**Hackathon Environment Setup Documentation**

**1. Introduction**

This documentation will guide you through the environment setup for the hackathon project. The goal of this project is to build a **Player Performance Prediction Model** using player data and key performance indicators (KPIs). This environment setup is intended for both backend and data science operations, with libraries and tools optimized for data processing, machine learning, and web deployment.

**2. Prerequisites**

Before you start setting up the environment, make sure you have the following installed:

* **Python 3.8+**: Python is the primary programming language for the project.
* **Git**: For cloning the project repository and version control.
* **IDE (e.g., VSCode, PyCharm)**: Recommended for code development.
* **Jupyter Notebook (optional)**: If you're working with data analysis or machine learning models interactively.
* **An API Key**: Required for OpenAI models (if you are using GPT, for example).

**3. Setting Up the Environment**

Follow the steps below to set up your local development environment.

**Step 1: Install Python**

Ensure Python 3.8 or later is installed. You can check the version by running:

bash

Copy code

python --version

If Python is not installed, you can download and install it from [the official Python website](https://www.python.org/downloads/).

**Step 2: Set up a Virtual Environment**

Using a virtual environment helps isolate dependencies and ensures that your project doesn't conflict with other Python projects on your machine. To set up a virtual environment:

1. Create a directory for your project, if you haven't already.
2. Open the terminal (or command prompt) and navigate to your project folder.
3. Run the following command to create a virtual environment:

bash

Copy code

python -m venv venv

1. Activate the virtual environment:
   * **On Windows:**

bash

Copy code

.\venv\Scripts\activate

* + **On macOS/Linux:**

bash

Copy code

source venv/bin/activate

Once activated, your terminal should show the virtual environment name in parentheses.

**Step 3: Install Required Libraries**

The project uses several Python libraries, such as pandas, numpy, matplotlib, sklearn, flask, and others. To install them, you should create a requirements.txt file containing all the dependencies.

Example requirements.txt:

Copy code

pandas

numpy

matplotlib

seaborn

sklearn

flask

requests

openai

scipy

Run the following command to install all dependencies listed in the requirements.txt:

bash

Copy code

pip install -r requirements.txt

**Step 4: Clone the Project Repository**

To work on the project, you'll need to clone the hackathon repository from GitHub (or any other version control system you're using). Run the following command:

bash

Copy code

git clone https://github.com/yourusername/hackathon-repo.git

Navigate to the cloned repository folder:

bash

Copy code

cd hackathon-repo

**Step 5: Configure API Keys and Environment Variables**

For some functionalities (such as accessing OpenAI models or any third-party services), you'll need to configure API keys.

1. Create a .env file in the root of your project folder.
2. Add the required keys and secrets in the .env file (ensure you don't share this file publicly).

Example .env file:

makefile

Copy code

OPENAI\_API\_KEY=your\_openai\_api\_key

You can load these environment variables in Python using the python-dotenv library. To install it, run:

bash

Copy code

pip install python-dotenv

Then, load the environment variables in your Python code like this:

python

Copy code

from dotenv import load\_dotenv

import os

# Load environment variables from .env file

load\_dotenv()

# Access your API key

openai\_api\_key = os.getenv("OPENAI\_API\_KEY")

**4. Running the Project**

Once the environment is set up and the dependencies are installed, you can run the project. Depending on your project structure, this may involve:

* Running a Flask server for the web interface:

bash

Copy code

python app.py

* Running a Jupyter Notebook for data analysis and model building:

bash

Copy code

jupyter notebook

* Running any Python script for specific functionalities.

**5. Testing the Setup**

Ensure that everything is working correctly by running some basic tests:

1. **Check if the Flask server is running**: Visit http://127.0.0.1:5000 in your browser. You should see the application running.
2. **Test the prediction model**: Input sample data and check if the prediction model works correctly by running the respective script or interacting with the web interface.

**6. Troubleshooting**

Here are some common issues and solutions:

* **Issue**: Unable to activate virtual environment.
  + **Solution**: Ensure you're running the correct activation command depending on your OS.
* **Issue**: Missing library dependencies.
  + **Solution**: Run pip install -r requirements.txt to ensure all libraries are installed.
* **Issue**: API key not found.
  + **Solution**: Make sure the .env file is correctly configured and the environment variables are loaded.
* **Issue**: Flask app not running.
  + **Solution**: Check for any syntax errors in the app.py file and ensure the Flask library is installed.

**7. Conclusion**

By following this setup guide, you'll have the hackathon environment running locally, ready for use. Feel free to modify the environment setup or configuration based on your specific use case.

Good luck with your hackathon project!