PROJECT REPORT

PROJECT TITLE:FUTURE SALES PREDICTION

1.Project Overview:

Title: Future sales prediction

Introduction:

Sales forecasting is the process of estimating future revenue by predicting how much of a product or service will sell in the next week, month, quarter, or year. At its simplest, a sales forecast is a projected measure of how a market will respond to a company's go-to-market efforts. explain the significance of the project. Significance: Forecasts are about the future. It's hard to overstate how important it is for a company to produce an accurate sales forecast. Privately held companies gain confidence in their business when leaders can trust forecasts. For publicly traded companies, accurate forecasts confer credibility in the market. Sales forecasting adds value across an organization. Finance relies on forecasts to develop budgets for capacity plans and hiring, and production uses sales forecasts to plan their cycles. Forecasts help sales operations with territory and quota planning, supply chain with material purchases and production capacity, and sales strategy with channel and partner strategies.

2.Data Collection:

The input source for this project is a sales dataset ,which is downloaded from kaggle datasets,which contains 200 rows * 5 columns

DATA PREPROCESSING

```
df =pd.read_csv("/content/drive/MyDrive/EDA dataset/Sales (1).csv")
df
```

	Date	TV	Radio	Newspaper	Sales
0	12-04-2023	230.1	37.8	69.2	22.1
1	13-04-2023	44.5	39.3	45.1	10.4
2	14-04-2023	17.2	45.9	69.3	12.0
3	15-04-2023	151.5	41.3	58.5	16.5
4	16-04-2023	180.8	10.8	58.4	17.9
...	...	...	...	...	...
195	24-10-2023	38.2	3.7	13.8	7.6
196	25-10-2023	94.2	4.9	8.1	14.0
197	26-10-2023	177.0	9.3	6.4	14.8
198	27-10-2023	283.6	42.0	66.2	25.5
199	28-10-2023	232.1	8.6	8.7	18.4

200 rows × 5 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Date       200 non-null    object 
 1   TV         200 non-null    float64
 2   Radio      200 non-null    float64
 3   Newspaper  200 non-null    float64
 4   Sales      200 non-null    float64
dtypes: float64(4), object(1)
memory usage: 7.9+ KB
```

```
duplicates = df.duplicated()

# Print the duplicate rows
print(df[duplicates])
```

```
Empty DataFrame
Columns: [Date, TV, Radio, Newspaper, Sales]
Index: []
```

```
# Drop duplicate rows
data = df.drop_duplicates()

# Print the updated DataFrame
print(data)
```

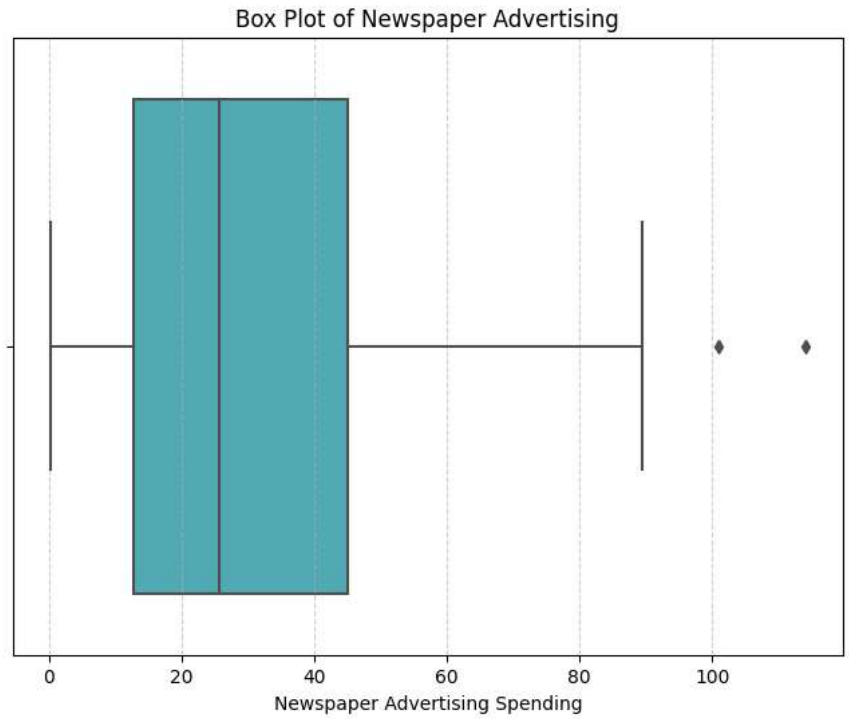
```
            Date     TV  Radio  Newspaper  Sales
0    12-04-2023  230.1   37.8       69.2   22.1
1    13-04-2023   44.5   39.3       45.1   10.4
2    14-04-2023   17.2   45.9       69.3   12.0
3    15-04-2023  151.5   41.3       58.5   16.5
4    16-04-2023  180.8   10.8       58.4   17.9
..          ...    ...    ...        ...    ...
195  24-10-2023   38.2    3.7       13.8    7.6
196  25-10-2023   94.2    4.9        8.1   14.0
197  26-10-2023  177.0    9.3        6.4   14.8
198  27-10-2023  283.6   42.0       66.2   25.5
199  28-10-2023  232.1    8.6        8.7   18.4

[200 rows x 5 columns]
```

```
#checking for missing values
df.isna().head()
```

	Date	TV	Radio	Newspaper	Sales
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False

Detecting outliers using Box plot

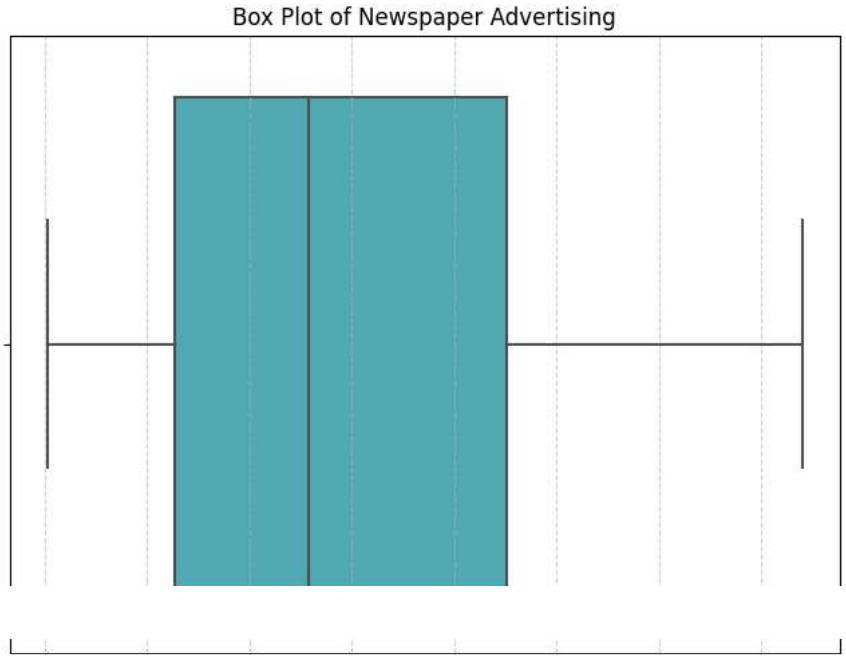
```
# Create the box plot
plt.figure(figsize=(8, 6))
sns.boxplot(x='Newspaper', data=df, palette='YlGnBu')
plt.title('Box Plot of Newspaper Advertising')
plt.xlabel('Newspaper Advertising Spending')
plt.grid(axis='x', linestyle='--', alpha=0.6)

# Show the plot
plt.show()
```



Removed Outliers

```
import numpy as np

# Ambang batas atas (threshold) untuk Winsorizing
upper_threshold = 2 * np.std(df['Newspaper']) + np.mean(df['Newspaper'])

# Menerapkan Winsorizing pada kolom 'Newspaper'
df['Newspaper'] = np.where(df['Newspaper'] > upper_threshold, upper_threshold, df['Newspaper'])
```

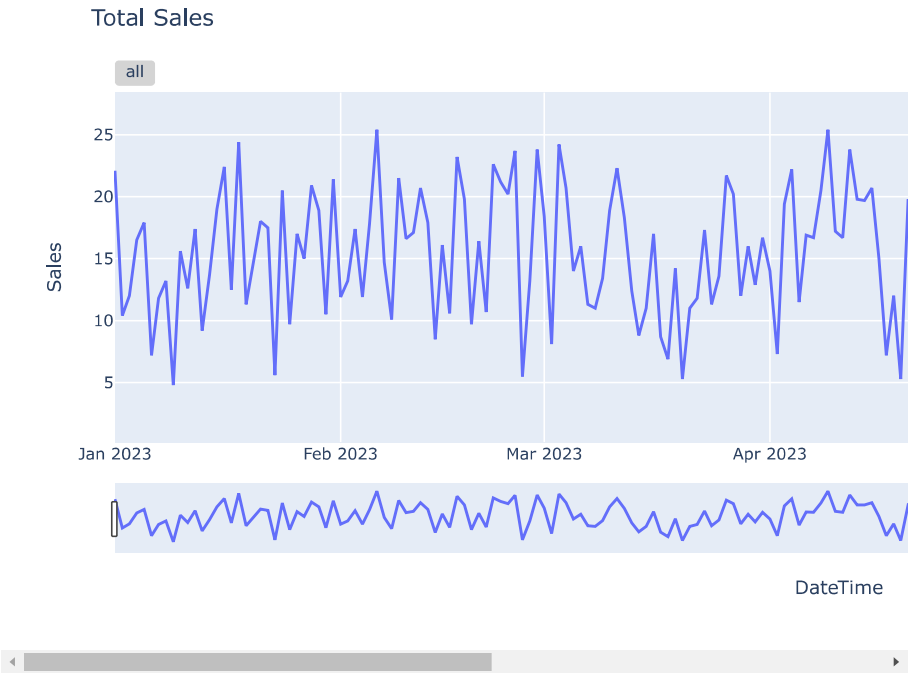
```
# Create the box plot
plt.figure(figsize=(8, 6))
sns.boxplot(x='Newspaper', data=df, palette='YlGnBu')
plt.title('Box Plot of Newspaper Advertising')
plt.xlabel('Newspaper Advertising Spending')
plt.grid(axis='x', linestyle='--', alpha=0.6)

# Show the plot
plt.show()
```



3.Data Exploration:

summary statistics of the data

```
fig = px.line(df, x='DateTime', y='Sales', title='Total Sales')

fig.update_xaxes(
    rangeslider_visible=True,
    rangeselector=dict(
        buttons=list([
            dict(step="all")
        ])
    )
)
fig.show()
```

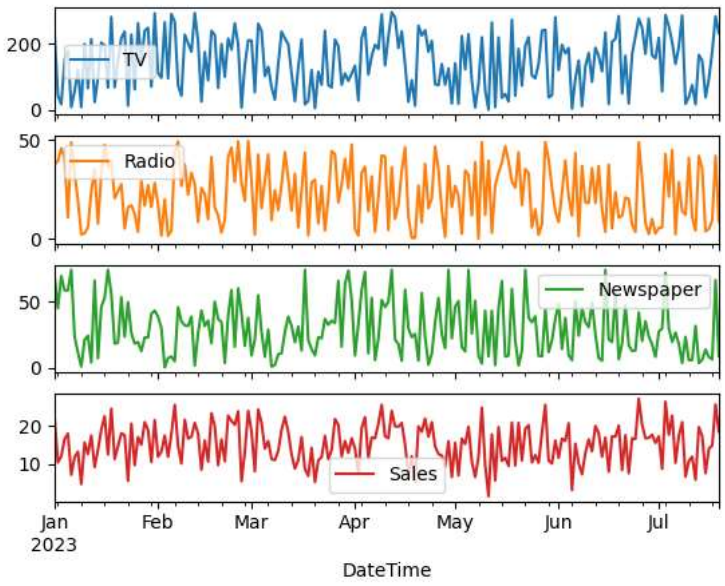


Visualizations of data distributions and patterns.

```
el_df=df.set_index('DateTime')
```

```
el_df.plot(subplots=True)
```

```
array([<Axes: xlabel='DateTime'>, <Axes: xlabel='DateTime'>,
       <Axes: xlabel='DateTime'>, <Axes: xlabel='DateTime'>], dtype=object)
```



Identifying any data cleaning or missing value handling.

```
print ("\nMissing values :  ", df.isnull().any())
```

```
Missing values :   Date         False
TV           False
Radio        False
Newspaper    False
Sales        False
DateTime     False
dtype: bool
```

4.Feature Engineering:

Features selected for prediction:

- used Time series forecasting model(Arima model)
- training model and testing model
- Resampled the data before training and testing
- forecasted the result using model.predict()

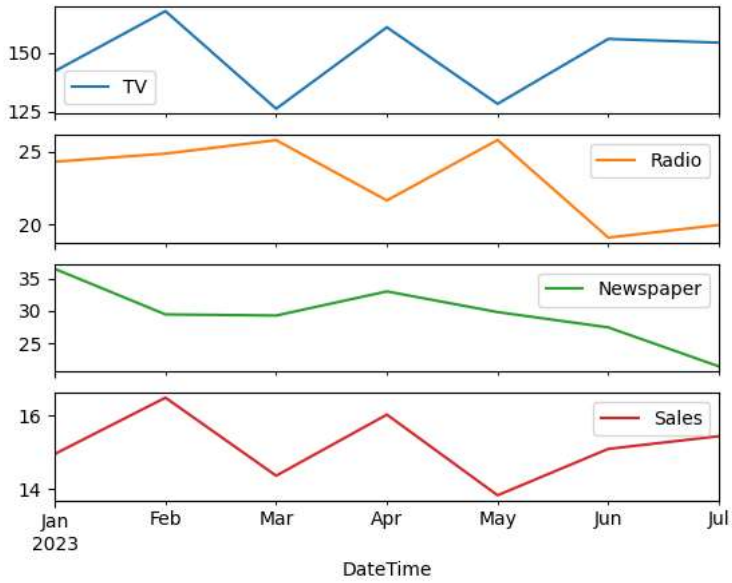
Transformations:

```
el_df.resample('M').mean()
```

```
<ipython-input-115-421011436e0d>:1: FutureWarning:

The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future ve

                      TV      Radio  Newspaper      Sales
```

```
el_df.resample('M').mean().plot(subplots=True)
```

```
<ipython-input-116-052b9850bc35>:1: FutureWarning:

The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future ve

array([<Axes: xlabel='DateTime'>, <Axes: xlabel='DateTime'>,
       <Axes: xlabel='DateTime'>, <Axes: xlabel='DateTime'>], dtype=object)
```




```
final_df=el_df.resample('M').mean()
final_df
```

```
<ipython-input-117-262a0f12b9cd>:1: FutureWarning:

The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future ve
```

	TV	Radio	Newspaper	Sales
DateTime				
2023-01-31	142.064516	24.319355	36.519498	14.954839
2023-02-28	167.635714	24.878571	29.460714	16.478571
2023-03-31	126.100000	25.803226	29.283942	14.367742
2023-04-30	160.776667	21.656667	33.016814	16.020000
2023-05-31	128.183871	25.816129	29.819498	13.838710
2023-06-30	155.813333	19.100000	27.446740	15.093333
2023-07-31	154.221053	19.968421	21.400000	15.436842

5.Model Selection:

Arima model Autoregressive integrated moving average(ARIMA) models predict future values based on past values.ARIMA makes use of lagged moving averages to smooth time series data it captures the patterns ,trends and seasonality of the data using a combination of past values,differences and errors.

- install pmdarima
- import pmdarima

The parameters can be defined as:

• p: the number of lag observations in the model, also known as the lag order.

• d: the number of times the raw observations are differenced; also known as the degree of differencing.

• q: the size of the moving average window, also known as the order of the moving average.

```
!pip install pmdarima
```

```
Requirement already satisfied: pmdarima in /usr/local/lib/python3.10/dist-packages (2.0.4)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (1.3.2)
Requirement already satisfied: Cython!=0.29.18,!=0.29.31,>=0.29 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (3.0.4)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (1.23.5)
Requirement already satisfied: pandas>=0.19 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (1.5.3)
Requirement already satisfied: scikit-learn>=0.22 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (1.2.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (1.11.3)
Requirement already satisfied: statsmodels>=0.13.2 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (0.14.0)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (2.0.7)
Requirement already satisfied: setuptools!=50.0.0,>=38.6.0 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (67.7.2)
Requirement already satisfied: packaging>=17.1 in /usr/local/lib/python3.10/dist-packages (from pmdarima) (23.2)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.19->pmdarima) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.19->pmdarima) (2023.3.post1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.22->pmdarima) (3.2.
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.13.2->pmdarima) (0.5.3)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from patsy>=0.5.2->statsmodels>=0.13.2->pmdarima) (1.16.0
```

```
import pmdarima as pm
```

```
model = pm.auto_arima(final_df['Sales'],
                      m=12, seasonal=False,
                     start_p=0, start_q=0, max_order=4, test='adf',error_action='ignore',
                         suppress_warnings=True,
                     stepwise=True, trace=True)
```

```
/usr/local/lib/python3.10/dist-packages/pmdarima/arima/_validation.py:62: UserWarning:

m (12) set for non-seasonal fit. Setting to 0

Performing stepwise search to minimize aic
 ARIMA(0,0,0)(0,0,0)[0]             : AIC=59.957, Time=0.02 sec
 ARIMA(1,0,0)(0,0,0)[0]             : AIC=inf, Time=0.04 sec
 ARIMA(0,0,1)(0,0,0)[0]             : AIC=inf, Time=0.05 sec
 ARIMA(1,0,1)(0,0,0)[0]             : AIC=inf, Time=0.28 sec
 ARIMA(0,0,0)(0,0,0)[0] intercept   : AIC=21.474, Time=0.06 sec
 ARIMA(1,0,0)(0,0,0)[0] intercept   : AIC=20.463, Time=0.10 sec
 ARIMA(2,0,0)(0,0,0)[0] intercept   : AIC=22.433, Time=0.30 sec
 ARIMA(1,0,1)(0,0,0)[0] intercept   : AIC=inf, Time=0.37 sec
 ARIMA(0,0,1)(0,0,0)[0] intercept   : AIC=inf, Time=0.26 sec
 ARIMA(2,0,1)(0,0,0)[0] intercept   : AIC=34.692, Time=0.60 sec

Best model:  ARIMA(1,0,0)(0,0,0)[0] intercept
Total fit time: 2.115 seconds
```

we can then select the model that has the lowest AIC, BIC, and RMSE, and the best forecast and residual plots. In this case, it may be an ARIMA(0,0,0) model with a seasonal component.

6.Model Training:

Splitting data into “training” and “test” sets

```
train=final_df[(final_df.index.get_level_values(0) >= '2023-01-31') & (final_df.index.get_level_values(0) <= '2023-05-31')]
```

```
test=final_df[(final_df.index.get_level_values(0) > '2023-05-31')]
```

```
test=final_df[(final_df.index.get_level_values(0) > '2023-05-31')]
```

```
test
```

	TV	Radio	Newspaper	Sales

7.Model Evaluation:

Fitting ARIMA models

```
model.fit(train['Sales'])
```

```
ARIMA(1,0,0)(0,0,0)[0] intercept
```

```
forecast=model.predict(n_periods=4, return_conf_int=True)
```

```
forecast
```

```
(2023-06-30    16.519554
 2023-07-31    14.269223
 2023-08-31    16.158176
 2023-09-30    14.572567
 Freq: M, dtype: float64,
 array([[15.42250821, 17.61659955],
        [12.83691124, 15.70153389],
        [14.53059205, 17.78576082],
        [12.82041195, 16.32472192]]))
```

```
forecast_df = pd.DataFrame(forecast[0],index = test.index,columns=['Prediction'])
```

```
forecast_df
```

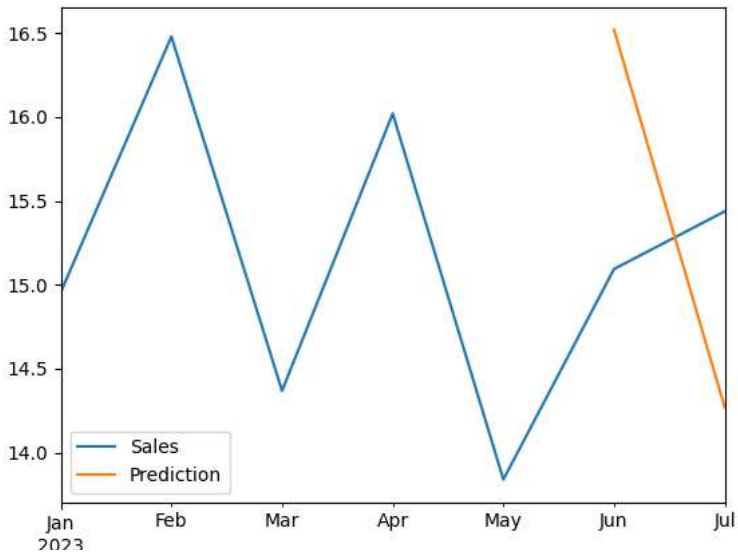
DateTime	Prediction
2023-06-30	16.519554
2023-07-31	14.269223

8.Future Sales Prediction:

```
import matplotlib.pyplot as plt
```

```
pd.concat([final_df['Sales'],forecast_df],axis=1).plot()
```

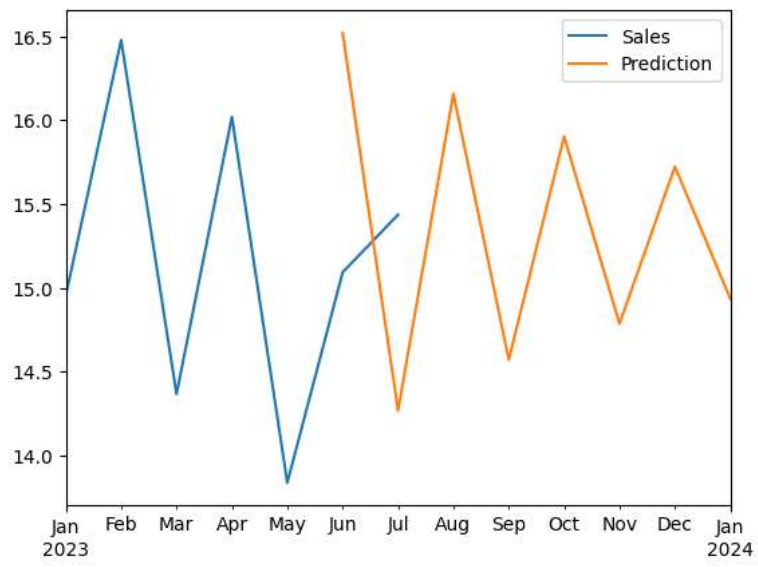
```
<Axes: xlabel='DateTime'>
```




```
forecast1=model.predict(n_periods=8, return_conf_int=True)
forecast_range=pd.date_range(start='2023-06-30', periods=8,freq='M')


forecast1_df = pd.DataFrame(forecast1[0],index =forecast_range,columns=['Prediction'])
```

```
pd.concat([final_df['Sales'],forecast1_df],axis=1).plot()
```

```
<Axes: >
```



9.Conclusion:

The main objective of the system is to analyze the future sales of a particular company and to predict whether a particular sales will increase or decrease by using different Arima models.

Sales forecasting allows companies to efficiently allocate resources for future growth and manage its cash flow. Sales forecasting also helps businesses to estimate their costs and revenue accurately based on which they are able to predict their short-term and long-term performance.

The future sales prediction project report concludes with several key findings and recommendations that can guide decision-makers in their efforts to improve sales forecasting and boost overall business performance.

Key Findings:

1. Accurate sales predictions are crucial for effective inventory management, resource allocation, and revenue optimization.
2. Historical data and advanced data analysis techniques, such as time series forecasting and machine learning models, play a significant role in enhancing sales predictions.
3. Data quality and completeness are essential for reliable predictions, highlighting the importance of data cleansing and preprocessing.
4. The choice of forecasting methods depends on the specific nature of the business, product line, and market dynamics.

Recommendations:

1. Invest in data quality: Regularly clean and update your sales data to ensure it's accurate and complete. Implement data quality checks and validation processes to minimize errors.
2. Embrace advanced analytics: Utilize time series forecasting methods, machine learning models, and predictive analytics tools to gain deeper insights into sales trends and drivers.
3. Leverage external data: Incorporate external data sources, such as economic indicators and industry trends, to enhance the accuracy of predictions and provide a more holistic view of the market.
4. Collaboration and domain expertise: Foster collaboration between data scientists, analysts, and domain experts to improve model accuracy by incorporating domain-specific knowledge.
5. Continuous improvement: Sales predictions should not be a one-time effort. Regularly reassess and recalibrate models to adapt to changing market conditions.

In conclusion, accurate sales predictions are critical for any business looking to optimize its operations and financial performance. By investing in data quality, advanced analytics, external data sources, collaboration, and ongoing improvement, organizations can increase their sales forecasting accuracy and make more informed decisions. This project report provides a foundation for improving sales predictions and should be viewed as a continuous effort to enhance business competitiveness and efficiency.

10.References:

Input dataset from kaggle

Libraries used are:

```
#import packages
import pandas as pd
import numpy as np
import matplotlib.colors as col
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import datetime
from pathlib import Path
import random
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from xgboost.sklearn import XGBRegressor
from sklearn.model_selection import KFold, cross_val_score, train_test_split
```

11.Summary

Summary of Future Sales Prediction Project

The Future Sales Prediction Project was undertaken to improve sales forecasting accuracy and facilitate more informed decision-making within the organization. Accurate sales predictions are essential for optimizing inventory management, resource allocation, and ultimately increasing revenue. This summary provides an overview of the project's objectives, methodologies, key findings, and recommendations.

Objectives:

1. Enhance sales forecasting accuracy.
2. Improve resource allocation and inventory management.
3. Drive informed decision-making and business growth.

Methodologies: The project utilized a combination of data-driven methodologies, including:

1. **Data Collection and Preparation:** Gathering historical sales data from multiple sources, ensuring data quality through cleansing and preprocessing.
2. **Time Series Forecasting:** Employing time series analysis techniques to identify trends, seasonality, and patterns in sales data.
3. **Machine Learning Models:** Developing predictive models using machine learning algorithms to forecast future sales.
4. **External Data Integration:** Incorporating external data sources, such as economic indicators and industry trends, to enrich the analysis.
5. **Domain Expertise:** Collaborating with subject matter experts to fine-tune models and incorporate industry-specific knowledge.

Key Findings: The project yielded several significant findings:

1. Identification of sales trends and patterns, helping to better understand historical sales behavior.
2. Evaluation of model performance, indicating the strengths and weaknesses of various predictive algorithms.
3. Identification of key drivers influencing sales outcomes, such as seasonality, marketing campaigns, and economic factors.

Recommendations: Based on the findings, the following recommendations have been put forward:

1. **Data Quality Improvement:** Invest in regular data cleansing and validation processes to ensure the accuracy and completeness of sales data.
2. **Advanced Analytics Adoption:** Embrace time series forecasting and machine learning models to enhance sales predictions.
3. **External Data Integration:** Incorporate external data sources to provide a more comprehensive view of the market.
4. **Collaboration and Knowledge Sharing:** Foster collaboration between data scientists, analysts, and domain experts to improve model accuracy.
5. **Continuous Improvement:** Treat sales predictions as an ongoing effort, regularly recalibrating models to adapt to changing market conditions.

In conclusion, the Future Sales Prediction Project highlights the critical role that accurate sales predictions play in modern business operations. By implementing the recommendations outlined in this project, the organization can enhance its sales forecasting accuracy, make more informed decisions, and maintain a competitive edge in the market. This project serves as a foundation for continuous improvement and adaptation to evolving market dynamics.