Setting Up and Running the Disorder Prediction Project

Prerequisites

1. \*\*Install PostgreSQL and pgAdmin\*\*:

- Download and install PostgreSQL from the official website.

- Download and install pgAdmin from the official website.

2. \*\*Configuration During Installation\*\*:

- \*\*Port\*\*: Ensure that PostgreSQL is set to use port `587`.

- \*\*Password\*\*: Remember the password “53044” you set during installation. This password should match the password in the `settings.py` file located in the `disease\_prediction` folder.

3. \*\*Create Database\*\*:

- Open pgAdmin and log in using the password which is “53044” 8 you set during installation.

- Create a new database named `predico`.

Setting Up the Virtual Environment

1. \*\*Create a Virtual Environment\*\*:

- Navigate to your project directory.

- Create a virtual environment.

- Activate the virtual environment.

2. \*\*Install Dependencies\*\*:

- Ensure your virtual environment is activated.

- Install the necessary dependencies: `django`, `joblib`, `scikit-learn`, `psycopg2`.

Run pip install dependencies

django

joblib

scikit-learn

psycopg2

Database Migrations and Running the Server

1. \*\*Make Migrations\*\*:

- In your project directory, make the necessary migrations.

2. \*\*Apply Migrations\*\*:

- Apply the migrations to create the necessary tables in your PostgreSQL database.

- Run python manage.py makemigrations

- Run python manage.py migrate

- Run python manage.py runserver

- Navigate to http://127.0.0.1:8000/ in your browser

3. \*\*Run the Development Server\*\*:

- Start the Django development server.

Additional Configuration (needed)

- Ensure your `settings.py` file in the `disease\_prediction` folder contains the correct database configuration:

- Database name: `predico`

- Ensure the username and password match the ones used during PostgreSQL installation.

Summary of Steps

1. Install PostgreSQL and pgAdmin.

2. Configure PostgreSQL to use port 587 and remember the password.

3. Create a database named `predico`.

4. Create and activate a virtual environment.

5. Install the dependencies: `django`, `joblib`, `scikit-learn`, `psycopg2`.

6. Make and apply migrations.

7. Run the Django development server.

8. Navigate to `http://127.0.0.1:8000/` in your web browser.

**Tool Explanation**

**RandomForestClassifier**

The RandomForestClassifier is an ensemble learning method for classification that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) of the individual trees. It's part of the sklearn.ensemble module in the scikit-learn library, which provides various ensemble learning methods.

Key features of the RandomForestClassifier:

* **Ensemble Learning**: Combines multiple decision trees to improve classification accuracy and control over-fitting.
* **Bootstrap Aggregation (Bagging)**: Uses multiple subsets of the original dataset with replacement to train individual trees.
* **Feature Importance**: Helps in determining the importance of various features in making predictions.

**How I Imported Data**

1. **Data Preparation**:
   * The data is organized into a dictionary where keys are symptom names and the target variable (prognosis), and values are lists indicating the presence (1) or absence (0) of each symptom for different disorders.
2. **Data Dictionary**:
   * For instance, each entry in the dictionary for symptoms like fatigue, weight\_gain, etc., corresponds to a binary list where each element represents a sample.
   * The prognosis key contains the disorder names corresponding to each sample.
3. **Convert to DataFrame**:
   * This dictionary is then converted into a Pandas DataFrame for easier manipulation and analysis.
4. **Handling Missing Symptoms**:
   * Ensure that all symptoms in the predefined list are included in the DataFrame, filling in with zeros where necessary to match the length of existing data.