```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import sklearn as sk
```

```
df=pd.read_excel("C:/[illegible]DATA/data 3/online/online online/Projects/Product demand forecasting/Online Retail.xlsx")
df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Data Understanding

```
df.shape
```

```
(541909, 8)
```

```
df.isnull().sum()
```

```
InvoiceNo         0
StockCode         0
Description    1454
Quantity          0
InvoiceDate       0
UnitPrice         0
CustomerID   135080
Country           0
dtype: int64
```

```
df.duplicated().sum()
```

```
5268
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541909
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   InvoiceNo   541909 non-null  object
 1   StockCode   541909 non-null  object
 2   Description 540455 non-null  object
 3   Quantity    541909 non-null  int64
 4   InvoiceDate 541909 non-null  datetime64[ns]
 5   UnitPrice   541909 non-null  float64
 6   CustomerID  406829 non-null  float64
 7   Country     541909 non-null  object
dtypes: datetime64[ns](1), float64(3), int64(1), object(4)
memory usage: 33.1 MB
```

```
df.describe()
```

	Quantity	InvoiceDate	UnitPrice	CustomerID
count	541909.000000	541909	541909.000000	406829.000000
mean	9.552250	2011-07-04 12:36:07.999999744	4.611114	15287.690570
min	-80995.000000	2010-12-01 08:26:00	-11062.060000	12346.000000
25%	1.000000	2011-03-28 11:34:00	1.250000	13953.000000
50%	3.000000	2011-07-19 17:17:00	2.080000	15152.000000
75%	10.000000	2011-10-19 11:37:00	4.130000	16791.000000
max	80995.000000	2011-12-09 12:50:00	38970.000000	18287.000000
std	218.081158	NaN	96.759853	1713.600303

```
df.describe(include='object')
```

	InvoiceNo	StockCode	Description	Country
count	541909	541909	540455	541909
unique	25900	4070	4223	38
top	573585	85123A	WHITE HANGING HEART T-LIGHT HOLDER	United Kingdom
freq	1114	2313	2369	495478

Data Handling

```
df_ind2.drop(columns=['Description', 'CustomerID'])
df_ind2.shape
```

```
(406829, 6)
```

```
df_ind2.[illegible]drop_duplicates()
```

```
print("Null values in dataset:\n",df_ind2.isnull().sum())
print("Duplicated values in a dataset:",df_ind2.duplicated.sum())
```

```
Null values in a dataset
 InvoiceNo    0
StockCode    0
Quantity     0
InvoiceDate  0
UnitPrice    0
Country      0
dtype: int64
Duplicated values in a dataset:0
```

```
# Filter invoices/cancellation
```

```
df_ind2["InvoiceNo"] = df_ind2["InvoiceNo"].astype(str)
df_ind2 = df_ind2[df_ind2["InvoiceNo"].str.startswith("C")]
df_ind2.shape
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\1[illegible]0:Pandas4FutureWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_ind2["InvoiceNo"] = df_ind2["InvoiceNo"].astype(int)
```

```
(19[illegible], 8)
```

```
# Filter out price
```

```
neg_pr_mask_ind2 = df_ind2["UnitPrice"]<0[illegible]
print("Number of bad data (negative price) [illegible]
df_ind2[df_ind2.[illegible]
df_ind2.shape
```

```
Negative and bad data (negative price) [illegible]
(19[illegible], 8)
```

```
# Some invoiceNos entries are not actual products or [illegible]
```

```
df_ind2["StockCode"] = df_ind2["StockCode"].[illegible]
neg_pr_mask=df_ind2[df_ind2["StockCode"].[illegible]
print("Number of bad data (negative price) [illegible]
df_ind2[df_ind2.[illegible]
df_ind2.shape
```

```
Number of bad data (negative price) [illegible]
(19[illegible], 8)
```

```
df_ind2["InvoiceNo"] = df_ind2["InvoiceNo"].astype(str)
df_ind2 = df_ind2[df_ind2["InvoiceNo"].str.startswith("C")]
df_ind2.shape
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\1[illegible]0:Pandas4FutureWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_ind2["InvoiceNo"] = df_ind2["InvoiceNo"].astype(int)
```

```
(19[illegible], 8)
```

Feature Engineering

```
df_ind2["Description"].[illegible]
df_ind2.head()
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\1[illegible]0:FutureWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	TotalPrice
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	15.30
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	22.00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34

```
# Engineering new features
```

```
df_ind2["Year"]=df_ind2["InvoiceDate"].dt.year
df_ind2["Month"]=df_ind2["InvoiceDate"].dt.month[illegible]
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\1[illegible]0:FutureWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_ind2["Year"]=df_ind2["Year"].astype(int)
df_ind2.[illegible]
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\1[illegible]0:FutureWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_ind2.[illegible]
```

	Description	Quantity	InvoiceDate	UnitPrice	InvoiceNo	TotalPrice	DayofWeek	Month	Year	Country
2315	PAPER CHAIN - LITTLE BIRDIE	80995	2.08000	1	16446.00	4	12	United Kingdom	2010
1968	MEDIUM CERAMIC TOP STORAGE JAR	74615	1.22000	195	81416.70	0	8	United Kingdom	2011
4016	MANUAL WRIT G LEADED ASTER DESIGN	54930	0.29000	472	13935.41	2	12	United Kingdom	2010
1758	JUMBO BAG RED RETROSPOT	40270	2.01889	1600	85040.04	3	11	United Kingdom	2011
3896	WHITE HANGING HEART T-LIGHT HOLDER	36726	2.85750	1571	102052.12	3	12	United Kingdom	2011

```
df_ind2.groupby('Month').[illegible]
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\2[illegible]0:FutureWarning:
Passing 'pilotte' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend' = False for the same effect.
```

```
sns.barplot(data=[illegible]
```

```
top_countries = df_ind2.groupby("Country")["TotalPrice"].sum().sort_values(ascending=False).head(10)
```

```
# Create the bar plot
plt.figure(figsize=(12,6))
sns.barplot(x=top_countries.index, y=top_countries.values, palette="Set1")
plt.xlabel('Country')
plt.ylabel('Total Revenue (GBP)')
plt.title('Top 10 Revenue generated Products')
plt.xticks(rotation=45)
plt.show(top_countries)
plt.grid(axis='y')
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\2[illegible]0:FutureWarning:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend' = False for the same effect.
```

```
sns.barplot(x=top_countries.index, y=top_countries.values, palette="Set1")
```

```
df_ind2.[illegible]
```

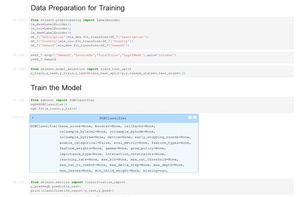
```
top_monthlies_sales = df_ind2.groupby("Month_Year")["TotalPrice"].sum().reset_index()
```

```
# Create the bar plot
plt.figure(figsize=(12,6))
sns.barplot(data=top_monthlies_sales, x="Month_Year", y="TotalPrice",[illegible]
plt.xlabel('Monthly Sales')
plt.ylabel('Month_Year')
plt.title('Total Revenue')
plt.xticks(rotation=45)
plt.tight_layout()
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\2[illegible]0:FutureWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_ind2["Month_Year"] = df_ind2["Month_Year"].astype(str)
```




```
df_ind2.[illegible]
```

```
top_monthly_sales = df_ind2.groupby("Month_Year")["TotalPrice"].sum().reset_index()
```

```
# Create the bar plot
plt.figure(figsize=(12,6))
sns.barplot(data=top_monthly_sales,x="Month_Year", y="TotalPrice",[illegible]
plt.xlabel('Monthly Sales')
plt.ylabel('Month_Year')
plt.title('Total Revenue')
plt.xticks(rotation=45)
plt.tight_layout()
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\2[illegible]0:FutureWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_ind2.[illegible]
```

```
df_ind2.[illegible]
```

```
top_monthly_sales = df_ind2.groupby("Month_Year")["TotalPrice"].sum().reset_index()
```

```
# Create the bar plot
plt.figure(figsize=(12,6))
sns.barplot(data=top_monthly_sales,x="Month_Year", y="TotalPrice",[illegible]
plt.xlabel('Monthly Sales')
plt.ylabel('Month_Year')
plt.title('Total Revenue')
plt.xticks(rotation=45)
plt.tight_layout()
```

```
C:\Users\pwb\ AppData\Local\Temp\ipykernel_25292\2[illegible]0:FutureWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_ind2.[illegible]
```

```
df_ind2.[illegible]
```

```
[illegible]
```

```
[illegible]
```

Data Preparation for Training

```
from sklearn.preprocessing import LabelEncoder
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

```
[illegible]
```

```
[illegible]
```

```
[illegible]
```

Train the Model

```
from sklearn import [illegible]
[illegible]
[illegible]
```

```
RandomClassifier
RandomClassifier(bootstrap=True, ccp_alpha=0.0,
                 class_weight=None, criterion='gini',
                 max_depth=None, max_features='sqrt',
                 max_leaf_nodes=None, max_samples=None,
                 min_impurity_decrease=0.0, min_impurity_split=None,
                 min_samples_leaf=1, min_samples_split=2,
                 min_weight_fraction_leaf=0.0, n_estimators=100,
                 n_jobs=None, oob_score=False, random_state=None,
                 verbose=0, warm_start=False)
```

```
from sklearn.metrics import [illegible]
[illegible]
```

```
[illegible]
```

```
[illegible]
```

```
comparison_df = pd.DataFrame(
    {'Product': X_test['Description'].values,
    'Predicted_Demand': y_pred,
    'Predicted_Demand': y_pred
    )
comparison_df['Product'] = dim_df.loc[comparison_df['Product'],
comparison_df['Predicted_Demand'] = inv_dem.loc[comparison_df['Predicted_Demand'],
comparison_df['Predicted_Demand'] = inv_dem.loc[comparison_df['Predicted_Demand']]
```

```
[illegible]
```

```
[illegible]
```

```
[illegible]
```