**PROJECT**: **Future**  **sales prediction**:

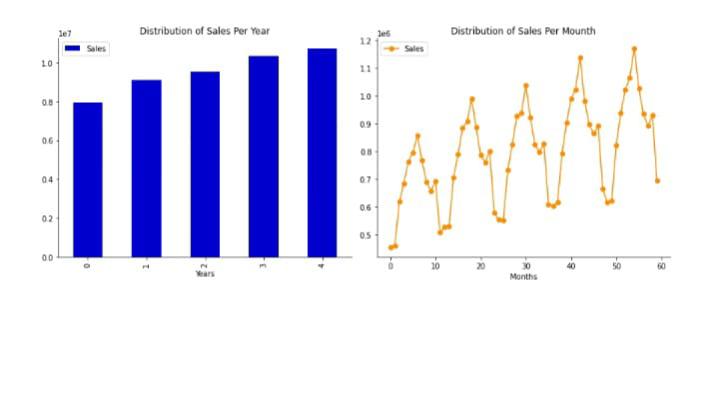
Phase-2

Synopsis:

* + - Explanation
    - Dataset and it’s details
    - Columns detail
    - Details about libraries download and it’s way of usages
    - Train and Test
    - Other explanation
    - Accuracy check

FUTURE SALES PREDICTION EXPLANATION:

Predicting the future sales of a product helps a business manage the manufacturing and advertising cost

of the product. There are many more benefits of predicting the future sales of a product.

In this ,

article, we will take you through the task of future sales prediction with machine learning using Python.

Future Sales Prediction example

The dataset given here contains the data about the sales of the product. The dataset is about the

advertising cost incurred by the business on various advertising platforms. Below is the description of all

the columns in the dataset:

TV: Advertising cost spent in dollars for advertising on TV;

Radio: Advertising cost spent in dollars for advertising on Radio;

Newspaper: Advertising cost spent in dollars for advertising on Newspaper;

Sales: Number of units sold;

So, in the above dataset, the sales of the product depend on the advertisement cost of the product. I

hope you now have understood everything about this dataset. Now in the section below, I will take you

through the task of future sales prediction with machine learning using Python.

Future Sales Prediction using Python

Let’s start the task of future sales prediction with machine learning by importing the necessary Python

libraries and the dataset:

Import pandas as pd

Import numpy as np

From sklearn.model\_selection import train\_test\_split

From sklearn.linear\_model import LLinearRegression

Data\_pd.read\_csv(https://raw.githubusercontent.com/amankharwal/Website-

data/master/advertising.csv)

Print(data.head())

TV Radio Newspaper Sales

0 230.1 37.8 69.2 22.1

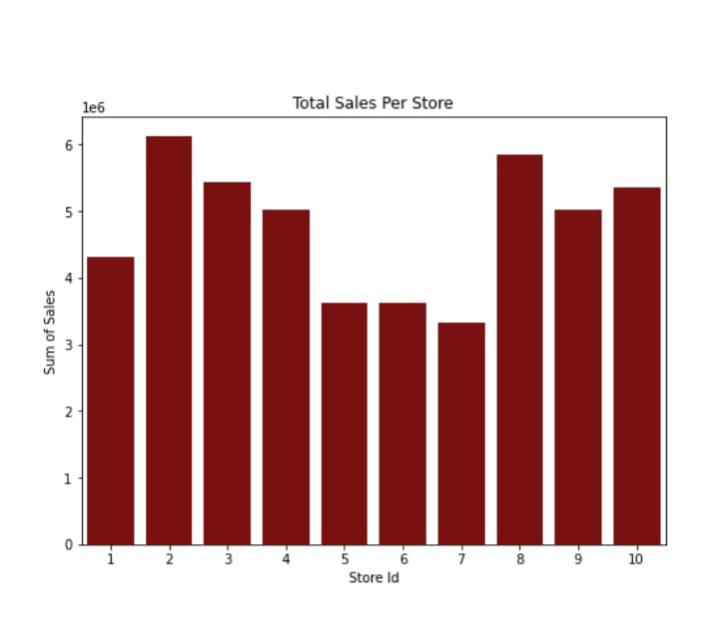
1 44.5. 39.3. 45.1 10.4

2 17.2. 45.9 69.3 12.0

3 151.5 41.3 58.5 16.5

4 180.8. 10.8 58.4 17.9

DATASET AND IT’S DETAILS:

 First step is to load the data and transform it into a structure that we will then use for each of our models. In its raw form, each row of data represents a single day of sales at one of ten stores. Our goal is to predict monthly sales, so we will first consolidate all stores and days into total monthly sales.

Dataset = pd.read\_csv('/input/demand\_forecasting\_kernals\_only/sample\_submission.csv’)

Df= dataset.copy()

Df.head()

Id sales

0 52

1 52

2 52

3 52

4 52

Def load\_data(file\_name):

"""Returns a pandas dataframe from a csv file.""""

Return pd.read\_csv(file\_name)

Sales\_data = load\_data('../input/demand-forecasting-kernels-only/train.csv')

Df\_s.sales\_data.copy(

Df\_s.info()

<class 'pandas.core.frame.DataFrame">

RangeIndex: 913000 entries, 0 to 912999

Data columns (total 4 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

date 913000 non-null object

store 913000 non-null int64

item 913000 non-null int64

sales 913000 non-null int64

Dtypes: int64(3), object(1)

Memory usage: 27.9+ MB

Df\_s.tail()

Date store item sales

912995 2017-12-27 10 50 63

912996 2017-12-28 10. 50. 59 912997 2017-12-29 10 50 74

912998 2017-12-30 10 50 62

912999 2017-12-31 10 50 82

# To view basic statistical details about dataset:

Df\_s[‘sales’].describe()

Count 913000.000000

Mean 52.250287

Std 28.801144

Min 0.000000

25% 30.000000

50% 47.000000

75% 70.000000

Max 231.000000

Name: sales, dtype: float64

Sales seem to be unbalanced!

Df\_s[‘sales’].plot()

COLUMN DETAILS:

Column explanation in the dataset:

The dataset we mentioned appears to be related to predicting sales based on different advertising mediums, such as television, radio, and newspapers. Here’s a brief explanation of the columns typically found in such a dataset:

1. Television: This column likely contains data on the amount of money spent on advertising via television for a specific period, such as a week or month.

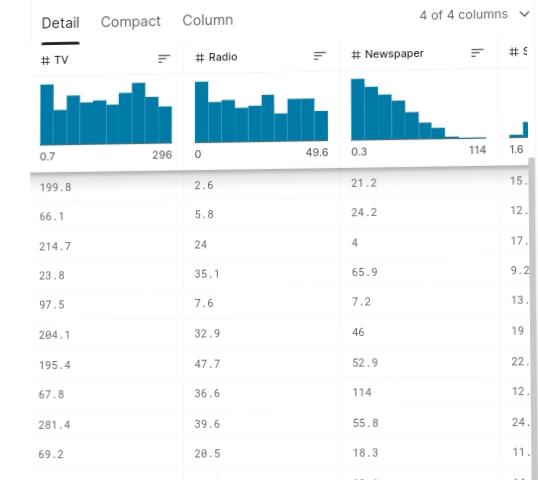
2. Radio: Similarly, this column contains data on advertising expenses for radio advertising.

3. Newspaper: This column contains data on advertising expenses for newspaper advertisements.

4. Sales: This column usually represents the sales figures for a product or service during the same period for which advertising expenses are recorded.

To predict future sales based on these advertising mediums, you can use regression analysis or machine learning techniques.

By analyzing historical data , we can build a predictive model that takes advertising expenses as input features (Television, Radio, Newspaper) and predicts future sales, helping businesses make informed decisions about their advertising budgets.



LIBRARIES:

To work on future sales prediction in data science, you will typically need to use libraries in Python, as Python is a popular language for data science and machine learning. Here are some essential libraries and steps for future sales prediction:

Python: Make sure you have Python installed on your system. You can download it from the official Python website (https://www.python.org/).

1. NumPy: NumPy is used for numerical operations in Python. You can install it using pip:

Pip install numpy

1. Pandas: Pandas is used for data manipulation and analysis. Install it with pip:

Pip install pandas

1. Matplotlib and seaborn: These libraries are essential for data visualization:

Pip install matplotlib seaborn

1. Scikit\_Learn: Scikit\_learnis a powerful library for machine learning. You can use it for building predictive models:

Pip install scikit-learn

1. Tensorflow Or Torch: If you plan to work with deep learning models, you’ll need one of these Libraries:

Pip install Tensorflow

# OR

Pip install torch

6.statsmodels: Statsmodels is useful for statistical modeling and

Hypothesis testing:

Pip install statsmodels

7.prophet:If you want to use a library specifically designed for

Data science and model development install it:

Pip install fbprophet

8.Jupyter notebook: While not a library, Jupyter Notebook is

an excellent interactive environment for data science and

model development. Install it with:

Pip install jupyter

9.Others data processing libraries:Depending on your data, you might need other libraries like scikit-learn’s preprocessing module (`sklearn.preprocessing`) for data preprocessing tasks.

10.Database Libraries: your data is stored in a database, consider using libraries like SQLAlchemy or PyMySQL to connect to and fetch data from the database.

Remember to keep your libraries and dependencies up-to-date. n do this by periodically running.

Pip Install-upgrade library-name. Once you have these libraries installed, you can start working on your future sales prediction project using the various data preprocessing, modeling, and evaluation techniques provided by these libraries.

TRAIN AND TEST:

Certainly! In machine learning and data analysis:

1. Training Data: This is the portion of the dataset used to train a machine learning model. The model learns patterns, relationships, and features from this data. It's like a teacher showing examples to a student.

2. Testing Data: After training, the model is tested on a separate dataset called the testing data. This data is not used during training and serves as a benchmark to evaluate how well the model generalizes to new, unseen data. It's like giving a test to the student to see how well they've learned.

The goal is to build a model that performs well on the testing data, which indicates that it can make accurate predictions on new, real-world data. This process helps assess the model's performance and prevents overfitting (where the model memorizes the training data but doesn't generalize well).

Example :

Import pandas as pd

Import numpy as np

Import matplotlib.pyplot as plt

From statsmodels.tsa.arima\_model import ARIMA

From sklearn.metrics import mean\_squared\_error

# Load your historical sales data into a DataFrame

Sales\_data = pd.read\_csv(‘sales\_data.csv’) # Replace ‘sales\_data.csv’ with your dataset

# Convert the date column to a datetime object and set it as the index

Sales\_data[‘Date’] = pd.to\_datetime(sales\_data[‘Date’])

Sales\_data.set\_index(‘Date’, inplace=True)

# Split the data into training and testing sets

Train\_size = int(len(sales\_data) \* 0.8)

Train\_data, test\_data = sales\_data[:train\_size], sales\_data[train\_size:]

# Fit an ARIMA model to the training data

Model = ARIMA(train\_data, order=(5,1,0)) # You can adjust the order parameters

Model\_fit = model.fit(disp=0)

# Make predictions on the test set

Predictions = model\_fit.forecast(steps=len(test\_data))[0]

# Calculate Mean Squared Error to evaluate the model

Mse = mean\_squared\_error(test\_data, predictions)

Rmse = np.sqrt(mse)

Print(f’Root Mean Squared Error (RMSE): {rmse}’)

# Plot the actual vs. predicted sales

Plt.plot(test\_data.index, test\_data, label=’Actual Sales’)

Plt.plot(test\_data.index, predictions, color=’red’, label=’Predicted Sales’)

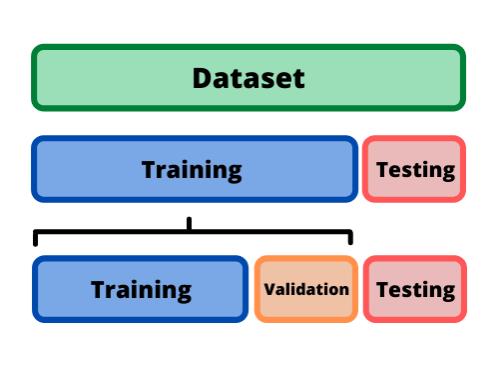
Plt.legend()

Plt.xlabel(‘Date’)

Plt.ylabel(‘Sales’)

Plt.title(‘Sales Forecasting with ARIMA’)

Plt.show()



SOME OF OTHER TOPICS:

Forecasting:

Forecasting future sales of a product offers many advantages. Predicting future sales of a product helps a company manage the cost of manufacturing and marketing the product. In this notebook, we will try to you through the task of future sales prediction with machine learning using Python.

EDA Libraries:

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

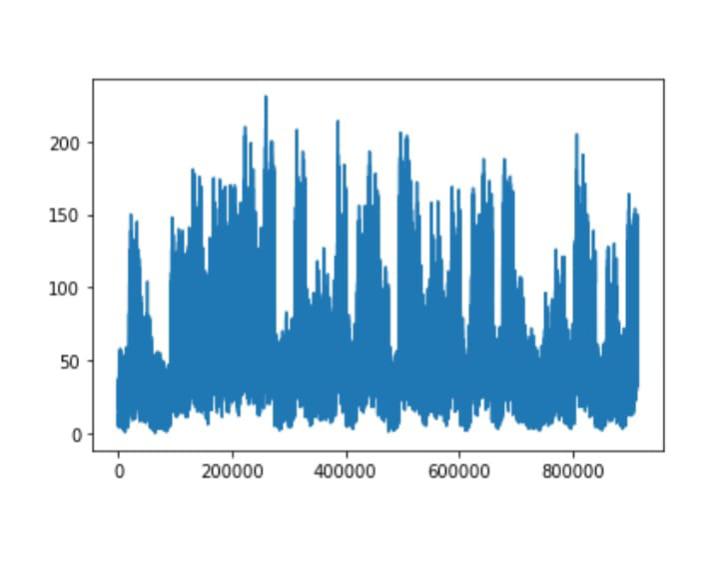
Import datetime

From pathlib import Path

Import random

DATA EXPLORATION:

Data exploration is the process of analyzing and visualizing a dataset to understand its characteristics and patterns. It involves tasks like summarizing data, identifying trends, detecting outliers, and generating insights. Data exploration helps researchers and analysts gain a better understanding of the data before diving into more in-depth analysis or modeling. It often includes techniques such as data visualization, descriptive statistics, and data cleaning to prepare the data for further analysis.

ACCURACY CHCK:

1.Mean Absolute Error (MAE): This measures the average absolute difference between predicted and actual sales. It gives an idea of the average magnitude of errors in your predictions.

2. Mean Squared Error (MSE): This squares the differences between predicted and actual sales before averaging them. It penalizes larger errors more heavily than MAE.

3. Root Mean Squared Error (RMSE): RMSE is the square root of MSE and provides a measure of the standard deviation of errors. It's often used to give a sense of the typical error in the prediction.

4. Mean Absolute Percentage Error (MAPE):MAPE calculates the percentage difference between predicted and actual sales. It's useful for understanding the relative error in your predictions.

5. R-squared (R2) Score: R2 measures the proportion of the variance in the dependent variable (sales) that is predictable from the independent variables in your model. A higher R2 indicates a better fit of your model to the data.

6. Forecast Bias: This measures the overall tendency of your predictions to be overestimates or underestimates. A positive bias indicates overestimation, while a negative bias indicates underestimation.

7. AIC and BIC: These are information criteria used in statistical modeling. Lower values indicate a better model fit.

8. Confusion Matrix (for classification problems): If your sales prediction problem involves classification (e.g., predicting whether a product will sell or not), you can use metrics like accuracy, precision, recall, and F1-score to evaluate the performance of your model.

9. Lift Curve (for marketing campaigns): If you're predicting sales for marketing campaigns, lift curves can help assess how much better your model performs compared to random targeting.

10. Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC) (for classification problems): These metrics are used to evaluate the performance of binary classifiers.

The choice of metrics depends on the specific nature of our sales prediction.

CONCLUSION:

This all are belongs into the future sales prediction. using this topics ,we can easily understand the project.