

```
In [268]: #import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

In [269]: data = pd.read_csv("Top_Crop.csv")
data.head()
```

Sl no	District_Name	Commodity	Max Price	Year	Production	Demand	
0	1	Banglore	Tomato	2000	2016	8567	8778
1	2	Banglore	Tomato	1600	2018	12357	12302
2	3	Mysore	Potato	1500	2016	9466	9466
3	4	Mysore	Onion	1500	2016	9076	9076
4	5	Banglore	Tomato	1500	2018	15646	15646

```
In [270]: data.isnull().sum()

Out[270]: Sl no          0
District_Name  0
Commodity      0
Max Price      0
Year           0
Production     0
Demand        0
dtype: int64

In [271]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 7 columns):
#   Column              Non-Null Count  Dtype
---  --
0   Sl no               20 non-null    int64
1   District_Name       20 non-null    object
2   Commodity           20 non-null    object
3   Max Price           20 non-null    int64
4   Year                20 non-null    int64
5   Production          20 non-null    int64
6   Demand              20 non-null    int64
dtypes: int64(5), object(2)
memory usage: 1.2+ KB

In [272]: data.columns

Out[272]: Index(['Sl no', 'District_Name', 'Commodity', 'Max Price', 'Year',
               'Production', 'Demand'],
              dtype='object')

In [273]: data["Commodity"].value_counts()

Out[273]: Tomato    7
          Potato    7
          Onion     3
          Name: Commodity, dtype: int64

In [274]: data["District_Name"].value_counts()

Out[274]: Bangalore    8
          Mysore       7
          Kolar         5
          Name: District_Name, dtype: int64

In [275]: sns.distplot(x=data["Demand"])

C:\Users\Lenovo\AppData\Local\Temp\ipykernel_11360\1271645973.py:1: UserWarning:
'distplot' is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either 'displot' (a figure-level function with
similar flexibility) or 'histplot' (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
sns.distplot(x=data["Demand"])

Out[275]: <AxesSubplot:ylabel='Density'>
```



```
In [276]: sns.distplot(x=data["Max Price"])

C:\Users\Lenovo\AppData\Local\Temp\ipykernel_11360\127164392.py:1: UserWarning:
'distplot' is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either 'displot' (a figure-level function with
similar flexibility) or 'histplot' (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
sns.distplot(x=data["Max Price"])

Out[276]: <AxesSubplot:ylabel='Density'>
```



```
In [277]: sns.countplot(x=data["Commodity"])

Out[277]: <AxesSubplot:xlabel='Commodity', ylabel='count'>
```



```
In [278]: sns.countplot(x=data["District_Name"])

Out[278]: <AxesSubplot:xlabel='District_Name', ylabel='count'>
```



```
In [279]: sns.barplot(data=data, x = "Year", y = "Demand", hue = "Commodity")

Out[279]: <AxesSubplot:xlabel='Year', ylabel='Demand'>
```



```
In [280]: crop = sns.lineplot(data=data, x="Max Price", y="Demand", color="#4c934c")
crop.set_title("Price V/S Demand")

Out[280]: Text(0.5, 1.0, 'Price V/S Demand')
```



```
In [281]: data.plot(x = "Year", y = "Demand", style = 'o' )
plt.title('Year v/s Demand')
plt.xlabel('Year')
plt.ylabel('Demand')
plt.show()
```



```
In [282]: data.plot(x = 'Max Price', y = 'Demand', style = 'o' )
plt.title('Price v/s Demand')
plt.xlabel('Price')
plt.ylabel('Demand')
plt.show()
```



```
In [283]: plt.figure(figsize=(8,4))
sns.heatmap(data.corr(),annot=True,fmt="0.1f")

Out[283]: <AxesSubplot:>
```

	Sl no	Max Price	Year	Production	Demand
Sl no	1.0	0.3	0.1	0.2	0.2
Max Price	0.3	1.0	0.2	0.1	0.1
Year	0.1	0.2	1.0	0.2	0.1
Production	0.2	0.1	0.2	1.0	1.0
Demand	0.2	0.1	0.1	1.0	1.0

```
In [284]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data["District_Name"] = le.fit_transform(data["District_Name"])
data.head()
```

Sl no	District_Name	Commodity	Max Price	Year	Production	Demand
0	1	0	2000	2016	8567	8778
1	2	0	1600	2018	12357	12302
2	3	2	1500	2016	9466	9466
3	4	2	1500	2016	9076	9076
4	5	0	1500	2018	15646	15646

```
In [285]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data["Commodity"] = le.fit_transform(data["Commodity"])
data.head()
```

Sl no	District_Name	Commodity	Max Price	Year	Production	Demand	
0	1	0	2	2000	2016	8567	8778
1	2	0	2	1600	2018	12357	12302
2	3	2	1	1500	2016	9466	9466
3	4	2	0	1500	2016	9076	9076
4	5	0	2	1500	2018	15646	15646

```
In [286]: x=data.iloc[:, :-1]
y=data.iloc[:, -1]
x.head()
```

Sl no	District_Name	Commodity	Max Price	Year	Production
0	1	0	2	2000	2016
1	2	0	2	1600	2018
2	3	2	1	1500	2016
3	4	2	0	1500	2016
4	5	0	2	1500	2018

```
In [287]: y.head()
```

0	8778
1	12382
2	9466
3	9876
4	15646

```
In [288]: from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size = 0.2,random_state = 2)

In [289]: xtrain.shape

Out[289]: (16, 6)

In [290]: ytrain.shape

Out[290]: (16,)
```

```
In [291]: xtest.shape

Out[291]: (4, 6)

In [292]: ytest.shape

Out[292]: (4,)
```

```
In [293]: from sklearn.linear_model import LinearRegression
data=LinearRegression()

In [294]: data.fit(xtrain,ytrain)

Out[294]: LinearRegression
LinearRegression()

In [295]: data.coef_

Out[295]: array([[ 3.21723699e+00,  2.51986739e+01, -4.10596841e+01, -2.35658550e+02,
                -3.26313724e+01,  9.87693079e-01]])

In [296]: data.intercept_

Out[296]: 65957.46871233374

In [297]: ypred = data.predict(xtest)

Out[297]: array([[11123.49892721, 15459.42312057, 11819.3006548 ,  8508.15468156]])

In [298]: data.score(xtest,ytest)

Out[298]: 0.994153490965179

In [300]: data.score(xtest,ytest)*100,"%"

Out[300]: (99.4153490965179, '%')

In [ ]:

In [ ]:
```