**MARKET BASKET INSIGHT**

**PHASE 3-DEVELOPMENT PART 1**

**TOPIC- LOADING AND PREPROCESSING THE DATASET**

**A.DINESH KUMAR (511721205002)**



* **INTRODUCTION**

***->****Now days marketing are very important in our era. So before going to deal with the data sets we need preprocess the datasets.*

***->****This introduction will guide you through the initial steps of the process. We'll explore how to import essential libraries, load the tweets dataset, and perform critical preprocessing steps. Data preprocessing is crucial as it helps clean, format, and prepare the data for further analysis. This includes handling missing values, encoding categorical variables, and ensuring that the data is appropriately scaled.*

* **LINK FOR CSV(EXCEL) FILE DATASET WITH THE SENTIMENTS**
* [**https://1drv.ms/u/s!Aq87zB23WYAigVpPHJsYkwuIojhw**](https://1drv.ms/u/s!Aq87zB23WYAigVpPHJsYkwuIojhw)

**CUSTOMER BASKETS WITH PURCHASED ITEMS**

****

* **SOURCE CODE TO INPUT THE DATASET FOR ANALYSIS**

Import necessary libraries

import pandas as pd

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

# Load your dataset into a DataFrame

# Replace 'your\_dataset.csv' with your actual dataset file

df = pd.read\_csv('your\_dataset.csv')

# Data preprocessing: you may need to adjust this based on your dataset

# For example, if your data is not in the transaction format, you'll need to transform it.

# Perform one-hot encoding to convert items into binary columns

basket = pd.get\_dummies(df, columns=['item\_column\_name'])

# Group the data by a unique identifier (e.g., order ID) and sum the item counts

basket = basket.groupby('unique\_identifier').sum()

# Apply Apriori algorithm to find frequent item sets

frequent\_itemsets = apriori(basket, min\_support=0.1, use\_colnames=True)

# Generate association rules

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1.0)

# You can now analyze the association rules and draw insights

print(rules)

**TOP 5 DATASETS FROM THE GIVEN**

****

* **SOURCE CODE TO PROCESS THE DATASET FOR ANALYSIS**

*import pandas as pd*

*data = pd.read\_csv("input/Tweets.csv")*

*data.columns = [col.replace(":", "\_") for col in data.columns]*

*columns\_to\_keep = ["tweet\_id"] + [col for col in data.columns if col[0]!="\_" and col!="tweet\_id"]*

*data = data[columns\_to\_keep]*

*conversion = {*

*"object": "TEXT",*

*"float64": "NUMERIC",*

*"int64": "INTEGER"*

*}*

*sql = """.separator ","*

*CREATE TABLE Tweets (*

*%s);*

*.import "working/noHeader/Tweets.csv" Tweets*

*""" % ",\n".join([" %s %s%s" % (key,*

*conversion[str(data.dtypes[key])],*

*" PRIMARY KEY" if key=="tweet\_id" else "")*

*for key in data.dtypes.keys()])*

*data.to\_csv("output/Tweets.csv", index=False)*

*open("working/import.sql", "w").write(sql)*

**NECESSARY STEPS TO FOLLOW:-**

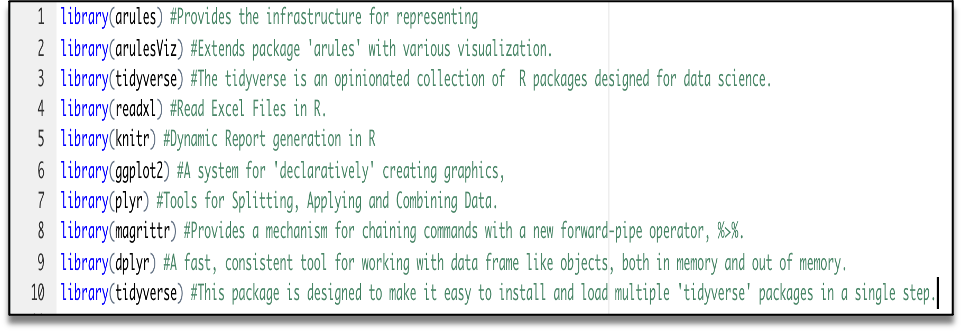
1. **IMPORT LIBRARIES:**

*Start by importing the necessary libraries:*

PROGRAM

*import pandas as pd import numpy as np*

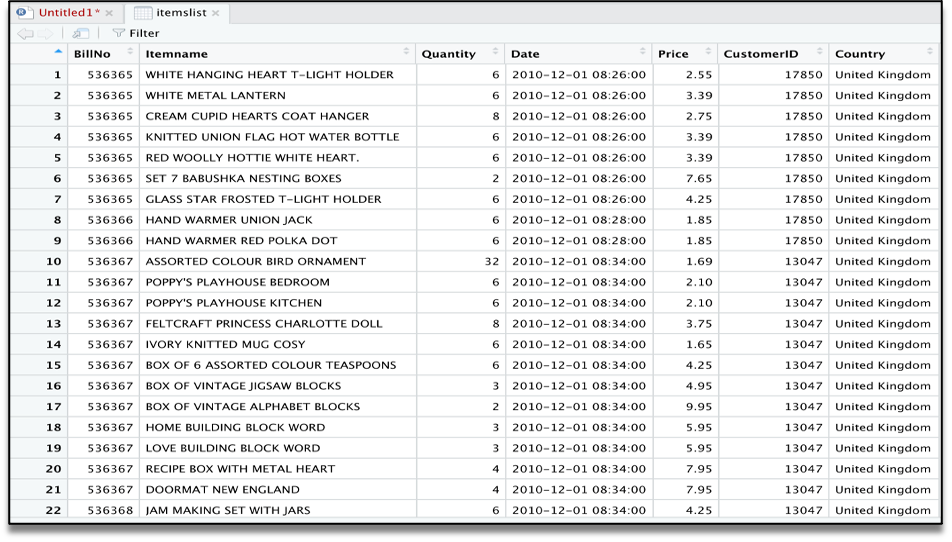
*from sklearn.model\_selection import train\_test\_split from sklearn.preprocessing import StandardScaler*

**

**2.LOAD THE DATASET**

*Load your dataset into a Pandas DataFrame. You can typically find house price datasets in CSV format, but you can adapt this code to other formats as needed.*

*tweets=pd.read\_csv(‘tweetscsv.csv’)*

**

**3.EXPLORATORY ANALYSIS**

*To begin this exploratory analysis, first use matplotlib to import libraries and define functions for plotting the data. Depending on the data, not all plots will be made. (Hey, I'm just a kerneling bot, not a Kaggle Competitions Grandmaster!)*

**PROGRAM:**

*from mpl\_toolkits.mplot3d import**Axes3D*

*fromsklearn.preprocessingimport**StandardScaler*

*importmatplotlib.pyplotas**plt # plotting*

*importnumpyas**np # linear algebra*

*import**os # accessing directory structure*

*import pandas as**pd # data processing*

*pd.read()*

**4.FEATURE ENGINEERING:**

*Depending on your dataset, you may need to create new features or transform existing ones. This can involve one-hot encoding categorical variables, handling date/time data, or scaling numerical features.*

**PROGRAM:**

*# Example: One-hot encoding for categorical variables*

*df = pd.get\_dummies(df, columns=[' Avg. Area Income ', ' Avg. Area House Age '])*

**5. SPILT THE DATA:**

*Split your dataset into training and testing sets. This helps you evaluate your model's performance later. X = df.drop('price', axis=1) # Features y = df['price'] # Target variable*

*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)*

**6.FEATURE SCALING:**

*Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 and std=1) is a common choice.*

**PROGRAM:**

*scaler = StandardScaler()*

*X\_train = scaler.fit\_transform(X\_train)*

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

**IMPORTANCE OF LOADING AND PREPROCESSING DATA:**

*Loading and preprocessing the dataset is an important first step in building any machine learning model. However, it is especially important for house price prediction models, as house price datasets are often complex and noisy.*

*By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.*

**CHALLENGES INVOLVED**

*There are a number of challenges involved in loading and preprocessing a house price dataset, including*:

* **HANDLING MISSING VALUES**

*House price datasets often contain missing values, which can be due to a variety of factors, such as human error or incomplete data collection. Common methods for handling missing values include dropping the rows with missing values, imputing the missing values with the mean or median of the feature, or using a more sophisticated method such as multiple imputation.*

* **ENCODING CATEGORICAL VARIABLES**

*House price datasets often contain categorical features, such as the type of house, the neighborhood, and the school district. These features need to be encoded before they can be used by machine learning models. One common way to encode categorical variables is to use one-hot encoding.*

* **SCALING THE FEATURES:**

*It is often helpful to scale the features before training a machine learning model. This can help to improve the performance of the model and make it more robust to outliers. There are a variety of ways to scale the features, such as min-max scaling and standard scaling.*

* **SPLITTING THE DATASET INTO TRAINING AND TESTING SETS:**

*Once the data has been pre-processed, we need to split the dataset into training and testing sets. The training set will be used to train the model, and the testing set will be used to evaluate the performance of the model on unseen data. It is important to split the dataset in a way that is representative of the real world distribution of the data.*

**HOW TO OVERCOME THE CHALLENGES**

*There are a number of things that can be done to overcome the challenges of loading and preprocessing a house price dataset, including:*

* **USE A DATA PREPROCESSING LIBRARY**

*There are a number of libraries available that can help with data preprocessing tasks, such as handling missing values, encoding categorical variables, and scaling the features.*

* **CAREFULLY CONSIDER THE NEEDS FOR YOUR DATA**

*The best way to preprocess the data will depend on the specific machine learning algorithm that you are using. It is important to carefully consider the requirements of the algorithm and to preprocess the data in a way that is compatible with the algorithm.*

* **VALIDATE THE PREPROCESSED DATA:**

*It is important to validate the preprocessed data to ensure that it is in a format that can be used by the machine learning algorithm and that it is of high quality. This can be done by inspecting the data visually or by using statistical methods.*

**1.LOADING THE DATASET:**

* *Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.*
* *The specific steps involved in loading the dataset will vary depending on the machine learning library or framework that is being used. However, there are some general steps that are common to most machine learning frameworks:*

1. **IDENTIFY THE DATA SET:**

*The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.*

1. **LOAD THE DATASET:**

*Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.*

1. **PREPROCESS THE DATA SET :**

*Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and test sets.*

***PROGRAM***

*Import pandas as pd*

*Import numpy as np*

*Import seaborn as sns*

*import matplotlib.pyplot as plt from sklearn.model\_selection import train\_test\_split from sklearn.preprocessing import StandardScaler*

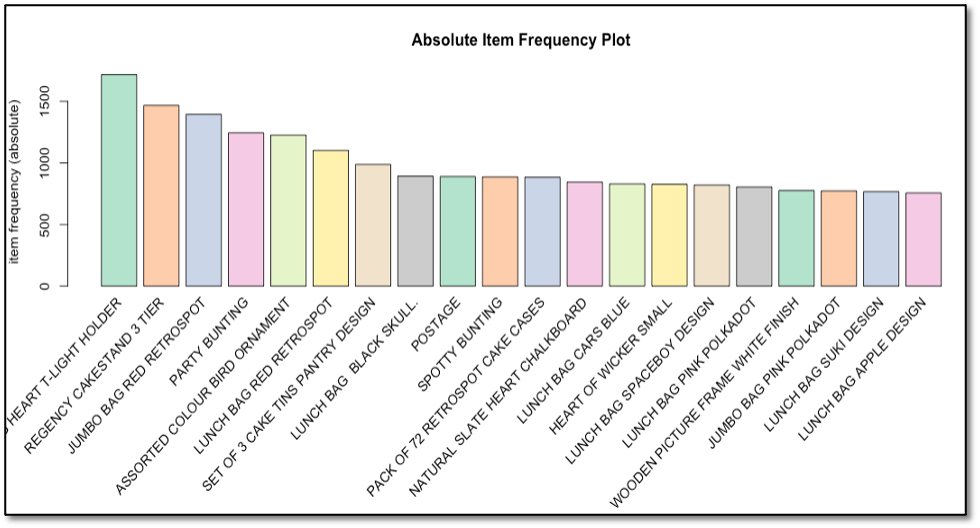
*from sklearn.svm import SVR*

*import xgboost as xg*

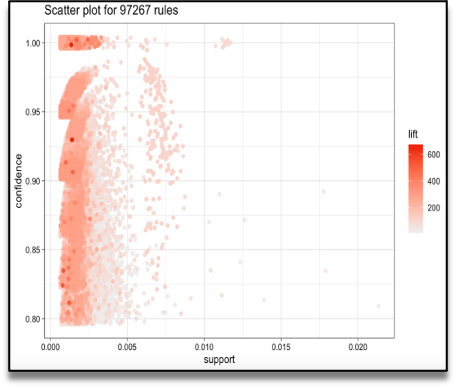
*warnings.filterwarnings(‘ignore’)*

*from sklearn.metrics import r2\_score, mean\_absolute\_error,mean\_squared\_error from sklearn.linear\_model import LinearRegression from sklearn.linear\_model import Lasso from sklearn.ensemble import RandomForestRegressor*

**BAR CHAT:**

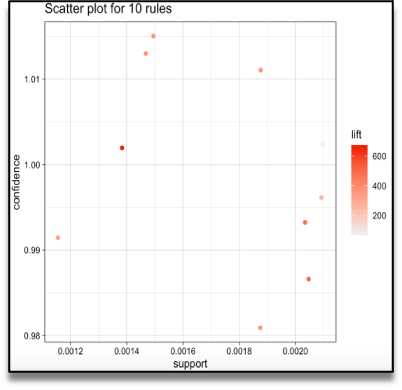
**

**SCATTER PLOT:**

**

#### Graph - Based Visualization and Group Method:

*Graph plots are a great way to visualize rules but tend to become congested as the number of rules increases. So, it is better to visualize a smaller number of rules with graph-based visualizations. We can see as well group method for top 10 items*.

**

**CONCLUSION:**

*. In the quest to build a house price prediction model, we have embarked on a critical journey that begins with loading and preprocessing the dataset. We have traversed through essential steps, starting with importing the necessary libraries to facilitate data manipulation and analysis*

*Withthese foundational steps completed, our dataset is now primed for the subsequent stages of building and training a basket market insight.*