Report: Sales Prediction Using SARIMA and Deep Learning Models

1. Introduction

Sales forecasting is a critical aspect of business planning and decision-making. Accurate predictions of future sales enable businesses to manage inventory, plan marketing strategies, and optimize operational efficiency. This project aims to enhance sales forecasting accuracy by leveraging both traditional time series models and advanced deep learning techniques. Specifically, we use the Seasonal AutoRegressive Integrated Moving Average (SARIMA) model and various Recurrent Neural Network (RNN) architectures, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Additionally, an ensemble approach combining SARIMA and LSTM predictions is explored to further improve forecasting performance.

2. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to gain initial insights into the sales data and to guide subsequent modeling efforts. Key steps and findings from the EDA are as follows:

2.1 Data Visualization

- **Time Series Plot**: The sales data was plotted over time to visualize the trend and seasonal patterns. This plot revealed a clear upward trend in sales with noticeable seasonal fluctuations.

2.2 Statistical Summary

- **Descriptive Statistics**: Basic statistics such as mean, median, standard deviation, and range were calculated to understand the central tendency and dispersion of the sales data.

2.3 Trend and Seasonality

- **Decomposition**: The time series was decomposed into trend, seasonal, and residual components using additive decomposition. This helped in understanding the underlying patterns in the data.

2.4 Stationarity Check

- **Augmented Dickey-Fuller Test**: The ADF test was conducted to check for stationarity. The null hypothesis of the presence of a unit root was rejected, indicating that the data was non-stationary. Differencing was applied to make the data stationary.

2.5 Autocorrelation and Partial Autocorrelation

- **ACF and PACF Plots**: The ACF and PACF plots were analyzed to identify the presence of autocorrelation and to determine the orders of ARIMA components. The plots suggested significant autocorrelations at various lags.

3. Methodology

3.1 Data Preparation

The sales data used in this project was preprocessed to handle missing values and ensure stationarity. The key steps included:

- **Handling Missing Values**: Missing values were imputed using forward fill or other appropriate methods.
- **Differencing**: Differencing was applied to the time series data to achieve stationarity, which is crucial for time series modeling.

3.2 SARIMA Model

The SARIMA model is a powerful statistical tool that captures seasonality, trends, and autocorrelations in time series data. The model parameters were selected based on:

- **Autocorrelation (ACF) and Partial Autocorrelation (PACF)**: Plots of ACF and PACF were used to identify the appropriate orders of the ARIMA components (p, d, q) and the seasonal components (P, D, Q, m).
- **Model Fitting and Validation**: The SARIMA model was fitted to the training data and validated using various statistical metrics such as AIC and BIC.

3.3 Deep Learning Models

3.3.1 LSTM

LSTM networks are designed to capture long-term dependencies in sequential data. The architecture included:

- **Input Layer**: Feeding the time series data into the network.
- **LSTM Layers**: Multiple LSTM layers to capture temporal patterns.
- **Dense Layer**: A dense layer to produce the final output.
- **Parameter**: used moving window as 24

3.3.2 GRU

GRU networks, similar to LSTM, are capable of capturing complex temporal dependencies with a simpler architecture. The GRU model was designed with:

- **Input Layer**: Similar preprocessing as LSTM.
- **GRU Layers**: Multiple GRU layers to learn from the sequential data.
- **Dense Layer**: Final layer for prediction.

3.3.3 RNN

The basic RNN model served as a baseline for comparison. The architecture was:

- **Input Layer**: Pre-processed time series data.
- **RNN Layers**: Stacked RNN layers to learn temporal patterns.
- **Dense Layer**: Output layer for the forecast.

3.4 Ensemble Model

An ensemble model combining SARIMA and LSTM predictions was created to leverage the strengths of both approaches. The ensemble prediction was calculated as:

$$\text{ensemble\_predictions} = 0.8 \times \text{sarima\_predictions} + 0.2 \times \text{lstm\_predictions}$$

ensemble_predictions=0.8×sarima_predictions+0.2×lstm_predictions

4. Results

4.1 Loss Functions

The training and validation loss functions for LSTM, GRU, and RNN models were plotted to evaluate their performance. The loss plots indicate the convergence and stability of the models over epochs.

4.2 Prediction Results

The actual sales data was plotted against the predictions from LSTM, GRU, RNN, and the ensemble model. The plots demonstrate how closely each model's predictions match the actual sales.

4.3 Performance Evaluation

- **SARIMA**: Captured seasonality and trends well but struggled with capturing complex patterns.
- **LSTM**: Provided accurate forecasts by learning long-term dependencies.
- **GRU**: Achieved similar performance to LSTM with a simpler architecture.
- **RNN**: Served as a baseline and showed lower accuracy compared to LSTM and GRU.
- **Ensemble Model**: Improved accuracy by combining the linear and non-linear patterns captured by SARIMA and LSTM, respectively.

5. Conclusion

This project demonstrates that combining traditional time series models with deep learning approaches can significantly enhance sales forecasting accuracy. The ensemble approach of SARIMA and LSTM leveraged the strengths of both models, resulting in more robust predictions. Future work could explore other ensemble techniques and incorporate external factors such as promotional events or economic indicators to further improve forecasting performance.

6. References

- Hyndman, R.J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.
- Chollet, F. (2017). Deep Learning with Python. Manning Publications.
- Seasonal ARIMA with Python: https://www.statsmodels.org/stable/examples/notebooks/generated/statespace_sarimax_stata.html