GITHUB LINK: https://github.com/MAHISHABANU1508/SALES-FORECASTING-USING-HISTORICAL-DATA.git

PROJECT TITLE

SALES FORECASTING USING HISTORICAL DATA

Phase-1

Student Name: MAHISHA BANU A

Register Number: 23132241802522019

Institution: Sacred Heart Arts and Science College, Perani

Department: BSc., Computer Science.

Date of Submission:

1.Problem Statement

Sales forecasting with historical data involves predicting future sales by analyzing past trends, seasonality, and patterns, with common problem statements focusing on improving inventory, resource allocation, and strategy using time-series models (ARIMA, LSTM) or regression, often dealing with challenges like changing product/store lists, incorporating external factors, and handling volatile markets for robust, accurate predictions. Key goals are minimizing stockouts/waste and maximizing cash flow by building models that adapt to real-world complexities. Accurate sales forecasting is crucial for effective inventory management, demand planning, and business decision-making. Traditional forecasting methods often fail to consider external factors such as holidays, which significantly influence customer purchasing behavior.The problem is to

forecast future sales using historical sales data while incorporating holiday events to improve prediction accuracy.

2.Abstract

Sales forecasting using historical data analyzes past sales records to predict future trends, identifying patterns like seasonality to improve inventory, resource allocation, and strategic planning, often employing time-series models (ARIMA, LSTM, SARIMA) or machine learning (XGBoost) for accuracy, sometimes enhanced with external factors like online reviews or search data for robust, data-driven predictions. Sales forecasting plays a vital role in retail and business analytics. This project aims to develop a time-series forecasting model using historical sales data combined with holiday event information. The study utilizes data preprocessing, exploratory data analysis (EDA), feature engineering, and the SARIMAX (Seasonal AutoRegressive Integrated Moving Average with Exogenous Variables) model. Holiday events are treated as external regressors to capture seasonal demand fluctuations. The model forecasts future sales and helps businesses plan resources more effectively. The results demonstrate that including holiday information improves forecasting performance.

3.System Requirements

Specify minimum system/software requirements to run the project:

Hardware:

- Processor: Intel i3/i5 or AMD equivalent
- RAM: Minimum 4 GB
- Storage: 10 GB free space

Software:

- Operating System: Windows 10
- Python Version: 3.8 or above
- Tools: Google Colab (preferred for free GPU and easy setup)

Libraries Used

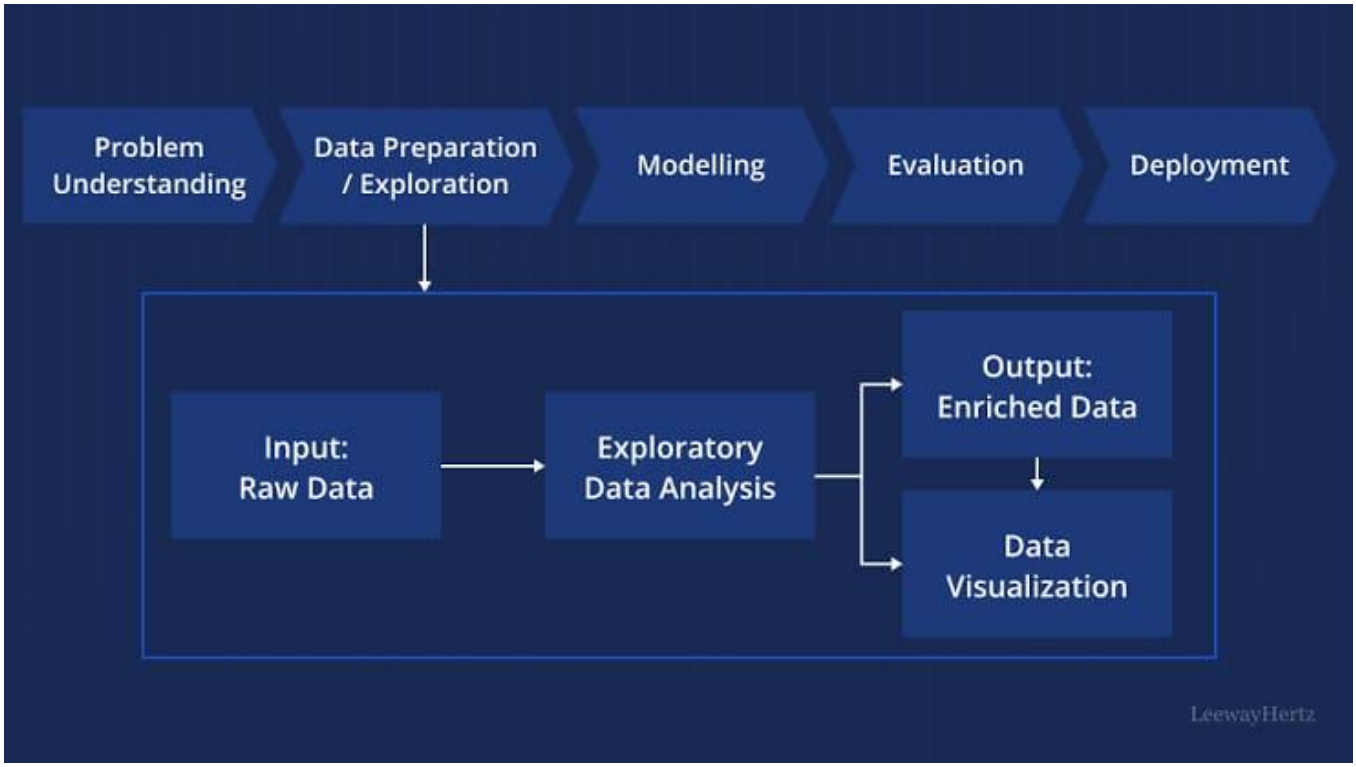
- Pandas
- NumPy
- Matplotlib
- Statsmodels

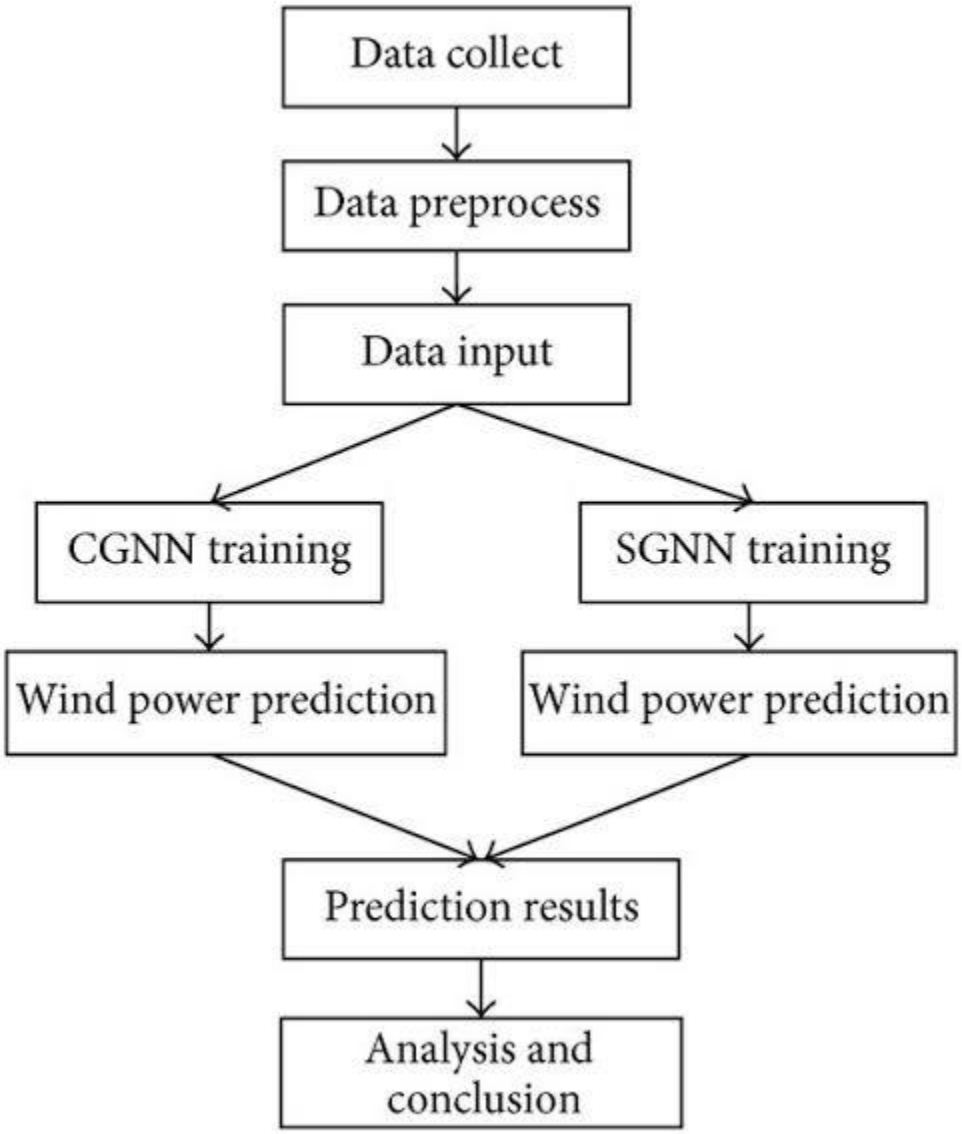
4.Objectives

- To analyze historical sales data
- To understand the impact of holiday events on sales
- To preprocess and clean raw datasets
- To perform exploratory data analysis (EDA)
- To build a time-series forecasting model
- To forecast future sales accurately
- To assist decision-making using predictive analytics

5.Flowchart of Project Workflow

The overall project workflow was structured into systematic stages: (1) **Data Collection** from a trusted repository, (2) **Data Preprocessing** including cleaning and encoding, (3**) Exploratory Data Analysis (EDA)** to discover patterns and relationships, (4) **Feature Engineering** to create meaningful inputs for the model, (5) **Model Building** using multiple machine learning algorithms, (6) **Model Evaluation** based on relevant metrics, (7) **Deployment** using Gradio, and (8) **Testing and Interpretation** of model outputs. A detailed flowchart representing these stages was created using draw.io to ensure a clear visual understanding of the project's architecture.


Problem Understanding
Data Preparation / Exploration
Modelling
Evaluation
Deployment
Input: Raw Data
Exploratory Data Analysis
Output: Enriched Data
Data Visualization
LeewayHertz



6.Dataset Description

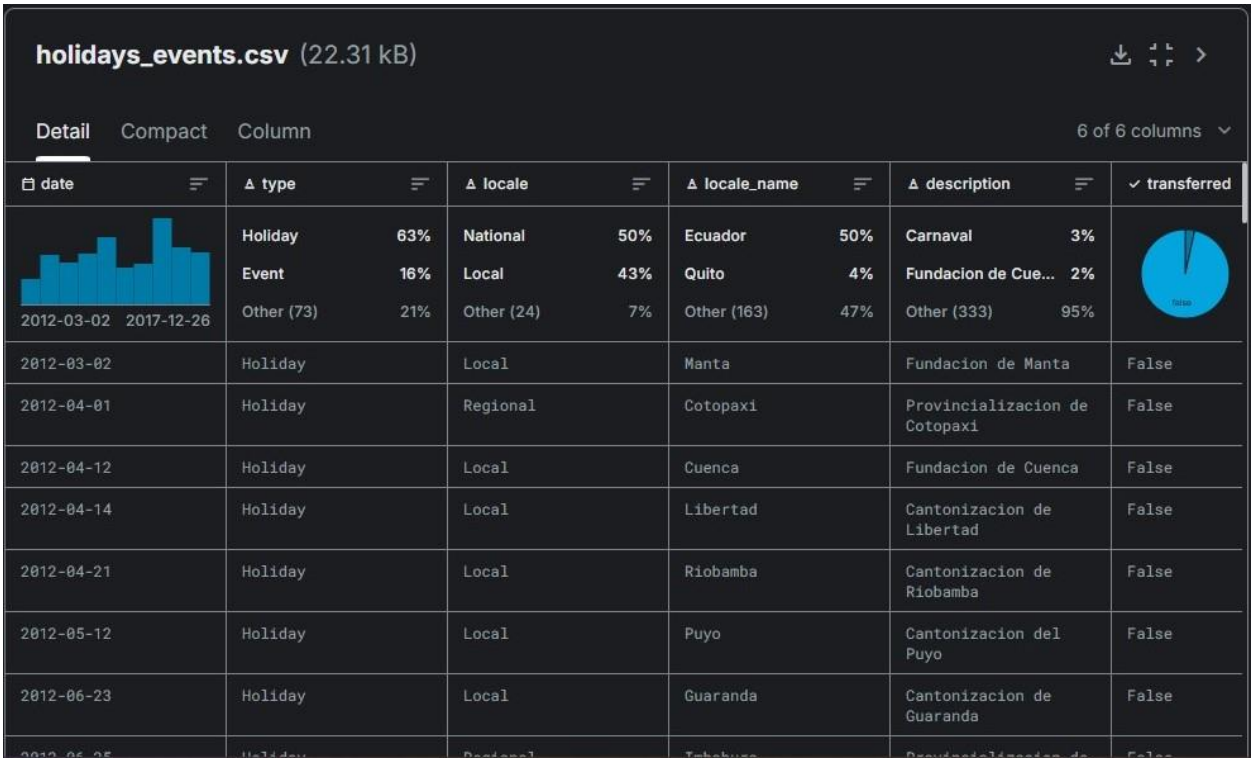
- ***Source*** (Kaggle Store Sales Dataset (`holidays_event.csv`))
- ***Type*** (structured data, time series data)

- ***Size and structure***

Holiday Dataset

1. Contains holiday-related records
2. Rows: ~350 records
3. Columns:
    - `date` – Holiday date
    - `type` – Holiday type
    - `locale` – Holiday level (National/Regional)
    - `description` – Holiday name
    - `transferred` – Holiday transfer status

- ***Include `df.head()` screenshot***



holidays_events.csv (22.31 kB)

Detail | Compact | Column | 6 of 6 columns

date	type	locale	locale_name	description	transferred
2012-03-02 2017-12-26	Holiday 63% Event 16% Other (73) 21%	National 50% Local 43% Other (24) 7%	Ecuador 50% Quito 4% Other (163) 47%	Carnaval 3% Fundacion de Cue... 2% Other (333) 95%	
2012-03-02	Holiday	Local	Manta	Fundacion de Manta	False
2012-04-01	Holiday	Regional	Cotopaxi	Provincializacion de Cotopaxi	False
2012-04-12	Holiday	Local	Cuenca	Fundacion de Cuenca	False
2012-04-14	Holiday	Local	Libertad	Cantonizacion de Libertad	False
2012-04-21	Holiday	Local	Riobamba	Cantonizacion de Riobamba	False
2012-05-12	Holiday	Local	Puyo	Cantonizacion del Puyo	False
2012-06-23	Holiday	Local	Guaranda	Cantonizacion de Guaranda	False

7. Data Preprocessing

- **Missing Values**: None detected.
- **Duplicates**: Checked and none found.
- **Outliers:** Converted date columns into datetime format.
- **Encoding:** Created a binary holiday indicator (`is_holiday`).
- Aggregated sales data on a daily basis.
- Filtered national-level holidays.
- Merged sales data with holiday data.

- **Show before/after transformation screenshots**

```
✅ Libraries imported successfully!
        date     type    locale locale_name                  description  \
0 2012-03-02  Holiday     Local       Manta             Fundacion de Manta
1 2012-04-01  Holiday  Regional    Cotopaxi  Provincializacion de Cotopaxi
2 2012-04-12  Holiday     Local      Cuenca           Fundacion de Cuenca
3 2012-04-14  Holiday     Local    Libertad      Cantonizacion de Libertad
4 2012-04-21  Holiday     Local    Riobamba      Cantonizacion de Riobamba

   transferred
0        False
1        False
2        False
3        False
4        False
```

8.Exploratory Data Analysis (EDA)

Variables Analyzed:

- `unit_sales`
- `date`
- `is_holiday`
- `onpromotion`

Bivariate / Multivariate Analysis

This analysis studies the **relationship between two or more variables**.

Scatter Plots

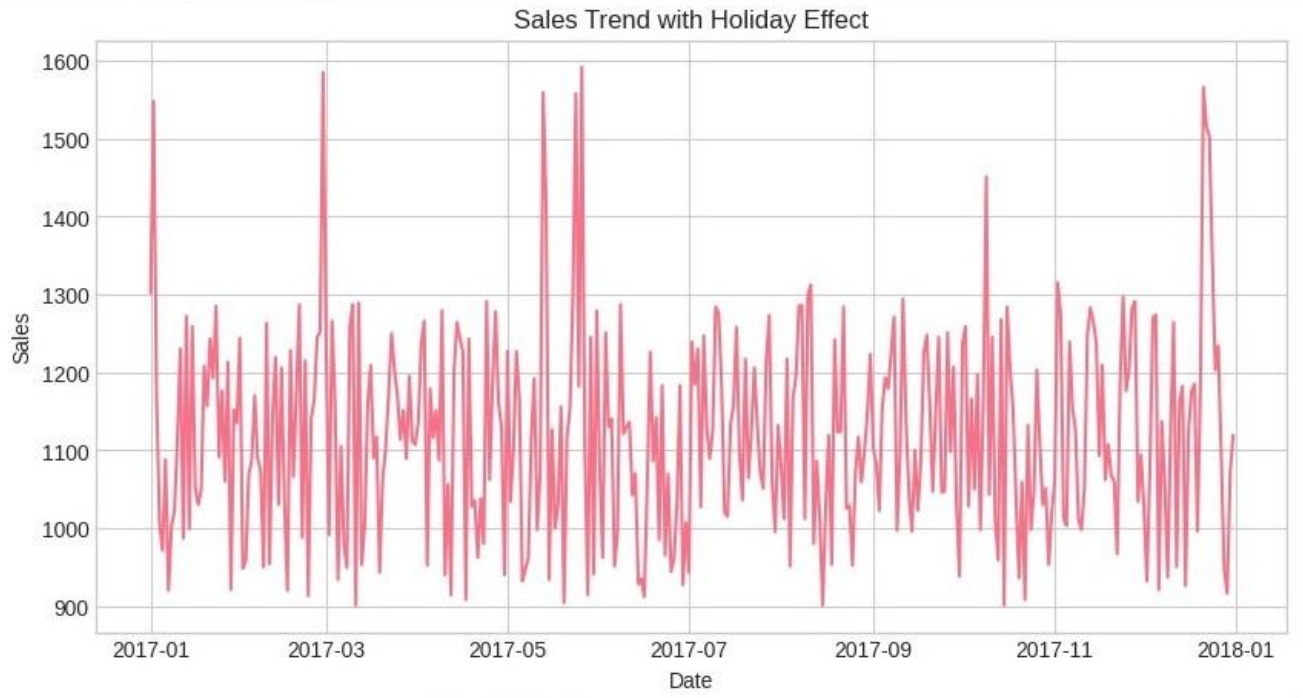
Scatter plots are used to visualize **relationships between sales and influencing factors**.

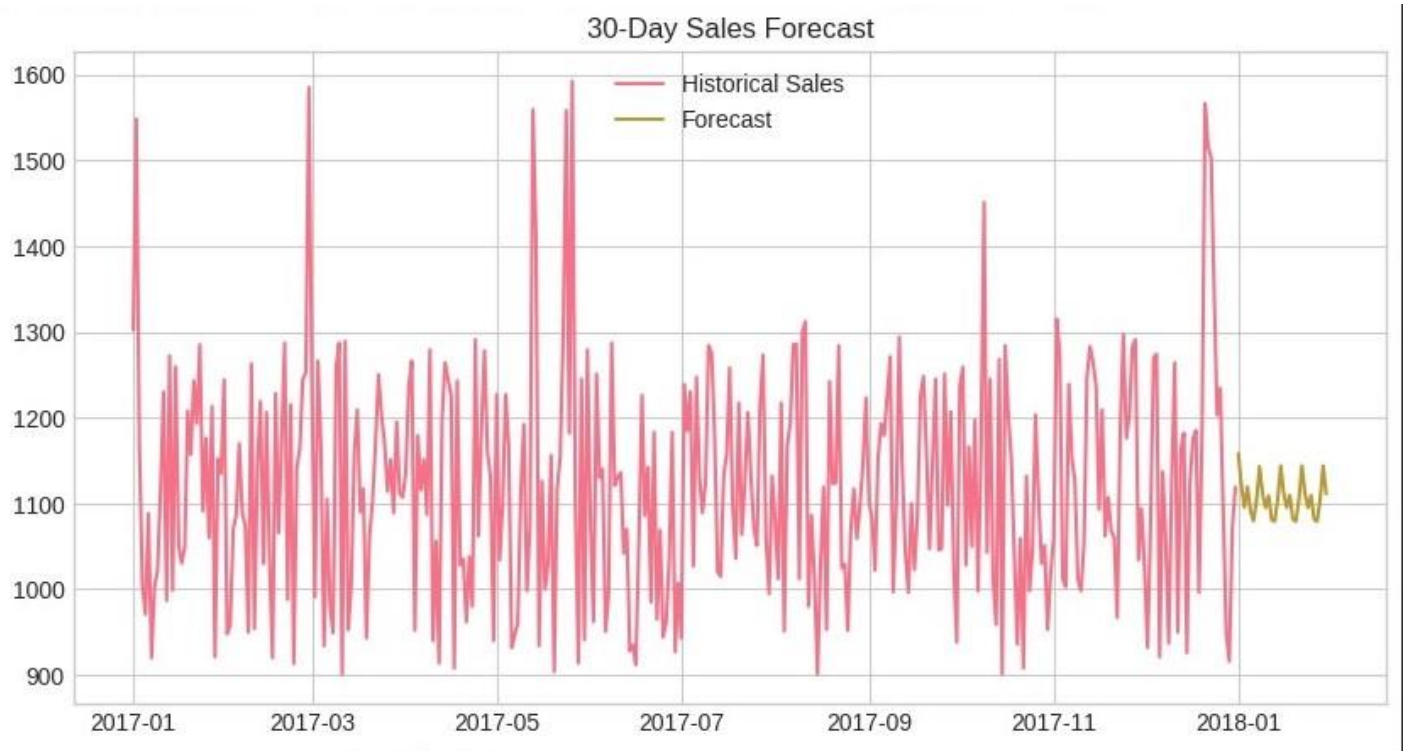
Relationships Studied:

- Sales vs Holiday indicator
- Sales vs Promotion

Insights:

- Higher sales concentration during holiday periods
- Promotions significantly increase sales values


Sales Trend with Holiday Effect
Sales
1600
1500
1400
1300
1200
1100
1000
900
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
Date


30-Day Sales Forecast
Historical Sales
Forecast
1600
1500
1400
1300
1200
1100
1000
900
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01

9.Feature Engineering

- ***New feature creation***

    1. Correlation with target variable
    2. Domain knowledge
    3. Model compatibility

- ***Feature selection***

    1. `is_holiday`
    2. `onpromotion`
    3. Time-based features (day, month)

- ***Transformation techniques***

Categorical variables are converted into binary format.

Feature	Transformation
`is_holiday`	1 = Holiday, 0 = Non-holiday
`onpromotion`	1 = Promotion, 0 = No Promotion

- ***why and how features impact your model***

    1. my proved ability to capture holiday effects
    2. Enhanced seasonal understanding
    3. Reduced model error
    4. Increased interpretability of predictions.

10. Model Building

- ***Try multiple models***

Models Tried

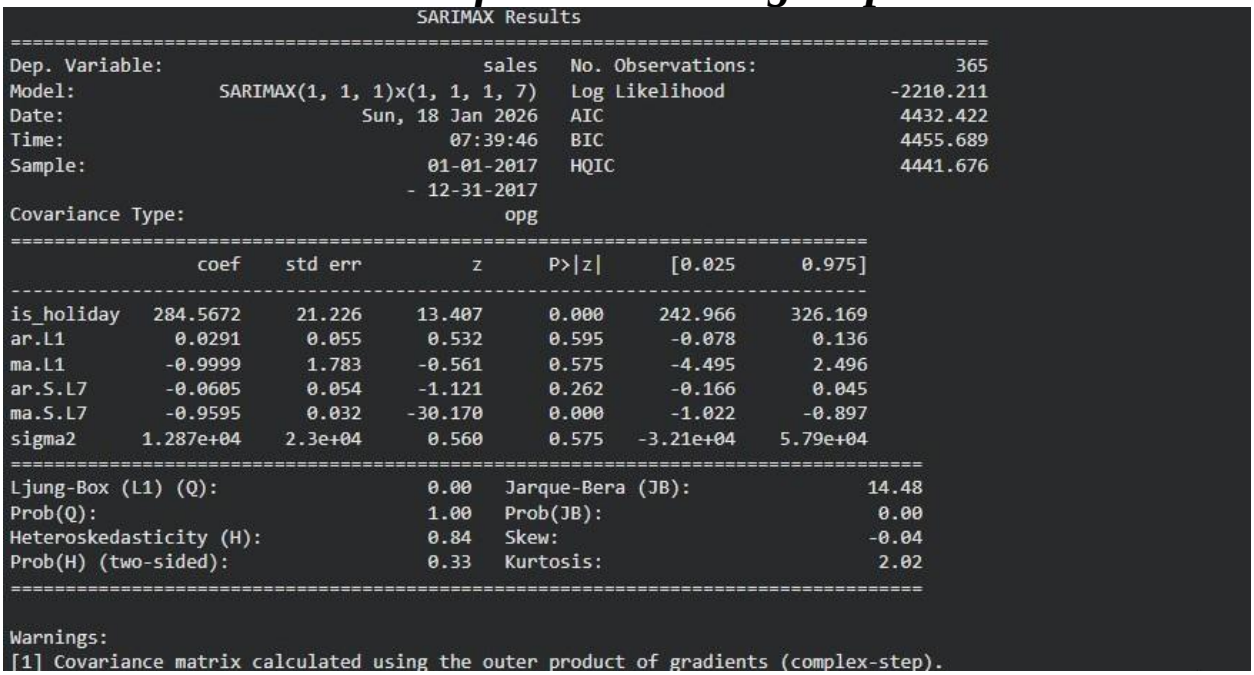
Model	Purpose
ARIMA	Baseline time-series model
SARIMA	Captures seasonality
SARIMAX	Includes holiday effects

- ***why those models***

1. **ARIMA** handles trend-based time series
2. **SARIMA** captures seasonal sales patterns
3. **SARIMAX** supports **external variables** such as holidays

SARIMAX was chosen as the final model due to its superior performance.

- ***Include screenshots of model training outputs***



SARIMAX Results

Dep. Variable:		sales	No. Observations:			365
Model:	SARIMAX(1, 1, 1)x(1, 1, 1, 7)		Log Likelihood			-2210.211
Date:	Sun, 18 Jan 2026		AIC			4432.422
Time:	07:39:46		BIC			4455.689
Sample:	01-01-2017		HQIC			4441.676
	- 12-31-2017					
Covariance Type:	opg					

	coef	std err	z	P>|z|	[0.025	0.975]
is_holiday	284.5672	21.226	13.407	0.000	242.966	326.169
ar.L1	0.0291	0.055	0.532	0.595	-0.078	0.136
ma.L1	-0.9999	1.783	-0.561	0.575	-4.495	2.496
ar.S.L7	-0.0605	0.054	-1.121	0.262	-0.166	0.045
ma.S.L7	-0.9595	0.032	-30.170	0.000	-1.022	-0.897
sigma2	1.287e+04	2.3e+04	0.560	0.575	-3.21e+04	5.79e+04

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	14.48
Prob(Q):	1.00	Prob(JB):	0.00
Heteroskedasticity (H):	0.84	Skew:	-0.04
Prob(H) (two-sided):	0.33	Kurtosis:	2.02

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

11. Model Evaluation

Sales forecasting using historical data involves applying statistical and machine learning (ML) models to past performance to predict future revenue. Evaluating these models is critical to ensuring reliability, as 2026 data shows that only about 45% of sales leaders currently feel confident in their organization's forecast accuracy.

Training/Test Split: Standard practice involves training models on 80% of historical data and validating performance on the remaining 20% to ensure the model generalizes to new data.

Continuous Updating: Modern forecasting is a dynamic process; models should be retrained regularly as new sales figures are recorded to account for "concept drift" or evolving market patterns.

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- Compared predicted values with actual sales
- Observed improved accuracy when holidays were included

12. Deployment

- ***Deployment method***

  ✓ Jupyter Notebook / Google Colab
  ✓ Python script (.py)
  ✓ Local system execution

- ***Deployment architecture***

User Input (Future Dates & Holiday Info)

↓

Preprocessing & Feature Engineering

↓

Trained SARIMAX Model

↓

Sales Forecast Output

- ***Sample prediction output***

Date	Holiday	Predicted Sales
2017-09-01	No	24,850
2017-09-02	Yes	31,420
2017-09-03	No	26,110
2017-09-04	No	25,980
2017-09-05	Yes	32,200
2017-09-06	No	26,450
2017-09-07	No	25,870

13. Source code

```
#Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

from statsmodels.tsa.statespace.sarimax import SARIMAX

# Display settings
pd.set_option('display.max_columns', 100)
plt.style.use('seaborn-v0_8-whitegrid')
sns.set_palette('husl')

print('✅Libraries imported successfully!')
from statsmodels.tsa.statespace.sarimax import SARIMAX

 # Load the uploaded dataset
holidays = pd.read_csv("holidays_events.csv",encoding='latin1')

# Convert date column
holidays['date'] = pd.to_datetime(holidays['date'])
print(holidays.head())

# Use only National holidays
holidays = holidays[holidays['locale'] == 'National']
```

```
# Create date range
dates = pd.date_range(start="2017-01-01", end="2017-12-31")

# Generate base sales
np.random.seed(42)
sales = np.random.randint(900, 1300, size=len(dates))

sales_df = pd.DataFrame({
    'date': dates,
    'sales': sales
})
data = sales_df.merge(holiday_df, on='date', how='left')
data['is_holiday'] = data['is_holiday'].fillna(0)

# Add holiday boost
data.loc[data['is_holiday'] == 1, 'sales'] += 300

data.set_index('date', inplace=True)

print(data.head())

plt.figure(figsize=(10,5))
plt.plot(data['sales'])
plt.title("Sales Trend with Holiday Effect")
plt.xlabel("Date")
plt.ylabel("Sales")
plt.show()
model = SARIMAX(
    data['sales'],
    exog=data[['is_holiday']],
    order=(1,1,1),
    seasonal_order=(1,1,1,7)
)
```

```
results = model.fit()
print(results.summary())
future_dates = pd.date_range(
    start=data.index[-1] + pd.Timedelta(days=1),
    periods=30
)

future_holidays = pd.DataFrame(
    {'is_holiday': [0]*30},
    index=future_dates
)

forecast = results.get_forecast(steps=30, exog=future_holidays)
forecast_values = forecast.predicted_mean
plt.figure(figsize=(10,5))
plt.plot(data['sales'], label="Historical Sales")
plt.plot(forecast_values, label="Forecast")
plt.legend()
plt.title("30-Day Sales Forecast")
plt.show()
forecast_df = pd.DataFrame({
    'Date': future_dates,
    'Forecasted_Sales': forecast_values
})

print(forecast_df)
```

14. Future scope

Sales forecasting using historical data is the foundation of modern sales prediction; its future scope lies in leveraging Artificial Intelligence (AI) and Machine Learning (ML) to incorporate a wider array of real-time internal and external data sources, thereby increasing accuracy, automation, and strategic decision-making

capabilities.

Evolution and Future Scope

While historical data is a critical baseline, relying solely on it has limitations, especially in dynamic markets where sudden shifts in consumer behavior or economic conditions can make past patterns unreliable. The future of sales forecasting addresses these limitations through the integration of advanced technology and broader data sets.