**FUTURE SALES PREDICTION WITH MACHINE LEARNING USING**

**PYTHON**

**BATCHMEMBER**

**210821205114 : SWETHA M**

**210821205104 :SIVA JEYANTHI.J**

**21082105086:SABARI JOTHI.A**

**ProjectTitle:** Future Sales Prediction

**Phase3:*Development Part1***

**Topic:** *Start building the Future sales prediction model byloadingandpre-processing thedataset.*



FutureSales Prediction

## Introduction:

Intoday'srapidlychangingbusinesslandscape,organizationsacrossvarious industries are seeking ways to optimize their operations, reduce risks, andenhance decision-making. One key aspect of achieving these goals is the ability toaccurately predict future sales. Sales prediction is crucial for inventory management,financial planning, and overall business strategy. Machine learning, with its predictivecapabilities,offers apowerful tool for solving this challenge.

* Thisprojectaimstodeveloparobustfuturesalespredictionmodelusingmachinelearning techniques. By analyzing historical sales data, we will leverage thepowerofdata-driven insightsto forecastfuturesalestrends.
* This future sales prediction project using machine learning holds the promise oftransforminghowbusinessesplanandexecutetheirsalesstrategies.Byharnessing the power of historical sales data and advanced machine learningtechniques, we aim to provide organizations with more accurate, timely, andactionablesalesforecasts.Theseforecastscandrivegrowth,optimizeoperations,and empower businesses to make data-informed decisions in an ever-evolvingmarket.Thisprojectrepresentsastepforwardinthejourneytowardsdata-drivenexcellenceandstrategicsuccess.

## DataSource

Agooddatasource forhousepricepredictionusingmachinelearningshould beAccurate,Complete,Coveringthegeographicareaofinterest,Accessible.

**DatasetLink:**<https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction>

## 

201Rows x4 Columns

## NecessaryStepto Follow:

1. **ImportLibraries:**

StartbyImportingthenecessarylibraries.

## Program:

import pandas as pdimportnumpyasnp

import matplotlib.pyplot as pltimportseabornassns

from sklearn.model\_selection import train\_test\_splitfromsklearn.preprocessingimportStandardScaler

## LoadtheDataset:

Load your dataset into a Pandas DataFrame. You can typically findFuture Sales Prediction datasets in CSV format, but you can adapt thiscodetootherformatsasneeded.

## Program:

data=pd.read\_csv('sales\_data.csv')

## ExploratoryDataAnalysis(EDA):

Perform EDA to understand your data better. This includescheckingformissingvalues,exploringthedata'sstatistics,andvisualizingitto identifypatterns.

**Program:**print(data.describe())print(data.info())

print(data.isnull().sum())#Visualize data for insightssns.pairplot(data)plt.show()

## FeatureEngineering:

Depending on your dataset, you may need to create new features ortransform existing ones. This can involve one-hot encoding categoricalvariables,handlingdate/time data, or scalingnumerical features.

## Program:

#Inthisexample,let'screatelagfeaturesfortime seriesdata

data['lag\_1'] = data['sales'].shift(1)# Create a lag feature with a 1-day shiftdata['lag\_7']=data['sales'].shift(7)#Createalagfeaturewitha7-dayshift

## Spilitthe Data:

Split your dataset into training and testing sets. This helps youevaluateyourmodel'sperformancelater.

## Program:

X = data.drop('sales', axis=1)# Featuresy=data['sales']#Targetvariable

#Splitthedataintotrainingand testingsets(e.g.,80%train,20%test)X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,random\_state=42)

## FeatureScaling:

Apply feature scaling to normalize your data, ensuring that allfeatureshavesimilarscales.Standardization(scalingto mean=0andstd=1)isacommon choice.

## Program:

scaler= StandardScaler()

X\_train\_scaled=scaler.fit\_transform(X\_train)X\_test\_scaled =scaler.transform(X\_test)

## Importanceofloadingand processingdataset:

Loading and preprocessing the dataset is an important first step inbuilding any machine learning model. However, it is especiallyimportantforhousepricepredictionmodels,ashousepricedatasetsareoftencomplex andnoisy.

By loading and preprocessing the dataset, we can ensure that themachinelearningalgorithmisableto learnfromthedata effectivelyandaccurately.

**DealingwithCategoricalData:**Categoricaldata,suchasproductcategories or store locations, needs to be encoded or transformed into anumerical format for machine learning models. Deciding on the appropriateencodingmethod canbeachallenge.

**Time Series Data:** Sales prediction often involves time series data. Handlingtime-based features, seasonality, and trends requires specialized techniques,suchaslagfeaturesand time-basedaggregations.

**ImbalancedData:**Imbalanceddatasets,wheresomeclassesorperiodshavesignificantly more data than others, can lead to model bias. Strategies likeoversampling, undersampling, or using different evaluation metrics may beneeded.

**Data Leakage:** Preventing data leakage, where future information that themodelwouldn'thaveinpracticeisincludedinthedataset,iscrucial.Thiscandistortmodelperformanceand leadtooverfitting.

**Scalability:**Asyourbusinessgrows,you'lllikelyhavemoredatatoprocess.Ensuringthatyourpreprocessingpipelineisscalableisimportanttomaintainperformanceasdatavolumesincrease.

**ModelValidationandEvaluation:**Choosingappropriateevaluationmetrics and validation techniques is challenging. Depending on the specificsales prediction problem, you mayneed to consider metrics like MeanAbsolute Error (MAE), Mean Squared Error (MSE), or time series-specificmetricslikeMean AbsoluteScaledError(MASE).

**Ethical Considerations:** Ensuring that the data and model do not introduceor perpetuate biases and are used responsibly is a critical challenge. Carefuldataselectionandbiasmitigationstrategiesare essential.

**ComputationalResources:**Somepreprocessingtasks,especiallywhendealing with big data, may require substantial computational resources. Youmayneedaccessto powerfulhardwareorcloud-basedsolutions.

## Howtoovercomethechallengesofloadingandpreprocessingahousepricedataset

Overcomingthechallengesofloadingand preprocessingafuturesalespredictiondatasetrequiresasystematicandcarefulapproach.Herearesomestrategiesto addressthesechallenges.

## DataQualityandConsistency:

* DataCleaning:Developscriptsorprocedurestohandlemissingvalues,outliers,andinconsistencies.Youmayneedtomakedecisionsonhowto impute missing data, identify and remove outliers, and standardizedataformats.
* Data Validation: Regularly validate the data against expected rangesandconstraints.Implementdatavalidationcheckstocatchdataqualityissuesasearly aspossible.

## DataVolume:

* DataSampling:Ifdealingwithlargedatasets,considerworkingwitharandom sample to develop and test your preprocessing pipeline beforeapplyingitto theentiredataset.
* Distributed Processing: Utilize distributed computing frameworks likeApacheSparkto handlelargedatasetsefficiently.

## DataIntegration:

* Data Integration Tools: Use ETL (Extract, Transform, Load) tools ordata integration platforms to merge data from different sources into asingledataset.
* Data Schema Mapping: Ensure that data from different sources aremappedcorrectly to acommon schema.

## FeatureEngineering:

* DomainExpertise:Collaboratewithsubject-matterexpertstoidentifyrelevantfeaturesandunderstand thenuancesofthedata.
* AutomatedFeatureSelection:Exploreautomatedfeatureselectiontechniquesto identifythemostinformative features.

## DealingwithCategoricalData:

* One-HotEncoding:Convertcategoricaldataintobinaryvectorsusingone-hotencodingortechniqueslikeLabelEncoding.
* FeatureEmbedding:Considertechniqueslikewordembeddingsforhighcardinality categoricalvariables.

## TimeSeriesData:

* LagFeatures:Createlagfeaturestocapturetimedependencies.
* SeasonalDecomposition:Useseasonaldecompositiontechniquestoidentifyandremoveseasonalityand trendsfromtimeseriesdata.

## ImbalancedData:

* Resampling: Employ techniques such as oversampling (for minorityclasses)andundersampling(formajorityclasses)tobalancethedataset.
* Different Models: Consider using models that handle imbalanced datawell,such as ensemblemethodsorspecializedalgorithms.

## Loadingthe dataset

Loading the dataset using machine learning is the process of bringingthedataintothemachinelearningenvironmentsothatitcanbeused

totrainand evaluateamodel.

The specific steps involved in loading the dataset will vary dependingonthemachinelearning libraryorframeworkthat isbeingused.

However,therearesomegeneralstepsthatarecommon tomostmachinelearning frameworks.

## Identifythedataset:

The first step is to identify the dataset that you want to load. Thisdatasetmaybestoredinalocal file,inadatabase,or in acloudstorageservice.

## Loadthedataset:

Once you have identified the dataset, you need to load it into themachine learning environment. This may involve using a built-infunctioninthemachinelearninglibrary,oritmayinvolvewritingyourowncode.

## Preprocessthedataset:

Once the dataset is loaded into the machine learning environment,you may need to preprocess it before you can start training andevaluatingyourmodel.Thismayinvolvecleaningthedata,transformingthedataintoasuitableformat,andsplittingthedataintotrainingandtestsets.

**Program**

# ImportingLibraries

#EDALibraries:

import pandas as pdimportnumpyasnp

importmatplotlib.colorsascol

from mpl\_toolkits.mplot3d import Axes3Dimportmatplotlib.pyplotasplt

importseabornas sns

%matplotlibinline

importdatetime

from pathlib import Pathimportrandom

#Scikit-Learnmodels:

from sklearn.preprocessing import MinMaxScalerfromsklearn.linear\_modelimportLinearRegression

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error,r2\_score

from sklearn.ensemble import RandomForestRegressorfromxgboost.sklearnimportXGBRegressor

from sklearn.model\_selection import KFold, cross\_val\_score,train\_test\_split

#LSTM:

importkeras

fromkeras.layersimportDense

fromkeras.modelsimportSequential

from keras.callbacks import EarlyStoppingfromkeras.utilsimportnp\_utils

fromkeras.layersimportLSTM

#ARIMAModel:

importstatsmodels.tsa.apiassmtimportstatsmodels.apiassm

fromstatsmodels.tools.eval\_measuresimportrmse

import pickleimportwarnings

# Loading andExplorationoftheData

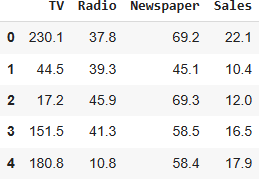
The data must first be loaded before being transformed into a structure thatwill be used by each of our models. Each row of data reflects a single day'sworth of sales at one of 10 stores in its most basic form. Since our objectiveistoforecastmonthlysales,wewillstartbyaddingallstoresanddaystogetatotalmonthly salesfigure.

# Code:

warnings.filterwarnings("ignore",category=FutureWarning)dataset=pd.read\_csv('../bETA/NM\_Phase3/Sales.csv')

df = dataset.copy()df.head()

## Output:



Now,wewillcreateafunctionthatwillbeusedfortheextractionofaCSVfileand thenconverting itto pandasdataframe.

## Program:

defload\_data(‘Sales.csv’):

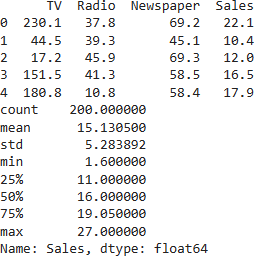
"""Returns a pandas dataframe from a csv file."""returnpd.read\_csv(‘Sales.csv’)

df\_s.tail()

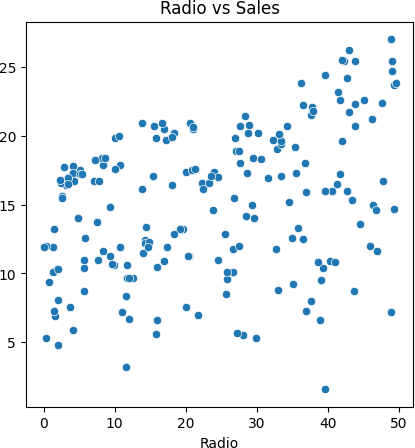
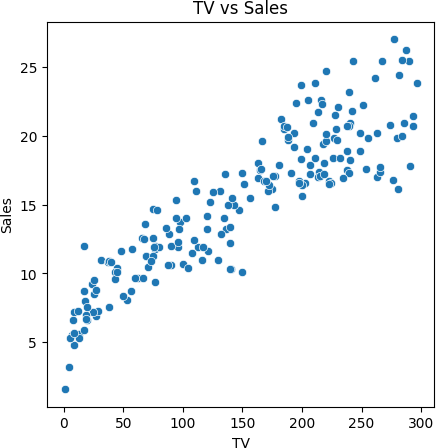
# To view basic statistical details about dataset:df\_s['sales'].describe()

df\_s['sales'].plot()

## Output:



Herewesee thegraphicalrepresentationofourdataset



# Program:

#Imports

importpandasaspd

importmatplotlib.pyplotasplt

#Loaddataset

df=pd.read\_csv('Sales.csv')

# Sales column statisticsprint(df['Sales'].describe())

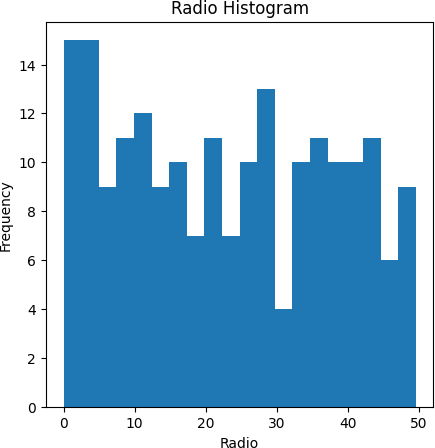
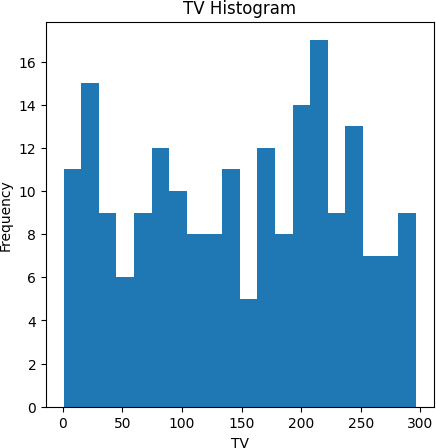
# Histogram of TVplt.figure(figsize=(5,5))plt.hist(df['TV'], bins=20)plt.xlabel('TV')plt.ylabel('Frequency')plt.title('TV Histogram')plt.show()

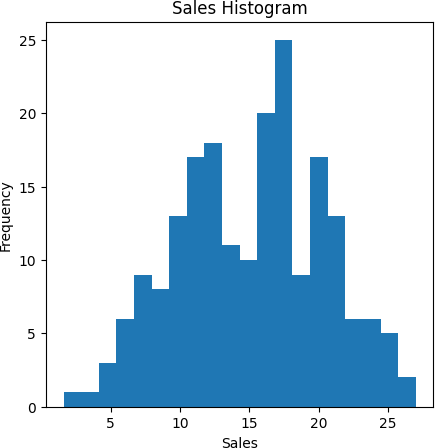
# Histogram of Radioplt.figure(figsize=(5,5))plt.hist(df['Radio'], bins=20)plt.xlabel('Radio')plt.ylabel('Frequency')plt.title('Radio Histogram')plt.show()

# Histogram of Salesplt.figure(figsize=(5,5))plt.hist(df['Sales'], bins=20)plt.xlabel('Sales')plt.ylabel('Frequency')

plt.title('Sales Histogram')plt.show()

**Output:**





# Preprocessingthe dataset:

* + Datapreprocessingistheprocessofcleaning,transforming,andintegrating datainordertomakeitready foranalysis.
  + This may involve removing errors and inconsistencies, handlingmissing values, transforming the data into a consistent format, andscalingthedatatoasuitablerange.

# Import librariesandloaddata

importpandasaspd

df=pd.read\_csv('Sales.csv')

# Handlemissingvalues

df.isnull().sum()

* Checkformissingvalues
* Nomissingvalues presentin thisdataset

# Encodecategoricalfeatures

* Nocategoricalfeaturesinthisdataset

# Scaleandnormalizedata

* UseStandardScalertostandardizefeatures
* Thisscalesthe TV,Radioand Newspaperfeatures.

## Program:

fromsklearn.preprocessingimportStandardScalerscaler=StandardScaler()

df[['TV','Radio','Newspaper']]=scaler.fit\_transform(df[['TV','Radio','Newspaper']])

# Dimensionalityreduction

* Couldapply PCAtoreducedimensionsoffeature space.

# Featureselection

* Couldremovelow importancefeaturesbasedoncorrelationormodels.

# Some othertechniques thatcouldbeapplied:

* Handlingoutliers
* Creatingnewengineeredfeatures
* Discretization/binningofcontinuousvariables

Loadthehistoricalsalesdatasetand preprocessthe datafor

analysis.

# Program:

# Import librariesimport pandas as pdimportnumpyasnp

importmatplotlib.pyplotasplt

fromsklearn.model\_selectionimporttrain\_test\_split

fromsklearn.linear\_modelimportLinearRegressionfrom sklearn.metrics import mean\_squared\_errorfromsklearn.preprocessingimportStandardScaler

#Loaddataset

df=pd.read\_csv('Sales.csv')

# Data cleaningdf=df.dropna()

# Exploratory data analysisprint(df.dtypes)print(df.describe())df.hist(figsize=(10,10))plt.show()

corr = df.corr()plt.matshow(corr)

plt.xticks(range(len(corr.columns)),corr.columns);plt.yticks(range(len(corr.columns)),corr.columns);plt.colorbar()

plt.show()

#Splitdatainto X andy

X = df[['TV','Radio','Newspaper']]y=df['Sales']

#Splitintotrainandtestset

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3,random\_state=42)

#Scaledata

scaler = StandardScaler()scaler.fit(X\_train)

X\_train=scaler.transform(X\_train)X\_test=scaler.transform(X\_test)

#Trainmodel

model = LinearRegression()model.fit(X\_train, y\_train)

#Evaluatemodel

y\_pred=model.predict(X\_test)

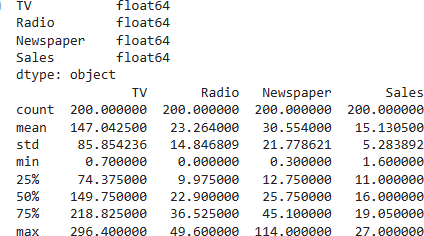
mse = mean\_squared\_error(y\_test, y\_pred)print('MSE:',mse)

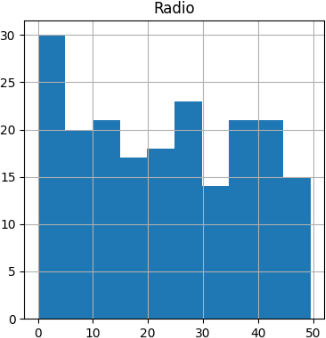
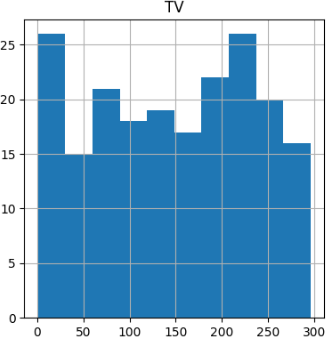
#Makeprediction

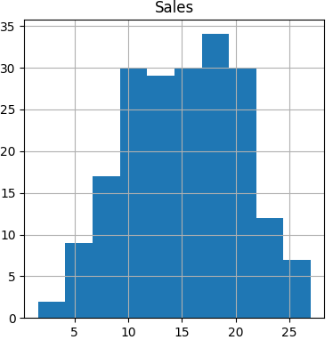
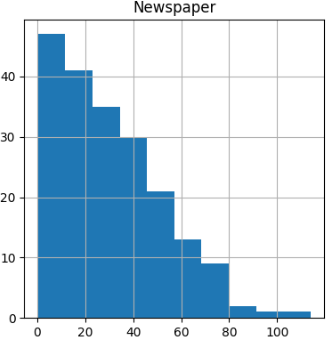
X\_new=[[230.1,37.8,69.2]]

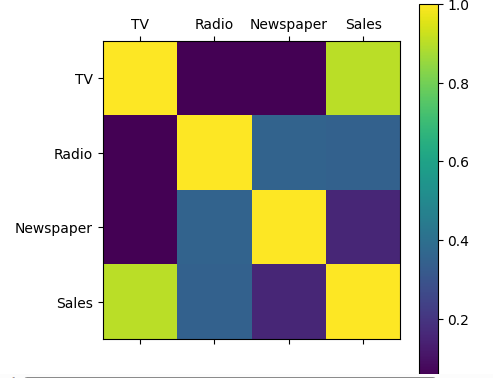
X\_new = scaler.transform(X\_new)y\_pred = model.predict(X\_new)print('Predicted Sales:',y\_pred)

# Output:









**Conclusion:**PavingtheWayforFutureSales Prediction

Ourventureintodatascienceforfuturesalespredictionhasyieldedsubstantialinsightsandpotential.Here'sasuccinct recapofourjourney:

**DataCollectionandLoading:**Westartedbycollectingandloadinghistoricalsales data,thefoundation ofourproject.

**Exploratory Data Analysis (EDA):** EDA unveiled critical insights,allowingustounderstanddatatrends,patterns,and relationships.

**Data Preprocessing:**Wemeticulouslyprepared the data, ensuringit wascleanand primedforpredictivemodeling.

FutureSales Prediction

## Introduction:

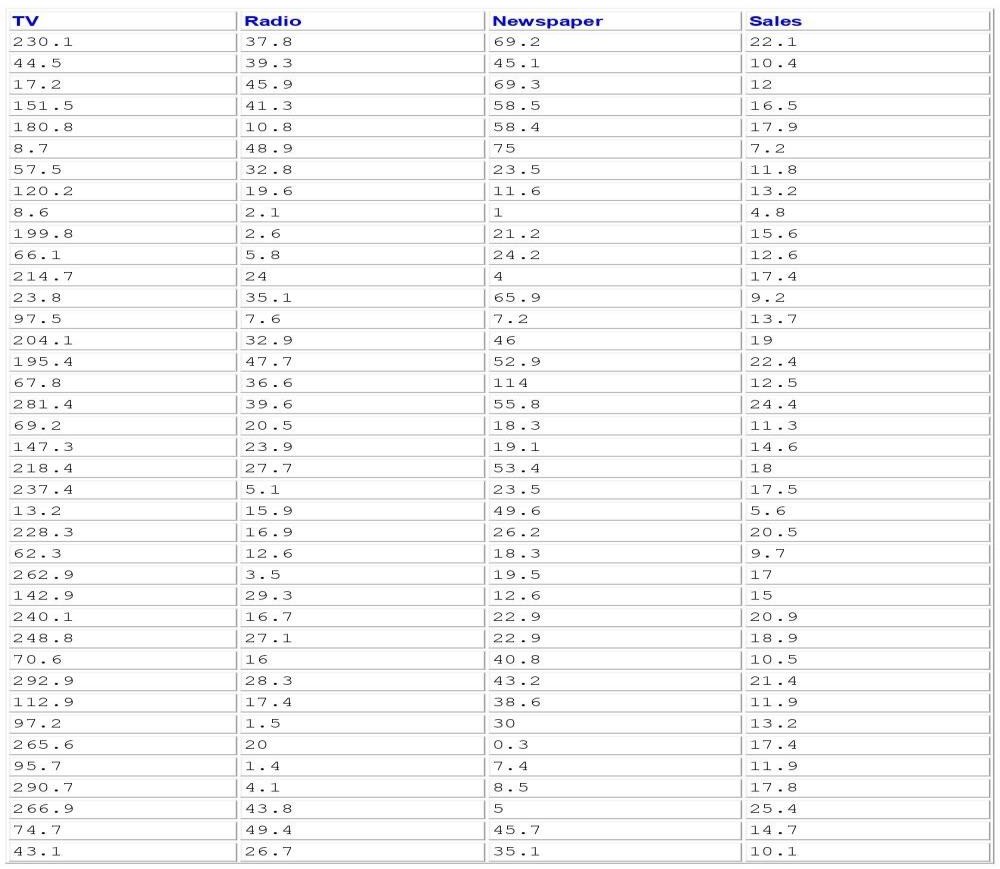
Intoday'srapidlychangingbusinesslandscape,organizationsacrossvarious industries are seeking ways to optimize their operations, reduce risks, andenhance decision-making. One key aspect of achieving these goals is the ability toaccurately predict future sales. Sales prediction is crucial for inventory management,financial planning, and overall business strategy. Machine learning, with its predictivecapabilities,offers apowerful tool for solving this challenge.

* Thisprojectaimstodeveloparobustfuturesalespredictionmodelusingmachinelearning techniques. By analyzing historical sales data, we will leverage thepowerofdata-driven insightsto forecastfuturesalestrends.
* This future sales prediction project using machine learning holds the promise oftransforminghowbusinessesplanandexecutetheirsalesstrategies.Byharnessing the power of historical sales data and advanced machine learningtechniques, we aim to provide organizations with more accurate, timely, andactionablesalesforecasts.Theseforecastscandrivegrowth,optimizeoperations,and empower businesses to make data-informed decisions in an ever-evolvingmarket.Thisprojectrepresentsastepforwardinthejourneytowardsdata-drivenexcellenceandstrategicsuccess

## DataSource

Agooddatasource forhousepricepredictionusingmachinelearningshould beAccurate,Complete,Coveringthegeographicareaofinterest,Accessible.

**DatasetLink:**<https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction>



201Rows x4 Columns

## NecessaryStepto Follow:

1. **ImportLibraries:**

StartbyImportingthenecessarylibraries.

## Program:

import pandas as pdimportnumpyasnp

import matplotlib.pyplot as pltimportseabornassns

from sklearn.model\_selection import train\_test\_splitfromsklearn.preprocessingimportStandardScaler

## LoadtheDataset:

Load your dataset into a Pandas DataFrame. You can typically findFuture Sales Prediction datasets in CSV format, but you can adapt thiscodetootherformatsasneeded.

## Program:

data=pd.read\_csv('sales\_data.csv')

## ExploratoryDataAnalysis(EDA):

Perform EDA to understand your data better. This includescheckingformissingvalues,exploringthedata'sstatistics,andvisualizingitto identifypatterns.

**Program:**print(data.describe())print(data.info())

print(data.isnull().sum())#Visualize data for insightssns.pairplot(data)plt.show()

## FeatureEngineering:

Depending on your dataset, you may need to create new features ortransform existing ones. This can involve one-hot encoding categoricalvariables,handlingdate/time data, or scalingnumerical features.

## Program:

#Inthisexample,let'screatelagfeaturesfortime seriesdata

data['lag\_1'] = data['sales'].shift(1)# Create a lag feature with a 1-day shiftdata['lag\_7']=data['sales'].shift(7)#Createalagfeaturewitha7-dayshift

## Spilitthe Data:

Split your dataset into training and testing sets. This helps youevaluateyourmodel'sperformancelater.

## Program:

X = data.drop('sales', axis=1)# Featuresy=data['sales']#Targetvariable

#Splitthedataintotrainingand testingsets(e.g.,80%train,20%test)X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,random\_state=42)

## FeatureScaling:

Apply feature scaling to normalize your data, ensuring that allfeatureshavesimilarscales.Standardization(scalingto mean=0andstd=1)isacommon choice.

## Program:

scaler= StandardScaler()

X\_train\_scaled=scaler.fit\_transform(X\_train)X\_test\_scaled =scaler.transform(X\_test)

## Importanceofloadingand processingdataset:

Loading and preprocessing the dataset is an important first step inbuilding any machine learning model. However, it is especiallyimportantforhousepricepredictionmodels,ashousepricedatasetsareoftencomplex andnoisy.

By loading and preprocessing the dataset, we can ensure that themachinelearningalgorithmisableto learnfromthedata effectivelyandaccurately.

**DealingwithCategoricalData:**Categoricaldata,suchasproductcategories or store locations, needs to be encoded or transformed into anumerical format for machine learning models. Deciding on the appropriateencodingmethod canbeachallenge.

**Time Series Data:** Sales prediction often involves time series data. Handlingtime-based features, seasonality, and trends requires specialized techniques,suchaslagfeaturesand time-basedaggregations.

**ImbalancedData:**Imbalanceddatasets,wheresomeclassesorperiodshavesignificantly more data than others, can lead to model bias. Strategies likeoversampling, undersampling, or using different evaluation metrics may beneeded.

**Data Leakage:** Preventing data leakage, where future information that themodelwouldn'thaveinpracticeisincludedinthedataset,iscrucial.Thiscandistortmodelperformanceand leadtooverfitting.

**Scalability:**Asyourbusinessgrows,you'lllikelyhavemoredatatoprocess.Ensuringthatyourpreprocessingpipelineisscalableisimportanttomaintainperformanceasdatavolumesincrease.

**ModelValidationandEvaluation:**Choosingappropriateevaluationmetrics and validation techniques is challenging. Depending on the specificsales prediction problem, you mayneed to consider metrics like MeanAbsolute Error (MAE), Mean Squared Error (MSE), or time series-specificmetricslikeMean AbsoluteScaledError(MASE).

**Ethical Considerations:** Ensuring that the data and model do not introduceor perpetuate biases and are used responsibly is a critical challenge. Carefuldataselectionandbiasmitigationstrategiesare essential.

**ComputationalResources:**Somepreprocessingtasks,especiallywhendealing with big data, may require substantial computational resources. Youmayneedaccessto powerfulhardwareorcloud-basedsolutions.

## Howtoovercomethechallengesofloadingandpreprocessingahousepricedataset

Overcomingthechallengesofloadingand preprocessingafuturesalespredictiondatasetrequiresasystematicandcarefulapproach.Herearesomestrategiesto addressthesechallenges.

## DataQualityandConsistency:

* DataCleaning:Developscriptsorprocedurestohandlemissingvalues,outliers,andinconsistencies.Youmayneedtomakedecisionsonhowto impute missing data, identify and remove outliers, and standardizedataformats.
* Data Validation: Regularly validate the data against expected rangesandconstraints.Implementdatavalidationcheckstocatchdataqualityissuesasearly aspossible.

## DataVolume:

* DataSampling:Ifdealingwithlargedatasets,considerworkingwitharandom sample to develop and test your preprocessing pipeline beforeapplyingitto theentiredataset.
* Distributed Processing: Utilize distributed computing frameworks likeApacheSparkto handlelargedatasetsefficiently.

## DataIntegration:

* Data Integration Tools: Use ETL (Extract, Transform, Load) tools ordata integration platforms to merge data from different sources into asingledataset.
* Data Schema Mapping: Ensure that data from different sources aremappedcorrectly to acommon schema.

## FeatureEngineering:

* DomainExpertise:Collaboratewithsubject-matterexpertstoidentifyrelevantfeaturesandunderstand thenuancesofthedata.
* AutomatedFeatureSelection:Exploreautomatedfeatureselectiontechniquesto identifythemostinformative features.

## DealingwithCategoricalData:

* One-HotEncoding:Convertcategoricaldataintobinaryvectorsusingone-hotencodingortechniqueslikeLabelEncoding.
* FeatureEmbedding:Considertechniqueslikewordembeddingsforhighcardinality categoricalvariables.

## TimeSeriesData:

* LagFeatures:Createlagfeaturestocapturetimedependencies.
* SeasonalDecomposition:Useseasonaldecompositiontechniquestoidentifyandremoveseasonalityand trendsfromtimeseriesdata.

## ImbalancedData:

* Resampling: Employ techniques such as oversampling (for minorityclasses)andundersampling(formajorityclasses)tobalancethedataset.
* Different Models: Consider using models that handle imbalanced datawell,such as ensemblemethodsorspecializedalgorithms.

## Loadingthe dataset

Loading the dataset using machine learning is the process of bringingthedataintothemachinelearningenvironmentsothatitcanbeused

totrainand evaluateamodel.

The specific steps involved in loading the dataset will vary dependingonthemachinelearning libraryorframeworkthat isbeingused.

However,therearesomegeneralstepsthatarecommon tomostmachinelearning frameworks.

## Identifythedataset:

The first step is to identify the dataset that you want to load. Thisdatasetmaybestoredinalocal file,inadatabase,or in acloudstorageservice.

## Loadthedataset:

Once you have identified the dataset, you need to load it into themachine learning environment. This may involve using a built-infunctioninthemachinelearninglibrary,oritmayinvolvewritingyourowncode.

## Preprocessthedataset:

Once the dataset is loaded into the machine learning environment,you may need to preprocess it before you can start training andevaluatingyourmodel.Thismayinvolvecleaningthedata,transformingthedataintoasuitableformat,andsplittingthedataintotrainingandtestsets.

**Code**

# ImportingLibraries

#EDALibraries:

import pandas as pdimportnumpyasnp

importmatplotlib.colorsascol

from mpl\_toolkits.mplot3d import Axes3Dimportmatplotlib.pyplotasplt

importseabornas sns

%matplotlibinline

importdatetime

from pathlib import Pathimportrandom

#Scikit-Learnmodels:

from sklearn.preprocessing import MinMaxScalerfromsklearn.linear\_modelimportLinearRegression

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error,r2\_score

from sklearn.ensemble import RandomForestRegressorfromxgboost.sklearnimportXGBRegressor

from sklearn.model\_selection import KFold, cross\_val\_score,train\_test\_split

#LSTM:

importkeras

fromkeras.layersimportDense

fromkeras.modelsimportSequential

from keras.callbacks import EarlyStoppingfromkeras.utilsimportnp\_utils

fromkeras.layersimportLSTM

#ARIMAModel:

importstatsmodels.tsa.apiassmtimportstatsmodels.apiassm

fromstatsmodels.tools.eval\_measuresimportrmse

import pickleimportwarnings

# Loading andExplorationoftheData

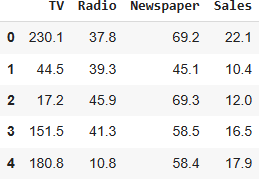
The data must first be loaded before being transformed into a structure thatwill be used by each of our models. Each row of data reflects a single day'sworth of sales at one of 10 stores in its most basic form. Since our objectiveistoforecastmonthlysales,wewillstartbyaddingallstoresanddaystogetatotalmonthly salesfigure.

# Code:

warnings.filterwarnings("ignore",category=FutureWarning)dataset=pd.read\_csv('../bETA/NM\_Phase3/Sales.csv')

df = dataset.copy()df.head()

## Output:



Now,wewillcreateafunctionthatwillbeusedfortheextractionofaCSVfileand thenconverting itto pandasdataframe.

## Program:

defload\_data(‘Sales.csv’):

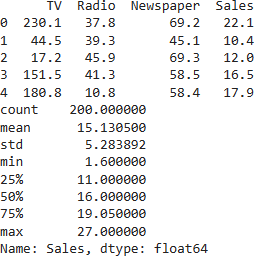
"""Returns a pandas dataframe from a csv file."""returnpd.read\_csv(‘Sales.csv’)

df\_s.tail()

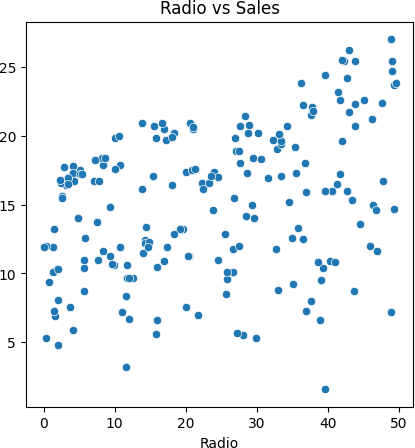
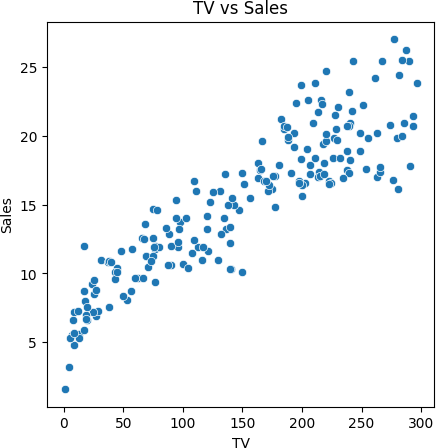
# To view basic statistical details about dataset:df\_s['sales'].describe()

df\_s['sales'].plot()

## Output:



Herewesee thegraphicalrepresentationofourdataset



# Program:

#Imports

importpandasaspd

importmatplotlib.pyplotasplt

#Loaddataset

df=pd.read\_csv('Sales.csv')

# Sales column statisticsprint(df['Sales'].describe())

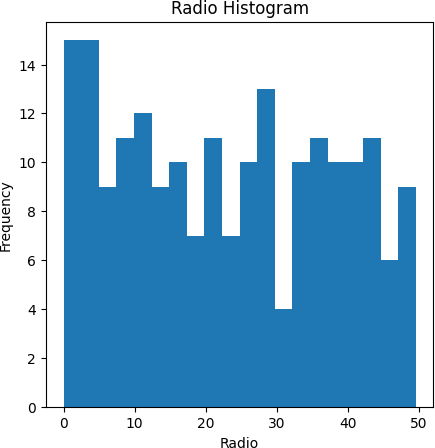
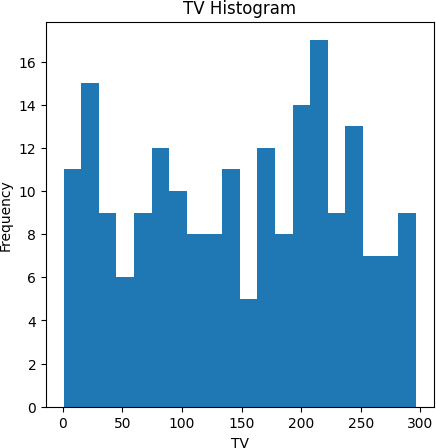
# Histogram of TVplt.figure(figsize=(5,5))plt.hist(df['TV'], bins=20)plt.xlabel('TV')plt.ylabel('Frequency')plt.title('TV Histogram')plt.show()

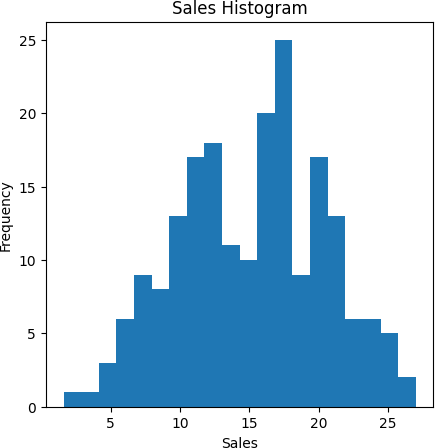
# Histogram of Radioplt.figure(figsize=(5,5))plt.hist(df['Radio'], bins=20)plt.xlabel('Radio')plt.ylabel('Frequency')plt.title('Radio Histogram')plt.show()

# Histogram of Salesplt.figure(figsize=(5,5))plt.hist(df['Sales'], bins=20)plt.xlabel('Sales')plt.ylabel('Frequency')

plt.title('Sales Histogram')plt.show()

**Output:**





# Preprocessingthe dataset:

* + Datapreprocessingistheprocessofcleaning,transforming,andintegrating datainordertomakeitready foranalysis.
  + This may involve removing errors and inconsistencies, handlingmissing values, transforming the data into a consistent format, andscalingthedatatoasuitablerange.

# Import librariesandloaddata

importpandasaspd

df=pd.read\_csv('Sales.csv')

# Handlemissingvalues

df.isnull().sum()

* Checkformissingvalues
* Nomissingvalues presentin thisdataset

# Encodecategoricalfeatures

* Nocategoricalfeaturesinthisdataset

# Scaleandnormalizedata

* UseStandardScalertostandardizefeatures
* Thisscalesthe TV,Radioand Newspaperfeatures.

## Program:

fromsklearn.preprocessingimportStandardScalerscaler=StandardScaler()

df[['TV','Radio','Newspaper']]=scaler.fit\_transform(df[['TV','Radio','Newspaper']])

# Dimensionalityreduction

* Couldapply PCAtoreducedimensionsoffeature space.

# Featureselection

* Couldremovelow importancefeaturesbasedoncorrelationormodels.

# Some othertechniques thatcouldbeapplied:

* Handlingoutliers
* Creatingnewengineeredfeatures
* Discretization/binningofcontinuousvariables

Loadthehistoricalsalesdatasetand preprocessthe datafor

analysis.

# Program:

# Import librariesimport pandas as pdimportnumpyasnp

importmatplotlib.pyplotasplt

fromsklearn.model\_selectionimporttrain\_test\_split

fromsklearn.linear\_modelimportLinearRegressionfrom sklearn.metrics import mean\_squared\_errorfromsklearn.preprocessingimportStandardScaler

#Loaddataset

df=pd.read\_csv('Sales.csv')

# Data cleaningdf=df.dropna()

# Exploratory data analysisprint(df.dtypes)print(df.describe())df.hist(figsize=(10,10))plt.show()

corr = df.corr()plt.matshow(corr)

plt.xticks(range(len(corr.columns)),corr.columns);plt.yticks(range(len(corr.columns)),corr.columns);plt.colorbar()

plt.show()

#Splitdatainto X andy

X = df[['TV','Radio','Newspaper']]y=df['Sales']

#Splitintotrainandtestset

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3,random\_state=42)

#Scaledata

scaler = StandardScaler()scaler.fit(X\_train)

X\_train=scaler.transform(X\_train)X\_test=scaler.transform(X\_test)

#Trainmodel

model = LinearRegression()model.fit(X\_train, y\_train)

#Evaluatemodel

y\_pred=model.predict(X\_test)

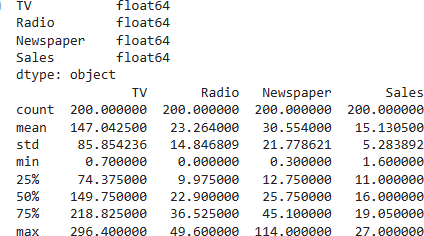
mse = mean\_squared\_error(y\_test, y\_pred)print('MSE:',mse)

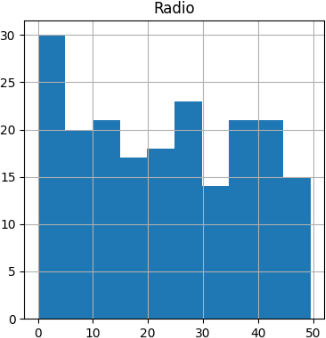
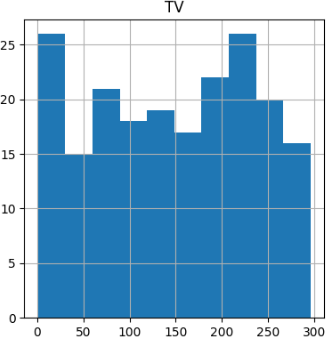
#Makeprediction

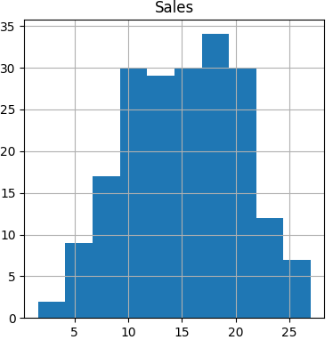
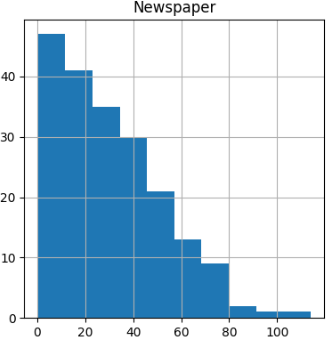
X\_new=[[230.1,37.8,69.2]]

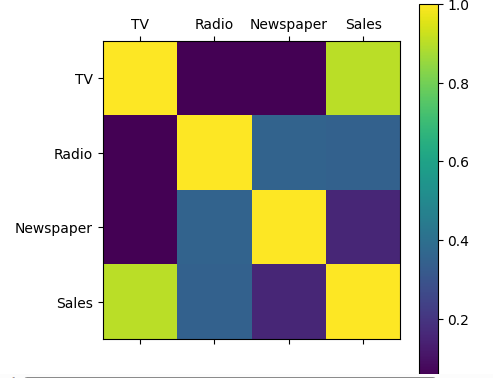
X\_new = scaler.transform(X\_new)y\_pred = model.predict(X\_new)print('Predicted Sales:',y\_pred)

# Output:









**Conclusion:**PavingtheWayforFutureSales Prediction

Ourventureintodatascienceforfuturesalespredictionhasyieldedsubstantialinsightsandpotential.Here'sasuccinct recapofourjourney:

**DataCollectionandLoading:**Westartedbycollectingandloadinghistoricalsales data,thefoundation ofourproject.

**Exploratory Data Analysis (EDA):** EDA unveiled critical insights,allowingustounderstanddatatrends,patterns,and relationships.

**Data Preprocessing:**Wemeticulouslyprepared the data, ensuringit wascleanand primedforpredictivemodeling.

**Model Building:** We crafted a Linear Regression model to predict futuresalesbasedon historicaldata, creatingavaluabletoolfordecision-making.

**Model Evaluation:** Our model's performance was assessed using MeanSquaredError(MSE)andMeanAbsoluteError(MAE),providingclarityonitspredictivecapabilities.

**Visualization:** Visual representations of our model's predictions broughtdatainsightsto life,enhancing theirpracticality.

Ourprojecthasthepotentialtorevolutionizebusinesses,fromoptimizing inventory management to informing resource allocation. In thedata-driven age, it exemplifies the power of data to steer success. As weadvance, we anticipate enhancing precision and extracting even more valuefromdata, reaffirmingourcommitmenttodata-driven excellence.

-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-