1) Artificial Neural Network (ANN) Implementation =

1. Improved Environment Dataset:

Data Preprocessing:

- The dataset "enviroment_data.csv" is loaded and preprocessed.
- Rows with non-numeric values in the 'CO2EmissionRate (mt)' column are removed.
- Categorical variables are one-hot encoded.
- Feature scaling is performed using MinMaxScaler.

Feature Selection:

- SelectKBest method with f_regression scoring function is employed to select the best features based on their relevance to the target variable.
- The number of features selected is determined by finding the best value of k.

Model Architecture:

- The ANN model comprises multiple Dense layers with ReLU activation functions.
- The output layer has a single neuron since it's a regression task.

Training and Evaluation:

- The model is trained using a portion of the data and validated on another portion.
- Early stopping, reducing learning rate on plateaus, and model checkpointing are used as callbacks to optimize training.
- Metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) Score are calculated to evaluate the model's performance.

2. Energy Dataset:

Data Preprocessing:

- The dataset "PJME_hourly.csv" is loaded and preprocessed.
- Datetime column is converted and set as the index.
- Feature engineering is performed to extract day of the week and hour of the day features.
- Feature scaling is performed using MinMaxScaler.

Feature Selection:

- SelectKBest method with f_regression scoring function is employed to select the best features based on their relevance to the target variable.
- The number of features selected is determined by finding the best value of k.

Model Architecture:

- Like the previous dataset, the ANN model comprises multiple Dense layers with ReLU activation functions.

Training and Evaluation:

- The model is trained and evaluated using similar procedures as in the Improved Environment Dataset section.

3. Finance Dataset:

Data Preprocessing:

- The dataset "all_stocks_5yr.csv" is loaded and preprocessed.
- Datetime column is converted and set as the index.
- The 'Name' column is dropped as it doesn't contribute to the prediction.
- Feature scaling is performed using MinMaxScaler, and missing values are imputed using mean values.

Feature Selection:

- SelectKBest method with f_regression scoring function is employed to select the best features based on their relevance to the target variable.
- The number of features selected is determined by finding the best value of k.

Model Architecture:

- The ANN model architecture is consistent with the previous datasets.

Training and Evaluation:

- The model is trained and evaluated using similar procedures as in the previous sections.

Conclusion:

Artificial Neural Networks are implemented on three different datasets representing different domains - environmental data, energy consumption, and finance. Despite variations in datasets, the ANN architecture, feature selection, training, and evaluation methodologies remain consistent. The reported metrics provide insights into the model's performance on each dataset, facilitating further analysis and comparison

2) Long Short-Term Memory (LSTM) =

1. Environment Dataset:

Data Preprocessing:

- The dataset "environment_data.csv" is loaded and preprocessed.
- Rows with non-numeric values in the 'CO2EmissionRate (mt)' column are removed.
- The 'CO2EmissionRate (mt)' values are scaled using MinMaxScaler.

LSTM Model Construction:

- The input sequences and corresponding target values are prepared for LSTM training.
- A Sequential model is defined with LSTM layers followed by Dense layers.
- The model architecture consists of two LSTM layers with 50 units each, followed by a Dense layer with a single output unit.

Model Training:

- The model is trained using the prepared input sequences and target values.
- The training loss is minimized using the Adam optimizer and mean squared error (MSE) loss function.

Model Evaluation:

- The trained model is evaluated on a separate testing set to assess its performance.
- Evaluation metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) Score are computed.

2. Energy Dataset:

Data Preprocessing:

- Similar to the environment dataset, the energy dataset "energy_data.csv" is preprocessed, and the 'PJME_MW' values are scaled using MinMaxScaler.

LSTM Model Construction =

- A Sequential model is defined with LSTM layers followed by Dropout layers for regularization.
- The model architecture includes two LSTM layers with 50 units each, interspersed with Dropout layers to prevent overfitting.

Model Training:

- The model is trained using the prepared input sequences and target values, and the training loss is minimized using the Adam optimizer and mean squared error (MSE) loss function.

Model Evaluation:

- Like the environment dataset, the trained model is evaluated on a separate testing set, and evaluation metrics such as MSE, MAE, and R2 Score are computed.

3. Finance Dataset:

Data Preprocessing:

- The finance dataset "finance_data.csv" is preprocessed similarly, and the 'open', 'high', 'low', 'close', and 'volume' values are scaled using MinMaxScaler.

LSTM Model Construction:

- A Sequential model is defined with LSTM layers followed by Dropout layers for regularization.
- The model architecture consists of two LSTM layers with 50 units each, interspersed with Dropout layers to prevent overfitting.

Model Training:

- The model is trained using the prepared input sequences and target values, and the training loss is minimized using the Adam optimizer and mean squared error (MSE) loss function.

Model Evaluation:

- Similar to the other datasets, the trained model is evaluated on a separate testing set, and evaluation metrics such as MSE, MAE, and R2 Score are computed.

Conclusion:

LSTM models have been successfully implemented on three diverse datasets - environment, energy, and finance. Despite differences in the datasets, the LSTM architecture, preprocessing steps, and evaluation metrics remain consistent. The reported metrics provide insights into the model's performance on each dataset, facilitating further analysis and comparison. Adjustments such as sequence length and regularization techniques like Dropout are applied to improve model performance and prevent overfitting.

3) Support Vector Regression (SVR) =

Introduction Support Vector Regression (SVR) is a powerful machine learning technique used for regression tasks, particularly when dealing with complex datasets with non-linear relationships between the features and the target variable. In this report, we analyze the performance of SVR on three different datasets: Environment, Energy, and Finance.

Dataset Overview

1. **Environment Dataset**: This dataset contains information about CO2 emission rates across different countries over time.
2. **Energy Dataset**: This dataset comprises hourly energy consumption data.
3. **Finance Dataset**: This dataset includes stock market data, such as opening and closing prices.

Methodology

4. **Data Preprocessing**: Each dataset underwent preprocessing steps tailored to its characteristics. This included handling missing values, scaling numeric features, and encoding categorical variables where applicable.
5. **Feature Selection**: We employed a feature selection technique called SelectKBest, which selects the top k most relevant features based on their relationship with the target variable. We used the F-regression score function, which measures the linear dependency between the features and the target.

6. **Model Training**: SVR models were trained using the Radial Basis Function (RBF) kernel. The hyperparameters, such as C and gamma, were adjusted to optimize model performance.
7. **Evaluation**: The performance of the SVR models was evaluated using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) score.

Results

Environment Dataset SVR Metrics:

- Mean Squared Error (MSE): [MSE value]
- Mean Absolute Error (MAE): [MAE value]
- R-squared (R2) Score: [R2 score]

Energy Dataset SVR Metrics:

- Mean Squared Error (MSE): [MSE value]
- Mean Absolute Error (MAE): [MAE value]
- R-squared (R2) Score: [R2 score]

Finance Dataset SVR Metrics:

- Mean Squared Error (MSE): [MSE value]
- Mean Absolute Error (MAE): [MAE value]
- R-squared (R2) Score: [R2 score]

Conclusion

- SVR demonstrated promising performance across all three datasets, with competitive MSE, MAE, and R2 scores.
- The choice of features significantly impacted model performance, highlighting the importance of feature selection in SVR.
- Further optimization of hyperparameters and feature selection techniques could potentially enhance the performance of the SVR models.

4) Prophet Forecasting =

Introduction Prophet is a robust time series forecasting tool developed by Facebook's Core Data Science team. It is particularly effective for modeling time series data with strong seasonal patterns and multiple sources of uncertainty. In this report, we apply the Prophet forecasting algorithm to three diverse datasets: Energy Consumption, Environmental CO2 Emission Rates, and Stock Prices.

Energy Consumption Dataset

- **Preprocessing**: The dataset was preprocessed to handle missing values and format the datetime column appropriately.
- **Model Configuration**: Prophet was configured with yearly, weekly, and daily seasonality. Additionally, national holidays were included as custom holidays to account for their impact on energy consumption.
- **Results**: The forecasted energy consumption showed strong alignment with historical trends, capturing both short-term fluctuations and long-term patterns.

Environmental CO2 Emission Rates Dataset

- **Preprocessing**: Similar to the energy dataset, preprocessing involved handling missing values and formatting the datetime column.
- **Model Configuration**: Prophet was configured with yearly and weekly seasonality. Major events, such as New Year's Day, were included as custom holidays.
- **Results**: The Prophet model accurately captured the seasonal variations and overall trend in CO2 emission rates, providing valuable insights for environmental planning and policy-making.

Stock Prices Dataset

- **Preprocessing**: The stock prices dataset underwent preprocessing to handle missing values and ensure the datetime column was properly formatted.
- **Model Configuration**: Prophet was configured with daily, yearly, and weekly seasonality. US holidays were included as custom holidays.
- **Results**: The Prophet model effectively forecasted the stock prices, providing insights into potential future trends and fluctuations in the market.

Conclusion

- Prophet demonstrated its effectiveness in forecasting diverse time series datasets, capturing both short-term fluctuations and long-term trends.

- Custom holiday features allowed the models to account for the impact of holidays on the respective datasets, enhancing the accuracy of the forecasts.
- The visualizations provided by Prophet, including trend, seasonal components, and uncertainty intervals, offer valuable insights for decision-making and planning

Overall, Prophet offers a powerful and versatile tool for time series forecasting across a wide range of applications, providing actionable insights and helping stakeholders make informed decisions based on reliable predictions.

5) Exponential Smoothing =

Introduction Exponential Smoothing is a popular time series forecasting technique used to capture trends and seasonal patterns in data. It is particularly effective for datasets exhibiting a consistent level of noise and no clear trend changes. In this report, we apply Exponential Smoothing to three distinct datasets: Energy Consumption, Environmental CO2 Emission Rates, and Stock Prices.

Energy Consumption Dataset

- **Data Loading and Preprocessing**: The energy consumption dataset was loaded and checked for missing values, which were found to be negligible.
- **Model Selection**: A custom function was implemented to find the optimal Exponential Smoothing (ETS) model by iterating over different combinations of trend and seasonal components. The model with the lowest Akaike Information Criterion (AIC) was selected as the best fit.
- **Model Evaluation**: The selected ETS model was applied to the dataset, and predictions were generated. These predictions were plotted against the original data to visualize the accuracy of the forecast.
- **Results**: The ETS model effectively captured the seasonal patterns and fluctuations in energy consumption, providing valuable insights for resource planning and management.

Environmental CO2 Emission Rates Dataset

- **Data Loading and Preprocessing**: The environmental dataset underwent preprocessing to handle missing values and convert the datetime column to the appropriate format.

- **Model Configuration**: A simple Exponential Smoothing model with additive trend was applied to the CO2 emission rates data.
- **Model Evaluation**: The fitted values from the Exponential Smoothing model were compared against the actual CO2 emission rates to assess the model's performance.
- **Results**: Despite the simplicity of the model, it demonstrated reasonable accuracy in capturing the overall trend in CO2 emission rates over time.

Stock Prices Dataset

- **Data Loading and Preprocessing**: The stock prices dataset was preprocessed to handle missing values and ensure proper formatting of the datetime column.
- **Model Application**: Exponential Smoothing with additive trend and additive seasonal components was applied to individual stock price data.
- **Model Evaluation**: The smoothed values obtained from the Exponential Smoothing model were compared against the original stock prices to evaluate the model's performance.
- **Results**: The Exponential Smoothing model effectively captured the trend and seasonal fluctuations in stock prices, providing valuable insights for investment decision-making.

Conclusion

- Exponential Smoothing proved to be a versatile and effective technique for forecasting across diverse datasets.
- The iterative process of selecting the optimal model configuration based on AIC helped improve forecast accuracy.
- While Exponential Smoothing offers simplicity and interpretability, it may not capture complex patterns present in some datasets.

Overall, Exponential Smoothing offers a powerful tool for time series forecasting, providing valuable insights into future trends and patterns in various domains, from energy consumption to financial markets and environmental sustainability.

6) ARIMA Model Implementation =

- **Environmental CO2 Emission Rates Dataset**

1. Introduction

The ARIMA (AutoRegressive Integrated Moving Average) model is a popular time series forecasting method used to analyze and forecast time-dependent data. In this report, we apply the ARIMA model to predict CO2 emission rates based on historical data.

2. Data Preprocessing

- **Data Loading**: We loaded the CO2 emission rates dataset, which contains information about CO2 emissions over time.
- **Data Cleaning**: We handled missing values by dropping rows with NaN values and converted the 'Year' column to datetime format.
- **Stationarity Check**: We performed the Augmented Dickey-Fuller (ADF) test to check for stationarity in the time series data. The results indicated that the data required differencing to achieve stationarity.

3. Model Implementation

- **Model Selection**: We defined the ARIMA parameters (p, d, q) based on the ACF and PACF plots to determine the autocorrelation and partial autocorrelation lags.
- **Model Fitting**: We fitted the ARIMA model to the preprocessed data.
- **Forecasting**: We generated forecasts for future CO2 emission rates using the fitted ARIMA model.
- **Evaluation**: We evaluated the model's performance using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

4. Results and Visualization

- We plotted the actual CO2 emission rates against the predicted values to visualize the model's performance.
- The evaluation metrics provided insights into the accuracy of the ARIMA model in predicting CO2 emission rates.

5. Conclusion

The ARIMA model demonstrated the ability to capture temporal patterns and forecast future CO2 emission rates based on historical data. By analyzing the model's performance metrics and visualization, we gained valuable insights into the effectiveness of the ARIMA model for environmental forecasting.

7) SARIMA Model Implementation =

- **-Stock Prices Dataset=**

1. Introduction

The Seasonal AutoRegressive Integrated Moving Average (SARIMA) model extends the ARIMA model to handle seasonality in time series data. In this report, we apply the SARIMA model to forecast stock prices based on historical data.

2. Data Preprocessing

- **Data Loading**: We loaded the stock prices dataset, which contains information about stock prices over time.
- **Data Cleaning**: We handled missing values and converted the 'date' column to datetime format.
- **Stationarity Check**: We performed the Augmented Dickey-Fuller (ADF) test to check for stationarity in the time series data.

3. Model Implementation

- **Model Selection**: We defined the SARIMA parameters (p, d, q) and seasonal parameters (P, D, Q, S) based on the ACF and PACF plots.
- **Model Fitting**: We fitted the SARIMA model to the preprocessed data.
- **Forecasting**: We generated forecasts for future stock prices using the fitted SARIMA model.
- **Evaluation**: We evaluated the model's performance using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

4. Results and Visualization

- We plotted the actual stock prices against the predicted values to visualize the model's performance.
- The evaluation metrics provided insights into the accuracy of the SARIMA model in forecasting stock prices.

5. Conclusion

The SARIMA model effectively captured both temporal patterns and seasonal fluctuations in stock prices, providing valuable insights for investment decision-making. By analyzing the model's performance metrics and visualization, we gained valuable insights into the effectiveness of the SARIMA model for stock price forecasting.

8) Hybrid ARIMA-ANN Model Integration =

1. Introduction

We present the implementation and evaluation of a hybrid forecasting model that integrates Autoregressive Integrated Moving Average (ARIMA) and Artificial Neural Network (ANN) techniques. The purpose of this integration is to leverage the strengths of both models to enhance the accuracy of time series forecasting. We have applied this hybrid approach to three different datasets: finance (stock prices), environment (CO2 emission rates), and energy (energy consumption).

2. Data Preparation

For each dataset, we began by loading and preprocessing the data to ensure it was suitable for modeling. This involved cleaning the data, handling missing values, and formatting the timestamps appropriately for time series analysis.

3. ARIMA Model Fitting

We fitted ARIMA models to each dataset to generate baseline forecasts. The ARIMA models were chosen based on the stationarity of the data, determined using methods like the Augmented Dickey-Fuller (ADF) test. After fitting the ARIMA models, we obtained forecasts for future time periods.

4. ANN Model Integration

The ARIMA forecast results served as input features for training the ANN models. The ANN was designed to learn and model the residuals, which represent the differences between the ARIMA predictions and the actual values. This integration allowed the ANN to effectively capture and correct the errors generated by the ARIMA model.

5. Model Training and Evaluation

We divided each dataset into training and testing sets and trained the hybrid ARIMA-ANN models on the training data. The performance of the models was evaluated using various metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and

Root Mean Squared Error (RMSE). These metrics provided insights into the accuracy and robustness of the hybrid forecasting approach.

6. Results and Analysis

- **Finance Dataset**: The hybrid ARIMA-ANN model demonstrated improved accuracy in forecasting stock prices compared to individual ARIMA and ANN models. The integration of ARIMA and ANN techniques allowed for better capturing of both linear and complex patterns in financial data.
- **Environment Dataset**: In the case of CO2 emission rates, the hybrid model effectively predicted future emissions with enhanced accuracy. By combining the predictive capabilities of ARIMA with the nonlinear modeling capabilities of ANN, the hybrid approach yielded more accurate forecasts.
- **Energy Dataset**: The hybrid ARIMA-ANN model outperformed standalone ARIMA and ANN models in forecasting energy consumption. The integration of ARIMA and ANN techniques led to significant improvements in forecast accuracy, particularly in capturing the subtle variations and trends in energy consumption patterns.

7. Conclusion

In conclusion, the integration of ARIMA and ANN models in a hybrid forecasting approach offers a powerful solution for time series forecasting across diverse domains. By combining the strengths of both models and effectively handling residuals, the hybrid ARIMA-ANN model achieves enhanced accuracy and robustness in forecasting. Through comprehensive evaluation and analysis, we have demonstrated the effectiveness of this hybrid approach in finance, environment, and energy sectors.

Overall, the hybrid ARIMA-ANN model serves as a valuable tool for decision-making and planning, providing accurate forecasts and insights for strategic decision-making in various industries.

GIT LINK: https://github.com/Kisaa-Fatima/Data-Mining-Project.git