Machine Learning Exercise Task: Weather Forecasting

Welcome to our machine learning exercise task! In this task, you will be working with a dataset containing one year's worth of weather data. Your goal is to build a machine learning model to forecast the weather for the next 2-3 days based on historical weather data.

Task Overview:

1. **Data Cleaning:**

    - Explore the dataset and identify any missing or inconsistent data.
    - Perform data cleaning procedures to handle missing values, outliers, and any other inconsistencies in the dataset.
    - Choose appropriate methods such as mean imputation, interpolation, or advanced imputation techniques based on the nature of the missing data.

2. **Feature Selection:**

    - Analyse the available features in the dataset and select relevant features that could influence weather forecasting.
    - Utilise feature selection techniques to identify the most important features for model training.

3. **Hyperparameter Tuning:**

    - Fine-tune the hyperparameters of the machine learning model to optimize its performance.

4. **Forecasting:**

    - Train the final model using the cleaned dataset and selected features.
    - Select a suitable algorithm for time series forecasting, considering factors such as seasonality, trend, and temporal dependencies in the data.
    - Build a machine learning model to forecast the temperature for the next 2-3 days.

Deliverables:

- A Jupyter Notebook or Python script containing your code for data cleaning, feature selection, data imputation, model development, hyperparameter tuning, and forecasting.
- Documentation explaining your approach, including any assumptions made and rationale behind your decisions.
- Forecast visualization.

Evaluation Criteria:

- Effectiveness of data cleaning and preprocessing techniques.
- Implementation of data imputation methods and handling of missing values.
- Selection of relevant features and justification for their inclusion.
- Quality of hyperparameter tuning and optimization strategies.
- Clarity and organization of code and documentation.

Submission Guidelines:

- Submit your completed task via a shared github repository by Wednesday, 6th of March (2024).
- Include all necessary files and documentation as outlined in the deliverables section.

Additional Optional Task:

Upon completing the forecast test, you have the option to extend your solution with an additional task:

1. Posting the Forecast to a Database:

After generating the weather forecast, consider implementing functionality to post the forecast data to a database(of your choice) for storage and retrieval. You may choose a suitable database system and design a schema to store forecast information, such as temperature, humidity, and other relevant parameters, for future reference or analysis.

Advanced Extra Optional Task:

For candidates looking to demonstrate advanced skills and knowledge, you can opt to complete the following task:

2. Creating a Simple RESTful API using Flask:

Develop a basic RESTful API using Flask to expose endpoints for accessing forecast data. The application should include the following routes:

- `GET /forecast`: This route should return a list of available forecast data.
- `POST /forecast`: Implement this route to allow posting new forecast data to the database.

You'll need to integrate your existing forecast generation code with Flask to enable posting and retrieving forecast data via the API.

Additional Notes:

- Feel free to reach out if you have any questions or need clarification on any aspect of the task.

- We're more interested in understanding your decision-making process, how you approach challenges, and your ability to adapt during the task than solely focusing on the forecast accuracy
- We encourage you to be creative and innovative in your approach, and we look forward to seeing your solutions!

Best of luck, and happy forecasting!