**Applied DataScience**

## Project Name : Future Sales Prediction

## Phase 3 : Developement Part 1

Description : Begin building the future sales prediction model by loading and preprocessing the dataset. Load the historical sales dataset and preprocess the data for analysis.

Dataset Link :<https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction>

## Procedure :

Following steps are used to load and preprocess the dataset for “Future Sales Prediction”.

## Step 1 :

Import the python libraries with below code.

Import pandas as pd

## Step 2 :

Load the dataset with below code.

FSP\_1=pd.read\_csv(‘<https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction>/download’)

Output :

0 TV Radio Newspaper Sales

1 230.1 37.8 69.2 22.1

2 44.5 39.3 45.1 10.4

3 17.2 45.9 69.3 12

4 151.5 41.3 58.5 16.5

… … … … … … … …

199 177 9.3 6.4 14.8

200 283.6 42 66.2 25.5

201 232.1 8.6 8.7 18.4

[201 rows x 4 columns]

## Step 3:

Print the basic information about the dataset with below code.

FSP\_1.info()

Output :

Data columns (total 4 columns):

# Column Non-Null Count Dtype

— —------ —----------- —----

0 TV non-null 201 int64

1 Radio non-null 201 int64

2 NewsPapernon-null 201 int64

3 Sales non-null 201 int64

Dtypes: int64(4)

Memory usage:6.0+ KB

## Step 4:

Preprocess the dataset

Check missing values with below code

FSP\_1.isnull().sum()

Output :

TV 0

Radio 0

Newspaper 0

Sales 0

Dtype:int64

Convert categorical features into numerical features

* Define a function to convert categorical features into numerical features.

def encode\_categorical\_feature(df,column):

Return pd.get\_dummies(df[column,drop\_first=True)

* Encode the TV value.

FSP\_2=encode\_categorical\_feature(FSP\_1,’TV’)

* Encode the Radio value.

FSP\_2=encode\_categorical\_feature(FSP\_1,’Radio’)

Output :

TV Radio NewsPaper Sales

0 69.2 22.1

## Step 5 :

Scale the numerical feature.

* Define a function to scale numerical feature.

from sklearn.preprocessing import StandardScaler

Def scale\_numerical\_feature(df,columns):

scaler=StandardScaler()

Scaled\_df=scaler.fit\_transform(df[columns])

Return scaled\_df

* Scale the numerical feature

Numerical\_feature=[‘Sales’]

FSP\_2=pd.concat([FSP\_1,scale\_numerical\_feature(FSP\_1,numerical\_feature)],axis=1)

## Step 6 :

Split the dataset into training and test sets with below code.

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

X=FSP\_1.drop(‘TV’,axis=1)

Y=FSP\_1[‘TV’]

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.25,random\_state=42)

Output :

(156,46)

This means that the training set contains 156 samples and test set contains 46 samples.

## Conclusion :

We have now loaded and preprocesed the future sales prediction dataset.