**Himanshu Kumar**

**Q1-** What is the % of Sales of Appliances in the Year 2012?

**Ans-** sd=Store\_data.drop(['CustID', 'PostalCode','City','Country','Market','ProdID'], axis=1)

sd.value\_counts('Status')

fin\_data= pd.get\_dummies(sd['Status'], drop\_first=False)

data=pd.concat([sd, fin\_data], axis=1)

data['Year'] = pd.DatetimeIndex(data['OrderDate']).year

data['date'] = pd.to\_datetime(data['OrderDate'])

data['quarter'] = data['date'].dt.to\_period('Q')

data['sales'] = data['SalePrice'] \* data["Quantity"]

year\_sales = pd.DataFrame(data.groupby("Year").sum()["sales"])

prod\_sales = pd.DataFrame(data.groupby("ProdSubCat").sum()["sales"])

year\_sales + prod\_sales.sum()

Appliances\_2012 = data[(data.ProdSubCat == 'Appliances') & (data.Year == 2012) & (data.Status == 'OK')]

A\_Sales = Appliances\_2012["sales"].sum()

t\_sales = data[(data.Year == 2012) & (data.Status == 'OK')]

sales\_t=t\_sales['sales'].sum()

print("% of App in total sales (in %)", (A\_Sales / sales\_t)\*100)

**Result - % of App in total sales (in %) 3.4059376468688876**

**Q2** - What is the % of Sales of “Home Office” in the Year 2011?

**Ans-** data.replace(['CustomerSegment', " ",""])

Home\_Office\_2011 = data[(data.CustomerSegment == 'Home Office') & (data.Year == 2011) & (data.Status == 'OK')]

ha\_Sales = Home\_Office\_2011["sales"].sum()

t\_sales11 = data[(data.Year == 2011) & (data.Status == 'OK')]

t\_sales11=t\_sales11['sales'].sum()

print("% of App in total sales (in %)", (ha\_Sales / t\_sales11)\*100)

**Result-% of App in total sales (in %) 25.28776832184422**

**Q3**- Find the Top 10 product names by sales in each State. Which product is highest in 3 States?

**Ans**- product\_sales = pd.DataFrame(data.groupby("ProdSubCat").sum()["Quantity"])

product\_sales\_state= pd.DataFrame(data.groupby("State").sum()["sales"])

product\_sales = product\_sales.sort\_values('Quantity', ascending=False)

product\_sales[:10]



product\_sales\_state= pd.DataFrame(data.groupby("State").sum()["sales"])

product\_sales\_state= product\_sales\_state.sort\_values('sales', ascending=False)

product\_sales\_state[:3]



**Q4** Find the State with Lowest overall profit?

**Ans-** data["total\_cost"] = data["CostPrice"] + data["ShipCost"]

data["Discount\_amount"] = data["SalePrice"] \* data["Discount"]

data["profit\_per\_unit"] = data["SalePrice"] - data["total\_cost"] - data["Discount\_amount"].round(2)

data["total\_profit"] = data["profit\_per\_unit"] \* data["Quantity"]

profit\_by\_state = pd.DataFrame(data.groupby("State").sum("total\_profit").round(2))

profit\_by\_state = profit\_by\_state.sort\_values("total\_profit", ascending=True )

Low\_prof\_State = profit\_by\_state[:3]

sns.lineplot(x= "State", y="total\_profit", data=Low\_prof\_State)



**Q5 -** Show FinYear wise strategic data with Months (Apr to Dec) in Rows and FinYear as columns?

**Ans-** data['month'] = pd.DatetimeIndex(data['OrderDate']).month

FinYear = data[['sales' , 'Year' ]]

FinYear.groupby('Year').sum()

fin\_mon = data[["month", "Year", "sales"]]

Final\_data\_month = fin\_mon.pivot\_table(index = "month", columns="Year", aggfunc=sum, dropna=False)

Final\_data\_month = Final\_data\_month.drop([1.0, 2.0,3.0])

Final\_data\_month

****

**Q6** - Create a Geo Area Map showing a strategic measure per geo location ?

## Ans- Step 1: Connect to your geographic data

## Step 2: Join your data

## Step 3: Format your geographic data in Tableau

## Step 4: Create a geographic hierarchy

## Step 5: Build a basic map

## Step 6: Change from points to polygons

## Step 7: Add visual detail

## Step 8: Customize your background map

## Step 9: Create custom territories

## Step 10: Create a dual axis map

## Step 11: Customize how others can interact with your map

**Q7** -Show all Invoice data with OrderDate wise Totals?

**Ans-**

**.**

**.**

**.**

**Q8-** Does Random forest need pruning? Why or why not ?

**Ans-**

**.**

**.**

**.**

**Q9** -Write a Python program using Scikit-learn to split the iris dataset into 80% train data and 20% test data. Out of total 150 records, the training set will contain 120 records and the test set contains 30 of those records. Train or fit the data into the model and calculate the accuracy of the model using the Random Forest Classifier Algorithm.

**Ans-** from sklearn import datasets

#Load dataset

iris = datasets.load\_iris()

# Creating a DataFrame of given iris dataset.

import pandas as pd

data=pd.DataFrame({

'sepal length':iris.data[:,0],

'sepal width':iris.data[:,1],

'petal length':iris.data[:,2],

'petal width':iris.data[:,3],

'species':iris.target

})

data.head()

# Import train\_test\_split function

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

X=data[['sepal length', 'sepal width', 'petal length', 'petal width']] # Features

y=data['species'] # Labels

# Split dataset into training set and test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2) # 80% training and 20% test

#Import Random Forest Model

from sklearn.ensemble import RandomForestClassifier

#Create a Gaussian Classifier

clf=RandomForestClassifier(n\_estimators=100)

#Train the model using the training sets y\_pred=clf.predict(X\_test)

clf.fit(X\_train,y\_train)

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

#Import scikit-learn metrics module for accuracy calculation

from sklearn import metrics

# Model Accuracy, how often is the classifier correct?

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))