-item" href="/trained_models#xg_boost">XGBoost</a></li>
          </ul>
        </li>
        <li class="nav-item mx-2">
          <a class="nav-link fs-5 nav-links" href="/explore_dataset">Dataset</a>
        </li>
      </ul>
    </div>
  </nav>

  {% block content %} {% endblock %}
  <!-- Render All Other's Page Content Here -->
  <hr class="hr mt-5">
  <footer class="footer text-center w-100">
    <div class="p-3">
      <h5 class="text-dark footer-text">
        © 2024: Designed and developed by <span class="name-link">
        <a href="https://www.linkedin.com/in/MadhvikBhalani/"
          class="text-dark linkdin-link" target="_blank">&lt;&#x2f;Madhvik Bhalani&gt;</a> </span>
      </h5>
    </div>
  </footer>

  <script src="{{ url_for('static', filename='js/bootstrap.bundle.min.js') }}"></script>
  <script src="{{ url_for('static', filename='js/jquery-3.7.1.min.js') }}"></script>
  <script src="{{ url_for('static', filename='js/dataTables.js') }}"></script>
  <script src="{{ url_for('static', filename='js/dataTables.bootstrap5.js') }}"></script>
  {% block dataset_table %} {% endblock %}

</body>

</html>
```

Step 3: Create Home Page



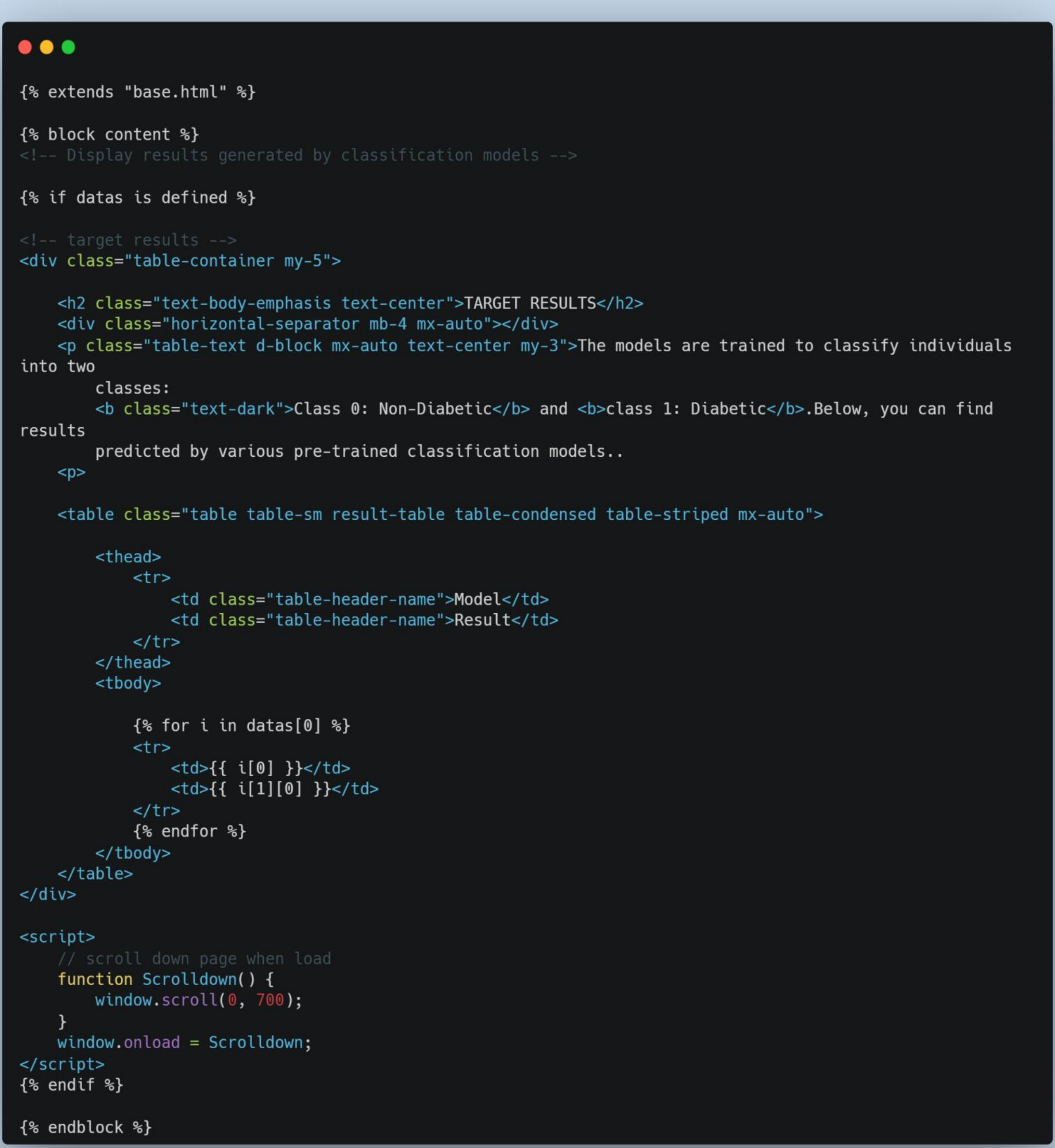
```
{% extends "base.html" %} {% block content %}
<section
  class=" home-container d-flex justify-content-between align-items-center mt-3" >
  <div class="mx-5">
    <h1 class="w-50">Predict Diabetes in Seconds!</h1>
    <p class="my-3 desc-box">
      In this project, i tackled a binary classification problem aiming to predict the
likelihood of an individual developing diabetes. Leveraging machine learning
      techniques, I explored various classification models including Logistic
      Regression, Decision Trees, Random Forests, and Support Vector
      Machine(SVM), among others. Extensive data preprocessing, feature
      engineering, and hyperparameter tuning were conducted to optimize model
      performance and predictive accuracy.
    </p>
    <button class="predict-btn"><a href="/predict_diabetes">Diabetes Assessment</a></button>
  </div>

  <div>
    <img
      class="home_img"
      src="{{url_for('static',filename='images/home_page.png')}}"
      alt="Home Page.."
    />
  </div>
</section>
{% endblock %}
```

Step 4: Create Web Form for Input Data



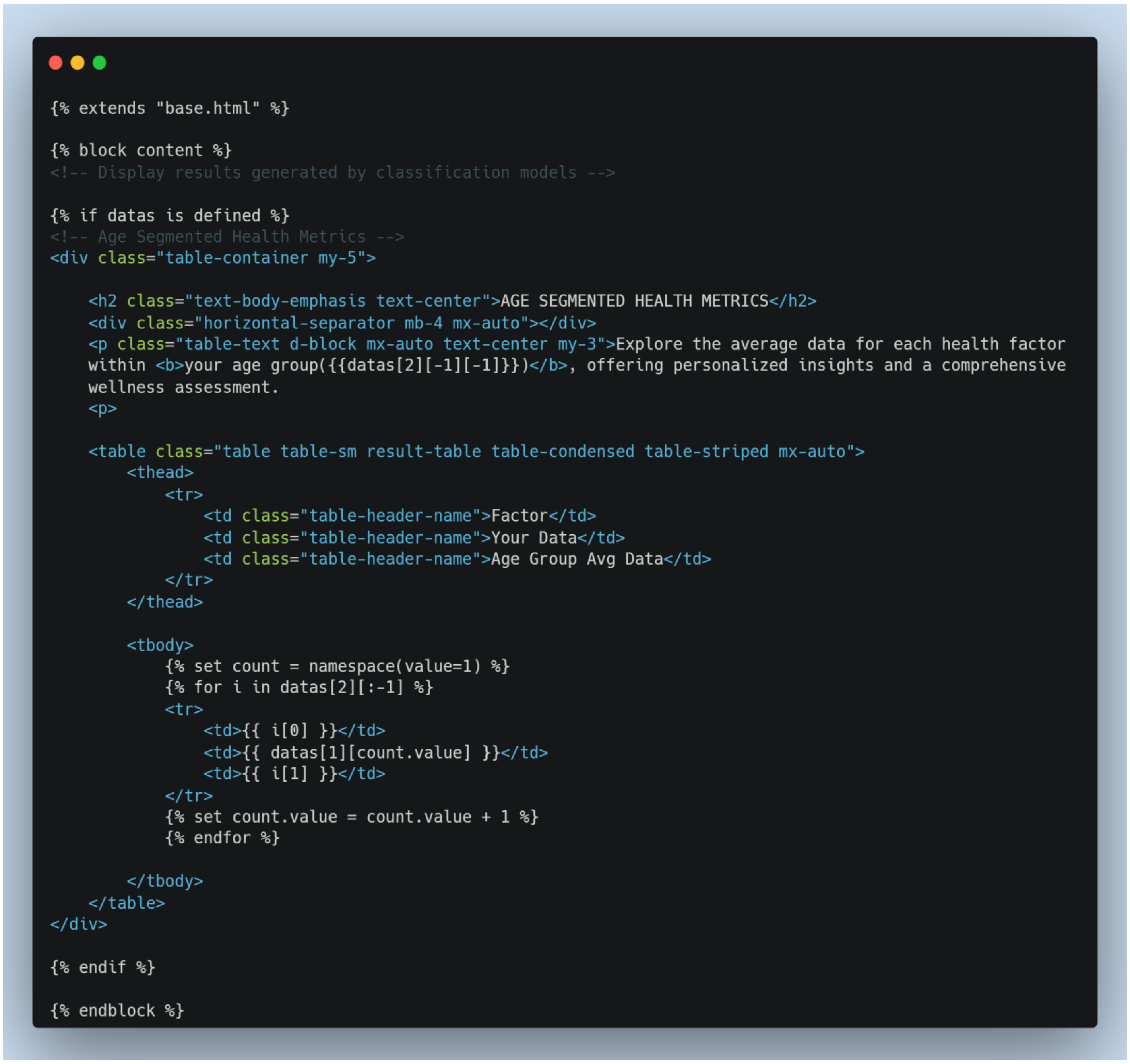
```
{% extends "base.html" %}

{% block content %}

<div class="main-prediction-container mt-5">

    <div class="data-container d-flex justify-content-between">
        <div>
            <img class="prediction-img" src="{{url_for('static',filename='images/Prediction_Page.png')}}"
                alt="Diabetes Prediction.." />
        </div>
        <div class="form-container px-5">

            <p class="text-center predict-page-title">Assess Your Diabetes</p>
            <div class="horizontal-separator mb-4 mx-auto"></div>

            <form action="/predict_diabetes" method="post" id="predictionForm">
                <div class="row">
                    <div class="col-md-6 mb-4">
                        <label for="age" class="form-label">Age</label>
                        <input type="number" class="form-control" id="age" name="age"
                               aria-describedby="ageHelp" required min="20">
                    </div>
                    <div class="col-md-6 mb-4">
                        <label for="pregnancies" class="form-label">Pregnancies</label>
                        <input type="number" class="form-control" id="pregnancies" name="pregnancies"
                        required>
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-6 mb-4">
                        <label for="glucose" class="form-label">Glucose</label>
                        <input type="number" step="any" class="form-control" id="glucose" name="glucose"
                        required>
                    </div>
                    <div class="col-md-6 mb-4">
                        <label for="bloodPressure" class="form-label">Blood Pressure</label>
                        <input type="number" step="any" class="form-control" id="bloodPressure"
                        name="bloodPressure" required>
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-6 mb-4">
                        <label for="skinThickness" class="form-label">Skin Thickness</label>
                        <input type="number" step="any" class="form-control" id="skinThickness"
                        name="skinThickness" required>
                    </div>
                    <div class="col-md-6 mb-4">
                        <label for="insulin" class="form-label">Insulin</label>
                        <input type="number" step="any" class="form-control" id="insulin" name="insulin"
                        required>
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-6 mb-3">
                        <label for="bmi" class="form-label">BMI (Body Mass Index)</label>
                        <input type="number" step="any" class="form-control" id="bmi" name="bmi" required>
                    </div>
                    <div class="col-md-6 mb-4">
                        <label for="diabetesPedigreeFunction" class="form-label">Diabetes Pedigree
                        Function</label>
                        <input type="number" step="any" class="form-control" id="diabetesPedigreeFunction"
                            name="diabetesPedigreeFunction" required>
                    </div>
                </div>
                <button type="submit" class="btn predict-btn px-2 d-block mx-auto py-1 fs-5">Check
                Result</button>
            </form>
        </div>
    </div>
</div>
{% endblock %}
```

Step 5: Visualize Prediction



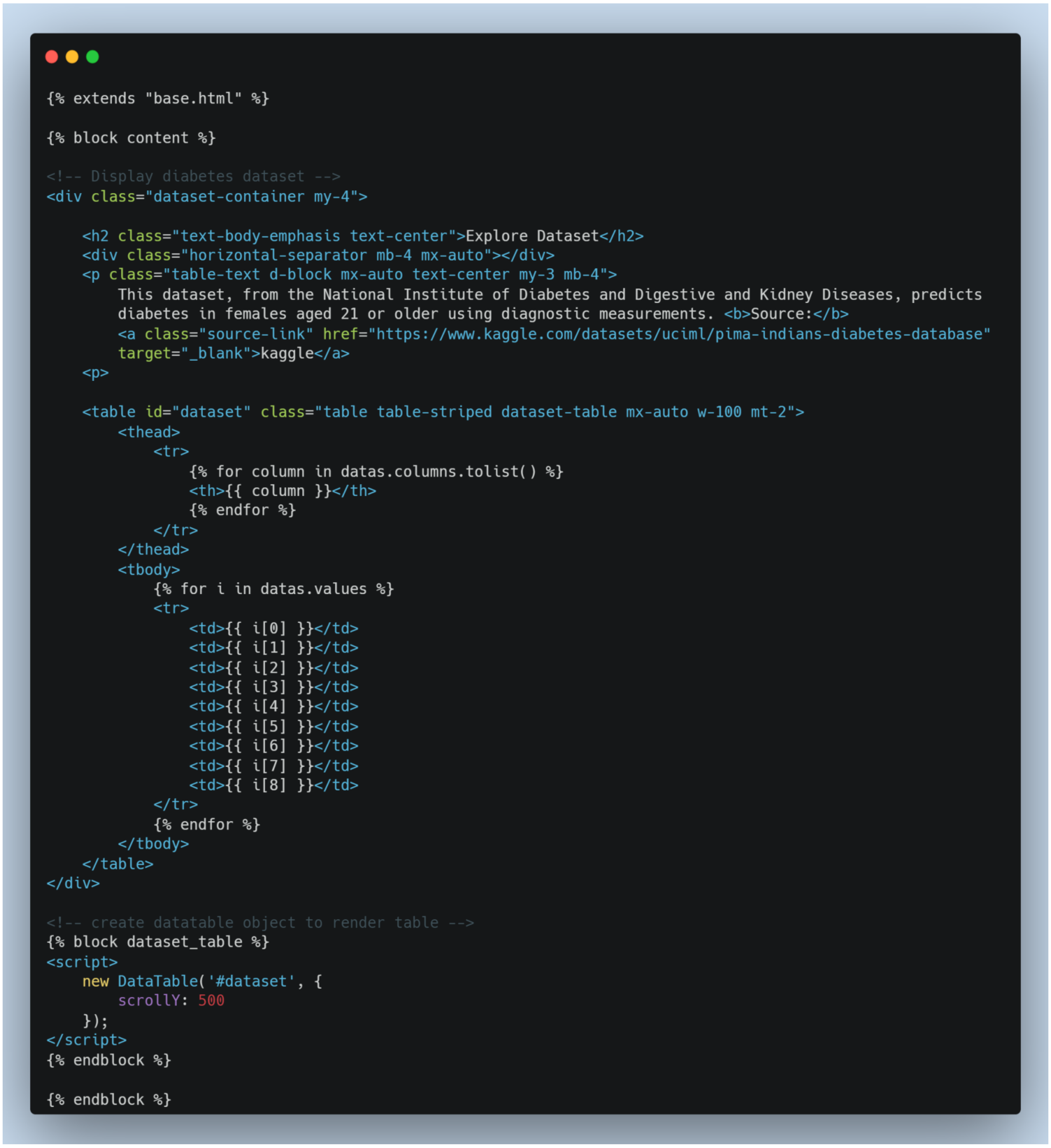
```
{% extends "base.html" %}

{% block content %}
<!-- Display results generated by classification models -->

{% if datas is defined %}

<!-- target results -->
<div class="table-container my-5">

    <h2 class="text-body-emphasis text-center">TARGET RESULTS</h2>
    <div class="horizontal-separator mb-4 mx-auto"></div>
    <p class="table-text d-block mx-auto text-center my-3">The models are trained to classify individuals
into two
        classes:
        <b class="text-dark">Class 0: Non-Diabetic</b> and <b>class 1: Diabetic</b>.Below, you can find
results
        predicted by various pre-trained classification models..
    <p>

    <table class="table table-sm result-table table-condensed table-striped mx-auto">

        <thead>
            <tr>
                <td class="table-header-name">Model</td>
                <td class="table-header-name">Result</td>
            </tr>
        </thead>
        <tbody>

            {% for i in datas[0] %}
            <tr>
                <td>{{ i[0] }}</td>
                <td>{{ i[1][0] }}</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>
</div>

<script>
    // scroll down page when load
    function Scrolldown() {
        window.scroll(0, 700);
    }
    window.onload = Scrolldown;
</script>
{% endif %}

{% endblock %}
```

Step 6: Visualize Age Segmented Health Metrics



```
{% extends "base.html" %}

{% block content %}
<!-- Display results generated by classification models -->

{% if datas is defined %}
<!-- Age Segmented Health Metrics -->
<div class="table-container my-5">

    <h2 class="text-body-emphasis text-center">AGE SEGMENTED HEALTH METRICS</h2>
    <div class="horizontal-separator mb-4 mx-auto"></div>
    <p class="table-text d-block mx-auto text-center my-3">Explore the average data for each health factor
    within <b>your age group({{datas[2][-1][-1]}})</b>, offering personalized insights and a comprehensive
    wellness assessment.
    <p>

    <table class="table table-sm result-table table-condensed table-striped mx-auto">
        <thead>
            <tr>
                <td class="table-header-name">Factor</td>
                <td class="table-header-name">Your Data</td>
                <td class="table-header-name">Age Group Avg Data</td>
            </tr>
        </thead>

        <tbody>
            {% set count = namespace(value=1) %}
            {% for i in datas[2][:-1] %}
            <tr>
                <td>{{ i[0] }}</td>
                <td>{{ datas[1][count.value] }}</td>
                <td>{{ i[1] }}</td>
            </tr>
            {% set count.value = count.value + 1 %}
            {% endfor %}

        </tbody>
    </table>
</div>

{% endif %}

{% endblock %}
```

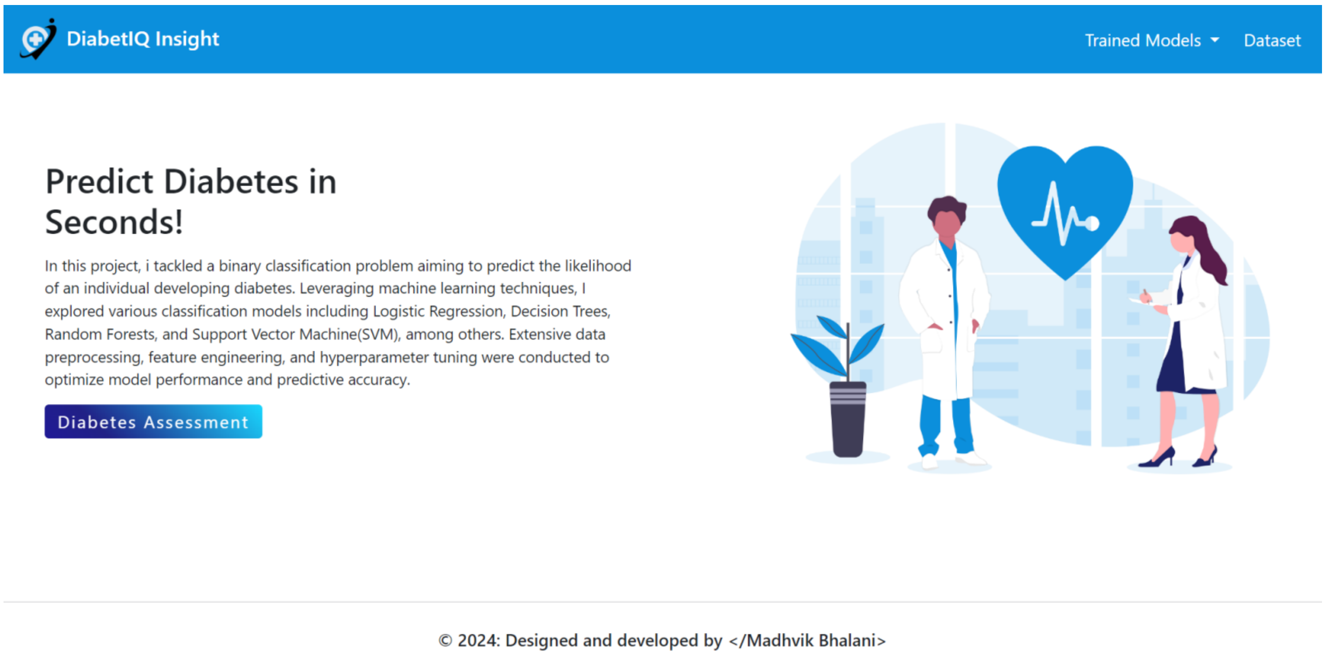
Step 7: Visualize Dataset



```
{% extends "base.html" %}

{% block content %}

<!-- Display diabetes dataset -->
<div class="dataset-container my-4">

    <h2 class="text-body-emphasis text-center">Explore Dataset</h2>
    <div class="horizontal-separator mb-4 mx-auto"></div>
    <p class="table-text d-block mx-auto text-center my-3 mb-4">
        This dataset, from the National Institute of Diabetes and Digestive and Kidney Diseases, predicts
        diabetes in females aged 21 or older using diagnostic measurements. <b>Source:</b>
        <a class="source-link" href="https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database"
        target="_blank">kaggle</a>
    <p>

    <table id="dataset" class="table table-striped dataset-table mx-auto w-100 mt-2">
        <thead>
            <tr>
                {% for column in datas.columns.tolist() %}
                <th>{{ column }}</th>
                {% endfor %}
            </tr>
        </thead>
        <tbody>
            {% for i in datas.values %}
            <tr>
                <td>{{ i[0] }}</td>
                <td>{{ i[1] }}</td>
                <td>{{ i[2] }}</td>
                <td>{{ i[3] }}</td>
                <td>{{ i[4] }}</td>
                <td>{{ i[5] }}</td>
                <td>{{ i[6] }}</td>
                <td>{{ i[7] }}</td>
                <td>{{ i[8] }}</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>
</div>

<!-- create datatable object to render table -->
{% block dataset_table %}
<script>
    new DataTable('#dataset', {
        scrollY: 500
    });
</script>
{% endblock %}

{% endblock %}
```

3. Snapshot of web page

i. Home page

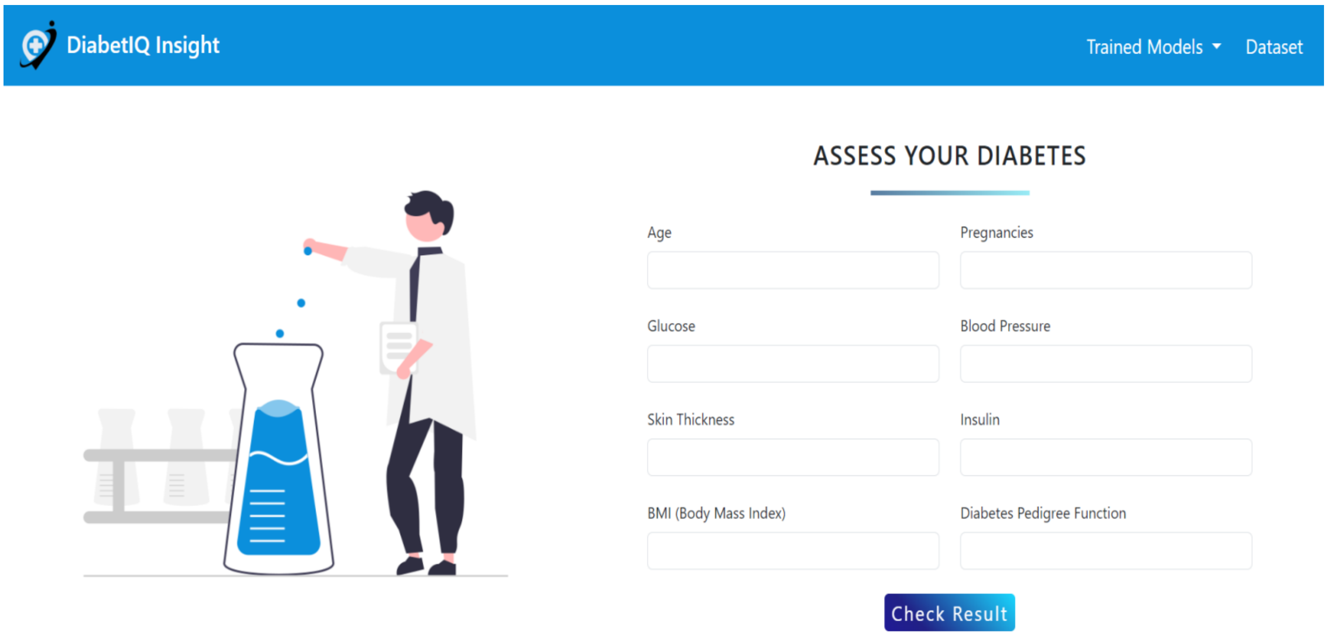


Predict Diabetes in Seconds!

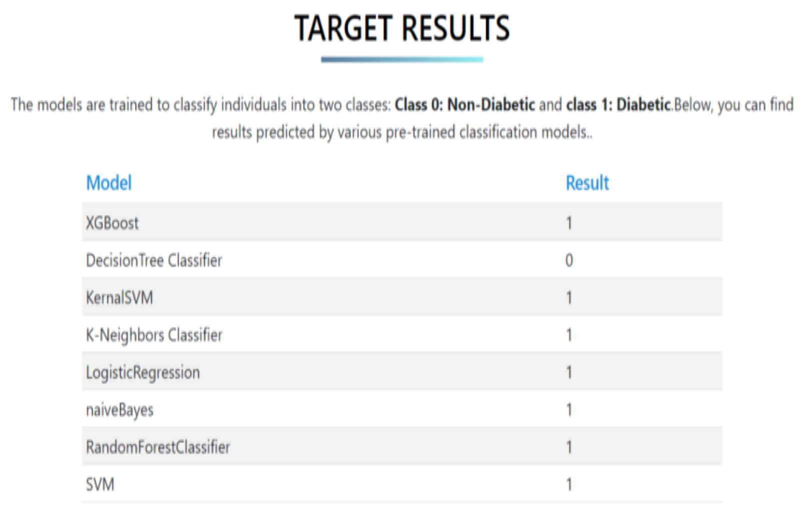
In this project, i tackled a binary classification problem aiming to predict the likelihood of an individual developing diabetes. Leveraging machine learning techniques, I explored various classification models including Logistic Regression, Decision Trees, Random Forests, and Support Vector Machine(SVM), among others. Extensive data preprocessing, feature engineering, and hyperparameter tuning were conducted to optimize model performance and predictive accuracy.

© 2024: Designed and developed by </Madhvik Bhalani>

ii. Diabetes Prediction form



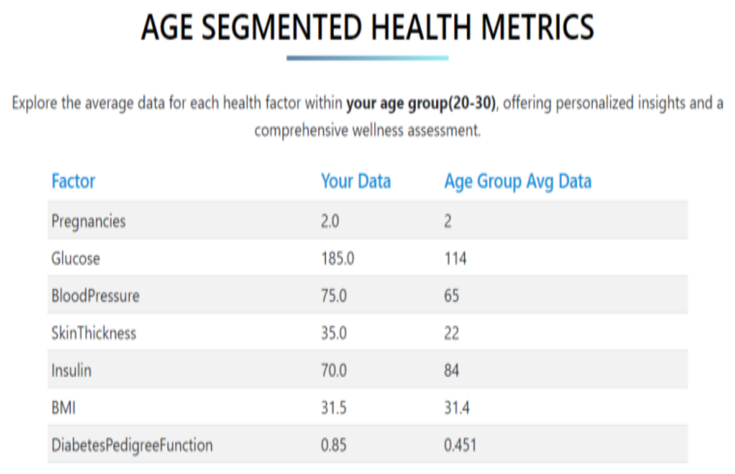
iii. Predicted Result and Age Segmented Health Metrics



TARGET RESULTS

The models are trained to classify individuals into two classes: **Class 0: Non-Diabetic** and **class 1: Diabetic**.Below, you can find results predicted by various pre-trained classification models..

Model	Result
XGBoost	1
DecisionTree Classifier	0
KernalSVM	1
K-Neighbors Classifier	1
LogisticRegression	1
naiveBayes	1
RandomForestClassifier	1
SVM	1



AGE SEGMENTED HEALTH METRICS

Explore the average data for each health factor within **your age group(20-30)**, offering personalized insights and a comprehensive wellness assessment.

Factor	Your Data	Age Group Avg Data
Pregnancies	2.0	2
Glucose	185.0	114
BloodPressure	75.0	65
SkinThickness	35.0	22
Insulin	70.0	84
BMI	31.5	31.4
DiabetesPedigreeFunction	0.85	0.451

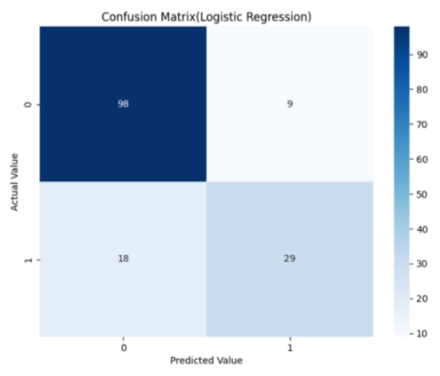
iv. Model data

Logistic Regression

Matrix

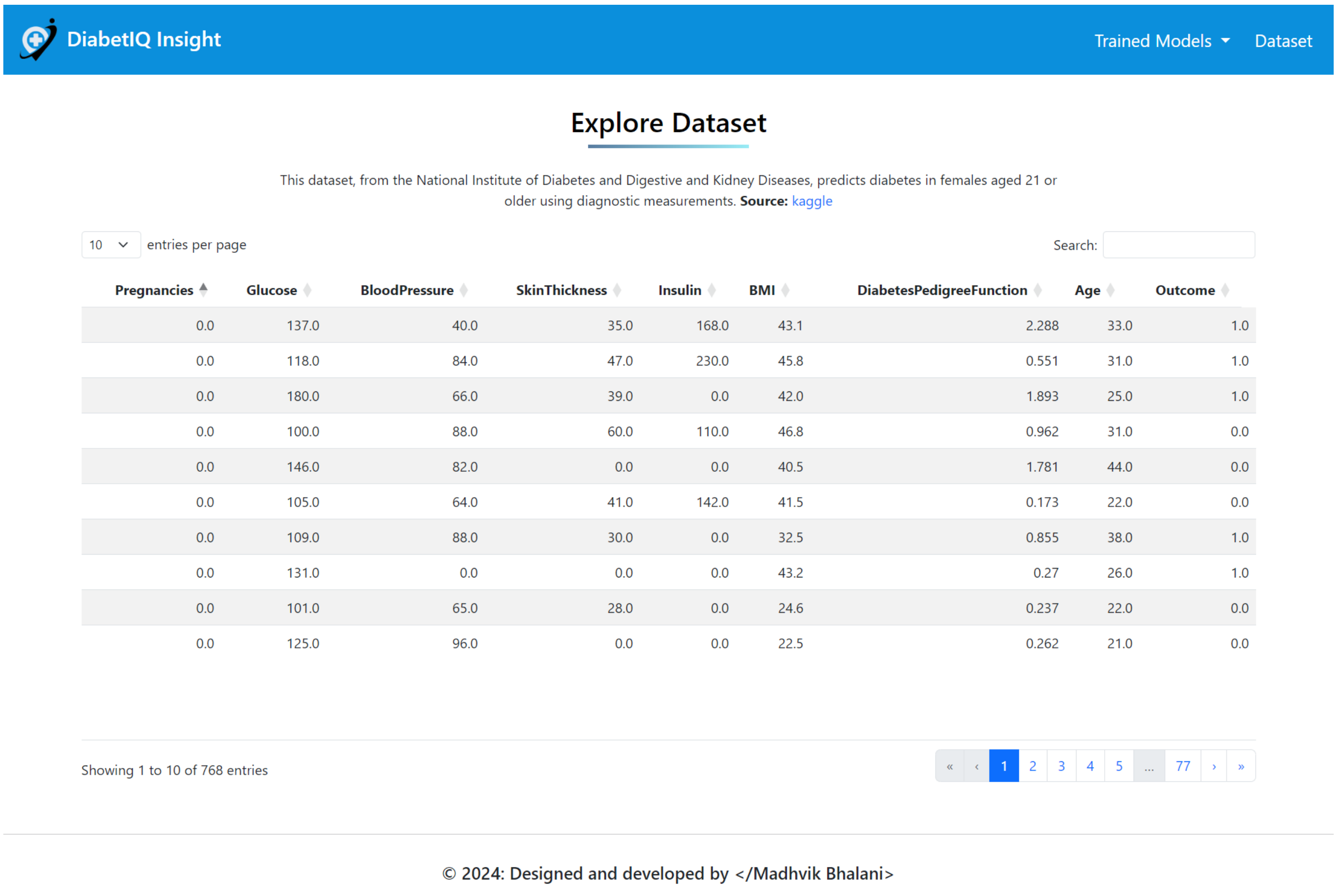
Metric	Value(%)
Accuracy	82.47
Precision	76.32
Recall	61.70
F1 score	68.24

Confusion Matrix



```python
from data_preprocessing import *
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score ,precision_score, recall_score, f1_score

# Training the Logistic Regression model on the Training set
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
y_pred = classifier.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
ac_score=accuracy_score(y_test, y_pred)
precision=precision_score(y_test, y_pred)
recall=recall_score(y_test, y_pred)
f1=f1_score(y_test, y_pred)

print(cm)
print("Accuracy: {:.2f}%".format(ac_score * 100))
print("Precision: {:.2f}%".format(precision * 100))
print("Recall: {:.2f}%".format(recall * 100))
print("F1 Score: {:.2f}%".format(f1 * 100))
```

v. Dataset



Explore Dataset

This dataset, from the National Institute of Diabetes and Digestive and Kidney Diseases, predicts diabetes in females aged 21 or older using diagnostic measurements. **Source:** kaggle

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0	1.0
0.0	118.0	84.0	47.0	230.0	45.8	0.551	31.0	1.0
0.0	180.0	66.0	39.0	0.0	42.0	1.893	25.0	1.0
0.0	100.0	88.0	60.0	110.0	46.8	0.962	31.0	0.0
0.0	146.0	82.0	0.0	0.0	40.5	1.781	44.0	0.0
0.0	105.0	64.0	41.0	142.0	41.5	0.173	22.0	0.0
0.0	109.0	88.0	30.0	0.0	32.5	0.855	38.0	1.0
0.0	131.0	0.0	0.0	0.0	43.2	0.27	26.0	1.0
0.0	101.0	65.0	28.0	0.0	24.6	0.237	22.0	0.0
0.0	125.0	96.0	0.0	0.0	22.5	0.262	21.0	0.0

Showing 1 to 10 of 768 entries

© 2024: Designed and developed by </Madhvik Bhalani>