Documentation: Random Forest Training and Evaluation

Overview

This document outlines the process of training and evaluating a Random Forest model. It provides details on data preparation, model training, and evaluation metrics, along with the necessary functions and their descriptions.

Training Script: `rf_training.py`

1. Data Preparation

Function: `load_and_preprocess_data`

- **Purpose**: Loads and preprocesses the data for training.
- **Inputs**:
  - `file_path (str)`: Path to the CSV file containing the training data.
- **Outputs**:
  - `X (np.ndarray)`: Feature matrix.
  - `y (np.ndarray)`: Target vector.
- **Steps**:
  - Load the dataset using pandas.
  - Select specific columns as features and target.
  - Return the feature matrix and target vector.

2. Train Random Forest Model

Function: `train_random_forest_model`

- **Purpose**: Trains the Random Forest model using grid search for hyperparameter tuning.
- **Inputs**:
  - `X (np.ndarray)`: Feature matrix.
  - `y (np.ndarray)`: Target vector.
- **Outputs**:
  - `best_model`: Best Random Forest model obtained after grid search.
  - `best_params`: Best hyperparameters from grid search.
- **Steps**:
  - Initialize a RandomForestRegressor.
  - Define a grid of hyperparameters.
  - Use GridSearchCV to find the best parameters.

    - Train the model on the data and return the best model and parameters.

3. Save Model

Function: `save_model_to_file`

- **Purpose**: Saves the trained Random Forest model to a file using joblib.
- **Inputs**:
    - `model (RandomForestRegressor)`: Trained Random Forest model.
    - `filename (str)`: Name of the file to save the model.
- **Outputs**: None.
- **Steps**:
    - Save the model to the specified path using joblib.

4. Main Function

Function: `main`

- **Purpose**: Orchestrates the data loading, model training, and saving steps.
- **Inputs**:
    - `file_path (str)`: Path to the input data.
    - `model_filename (str)`: Name of the file to save the trained model.
- **Outputs**: None.
- **Steps**:
    - Load and preprocess the data.
    - Train the Random Forest model.
    - Save the trained model to a file.

Evaluation Script: `rf_eval.py`

1. Load Test Data

Function: `load_test_data`

- **Purpose**: Loads and preprocesses the test data.
- **Inputs**:
    - `file_path (str)`: Path to the CSV file containing the test data.
- **Outputs**:
    - `pd.DataFrame`: Preprocessed test data.
- **Steps**:
    - Load the dataset using pandas.
    - Convert date columns to datetime objects.
    - Return the processed DataFrame.

2. Load Random Forest Model

Function: `load_rf_model`

- **Purpose**: Loads a trained Random Forest model from a file.
- **Inputs**:
  - `model_path (str)`: Path to the saved model file.
- **Outputs**:
  - `RandomForestRegressor`: Loaded Random Forest model.
- **Steps**:
  - Use joblib to load the model from the specified path.

3. Calculate Metrics

Function: `calculate_metrics`

- **Purpose**: Calculates evaluation metrics for predictions.
- **Inputs**:
  - `y_true (np.ndarray)`: True values of the target variable.
  - `y_pred (np.ndarray)`: Predicted values of the target variable.
- **Outputs**:
  - `dict`: Dictionary containing evaluation metrics (MAE, MSE, RMSE, MAPE).
- **Steps**:
  - Compute MAE, MSE, RMSE, and MAPE.
  - Return the metrics in a dictionary.

4. Evaluate Random Forest Model

Function: `evaluate_rf`

- **Purpose**: Evaluates the Random Forest model on test data.
- **Inputs**:
  - `test_df (pd.DataFrame)`: Test data containing features and target variable.
  - `model (RandomForestRegressor)`: Trained Random Forest model.
  - `output_path (str, optional)`: Path to save predictions CSV file.
- **Outputs**:
  - `dict`: Evaluation metrics.
- **Steps**:
  - Extract features and target from the test data.
  - Scale the features as in training.
  - Predict using the Random Forest model.
  - Save predictions to a CSV file.
  - Calculate and print evaluation metrics.

5. Main Function

Function: `main`

- **Purpose**: Orchestrates the test data loading, model loading, and evaluation steps.
- **Inputs**:
    - `test_file_path (str)`: Path to the test data file.
    - `model_filename (str)`: Path to the trained Random Forest model file.
- **Outputs**: None.
- **Steps**:
    - Load the test data.
    - Load the trained model.
    - Evaluate the model on the test data.