Supply Chain Management – Demand Forecasting Report

Project Overview

This project focuses on optimizing the supply chain of a Fashion & Beauty startup by predicting product demand using a machine learning model. The goal was to analyze historical sales and supply chain features, engineer the dataset for forecasting, and implement a deep learning model to forecast future product demand.

Tools and Technologies Used

- Language – Python
- Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, TensorFlow/Keras
- ML Technique: Neural Network (Regression)
- Other Tools: Jupyter Notebook

Dataset Overview

- Dataset is collected from a Fashion and Beauty Startup and is based on supply of chain of Makeup products.
- Rows – 100 and Columns – 24
- Features already present – Product type, SKU, Price, Availability, Number of products sold, Revenue generated, Customer demographics, Stock levels, Order quantities, Shipping times, Shipping carriers, Shipping costs, Supplier name, Location, Lead time, Production volumes, Manufacturing lead time, Manufacturing costs, Inspection results, Defect rates, Transportation modes, Routes, Costs
- Features Added – Date, Promotion, Weather, EconomicIndicators

Data Preprocessing

- Converted Date to datetime and extracted Month, DayOfWeek, and Quarter
- Added synthetic columns for Promotion, Weather, and EconomicIndicators
- One-hot encoded categorical variables
- Standardized features using StandardScaler
- Split data into 80% training and 20% testing sets

Data Cleaning and Feature Engineering

- We checked for missing values by using .isnull().sum() to verify that there were no null values in the dataset.
- We then renamed SKU to ProductID and Number of Products sold to HistoricalSales for consistency.
- For Feature Engineering,
    1. We extracted new features like Month, Day, etc. from Date to help the model understand seasonal patterns and then dropped the Date column as it was no longer needed.
    2. We used One-Hot encoding using pd.get_dummies() to convert categorical variables like Weather, EconomicIndicators, etc. into binary vectors.
    3. We used StandardScaler to scale all numeric features as it helps with faster and more stable training.
    4. Then we split the dataset into Training set (80%) and Test set (20%) using train_test_split() from scikit-learn.

Exploratory Data Analysis (EDA)

Some following facts and observations were revealed –

- Revenue and cost varied significantly between product types and locations.
- Longer lead times often led to higher defect rates.
- Promotions helped to boost sales.

Model Building

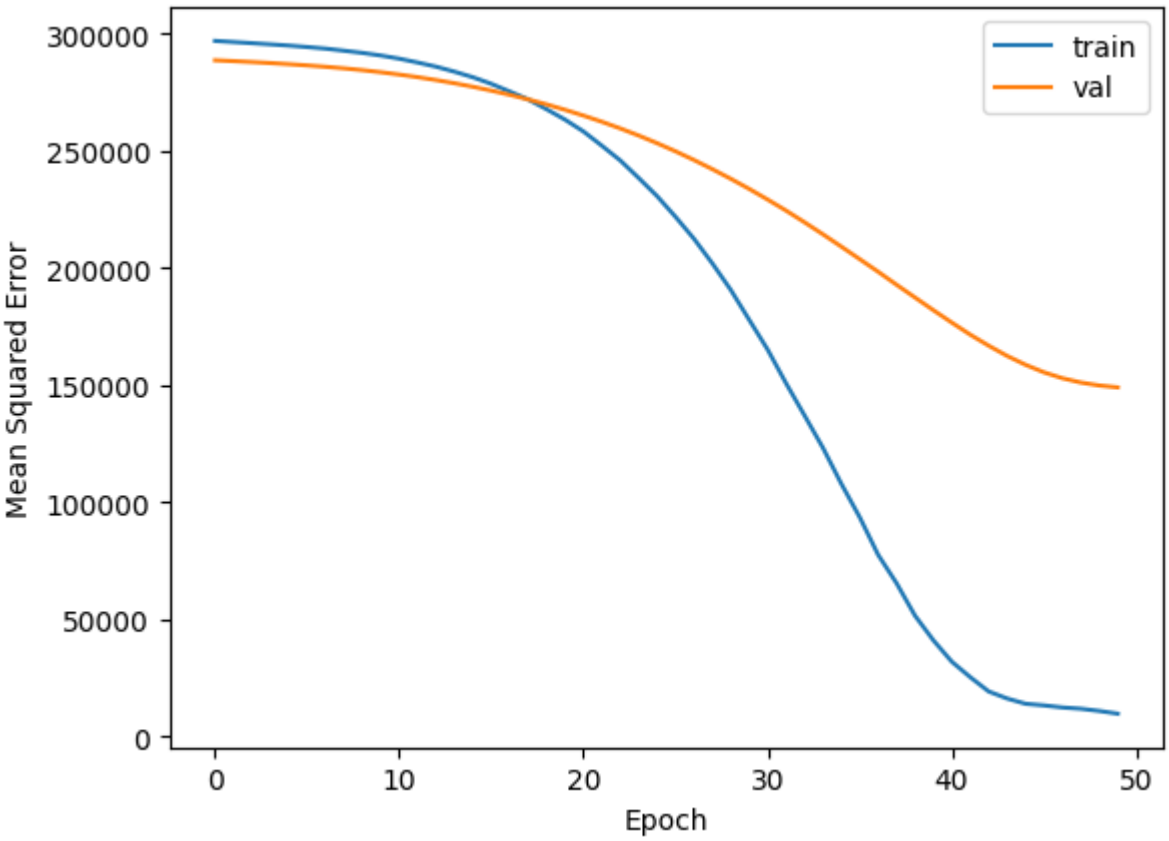
- We used a Neural Network with three hidden layers.
- The model was trained for 50 epochs.
- Structure: Input -> Dense(128) -> Dense(64) -> Dense(32) -> Output

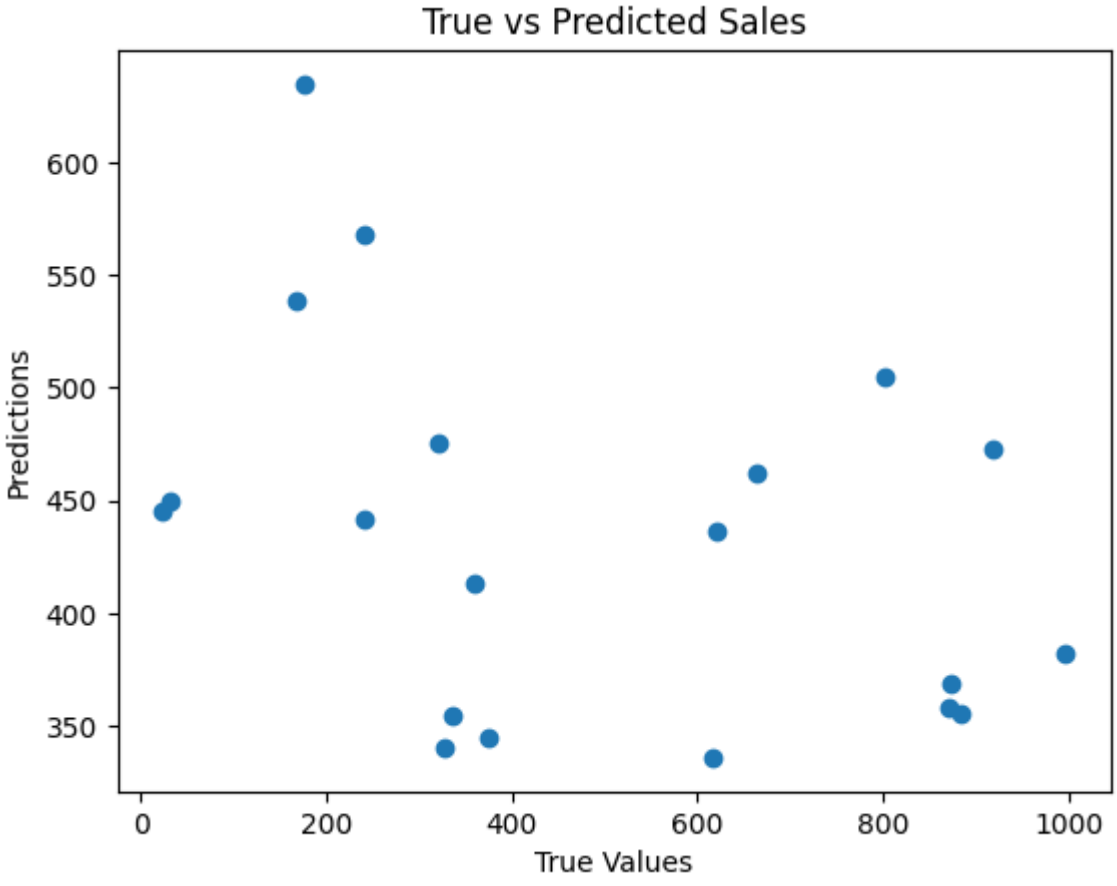
Model Evaluation and Performance

- The Metric used was Mean Squared Error (MSE).
- We used a scatter plot of true vs. predicted sales to show good predictive performance.
- Final MSE on Test set – 124400.2734375

Model Deployment

- The trained model was saved as ‘demand_forecasting_model.keras’.
- It can be reloaded and used for future predictions on new data.


train
val
Mean Squared Error
Epoch
300000
250000
200000
150000
100000
50000
0
0
10
20
30
40
50


True vs Predicted Sales
Predictions
True Values
600
550
500
450
400
350
0
200
400
600
800
1000

Conclusion

- The project successfully shows how ML can be applied to solve real-world supply chain challenges.
- Business Impact –
    1. Optimization on Inventory levels
    2. Reduction in risk of overstock or stockout
    3. Improvement in Customer Satisfaction