CAPSTRONE PROJECT

Introduction

Amazon is one of the world's largest e-commerce and technology companies, founded in 1994 by Jeff Bezos. It started as an online bookstore and has since expanded to sell a wide variety of products and services, including electronics, home goods, apparel, and more. In addition to retail, Amazon offers a number of other services, such as cloud computing through Amazon Web Services and streaming of video, music, and audiobooks through Amazon Prime. The company also manufactures and sells its own consumer electronics, including the Kindle e-reader and the Echo smart speaker. Amazon operates globally with its headquarters in Seattle, Washington.

Files Used:

1. Review Data - Video\_Games\_5
2. Rating Data - Video\_Games
3. Meta Data - meta\_Video\_Games

Objective:

Thomas, a global market analyst, wishes to develop an automated system to analyze and monitor an enormous number of reviews. By monitoring the entire review history, he will analyze tone, language, keywords, and trends over time to provide valuable insights that increase the success rate of existing and new products and marketing campaigns.

Scenarios:

Scenario 1: Inventory Optimization and Demand Forecasting Optimize inventory management by identifying the product categories (clustering as an outcome of text processing) on the customer review data. Predict what kind of products could be in demand (Time Series Analysis).

Scenario 2: Customer Retention and Sentiment Forecasting Customer retention strategy through feedback analysis (customer classification and clustering as an outcome of analyzing the review text). Trend and seasonality analysis to predict how frequently a particular category of customer would shop in the future. (Time Series Analysis) Time Series component: Trend, Seasonality Analysis to predict how frequently this the customer would buy new products.

Method:

**Libaries used:**

import pandas as pd

import json

import gzip

import matplotlib.pyplot as plt

import seaborn as sns

import re

import numpy as np

import warnings

warnings.filterwarnings("ignore")

import nltk

from nltk.corpus import stopwords

stop\_words = stopwords.words("english")

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

from sklearn.feature\_extraction.text import TfidfVectorizer,CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import recall\_score,f1\_score,accuracy\_score,precision\_score,confusion\_matrix,classification\_report,silhouette\_score

sns.set\_theme(style="darkgrid")

sns.set(rc={'figure.facecolor':'lightblue'})

from sklearn.pipeline import Pipeline,make\_pipeline

from sklearn.linear\_model import LogisticRegression

from sklearn.multiclass import OneVsRestClassifier

import fasttext

from time import time

from sklearn.cluster import KMeans,DBSCAN,AgglomerativeClustering

from sklearn.preprocessing import MinMaxScaler,StandardScaler

import scipy.cluster.hierarchy as sch

from sklearn.neighbors import NearestNeighbors

from datetime import datetime

from sklearn.metrics.pairwise import cosine\_similarity

from statsmodels.tsa.seasonal import seasonal\_decompose

from statsmodels.tsa.stattools import adfuller

from statsmodels.graphics.tsaplots import plot\_pacf,plot\_acf

from statsmodels.tsa.arima.model import ARIMA,sarimax

from sklearn.metrics import mean\_squared\_error

from pmdarima.arima import auto\_arima

import statsmodels.api as sm

from tqdm.notebook import tqdm

from nltk.sentiment import SentimentIntensityAnalyzer

from langdetect import detect

Pre-Processing Steps:

**Data Importation:**

All three data files are downloaded and imported for data exploration and cleaning process.

**Data Exploration:**

Once the data is imported, it's important to inspect the data to get a sense of its structure and quality. Same steps are followed for all three data files.

1. Data shape is found using df.shape
2. First and last 5 rows are inspected – df.head, df.tail
3. Statistical values of the numerical data - dfdescribe
4. Datatype of each column in the dataframe - df.info
5. Unnecessary columns were dropped from the dataframe.
6. Review Data - ["unixReviewTime","image","reviewerName"]
7. Meta Data – Only the following columns are used ["category","title","brand","main\_cat","asin","rank"]
8. Columns were renamed for rating data – ['item','user','rating','timestamp']

**Handling missing data**:

Missing data can be a problem when analysing data. There are several ways to handle missing data, such as removing rows with missing data, imputing missing values, or using a combination of both.

Missing data for review and rating data are found using df.isnull().sum()

As for Meta data, a different approach is used since different columns had lists,strings as values.

counts = m.apply(lambda x: x.apply(lambda y: 1 if y == [] or y=='' or y=={} else 0).sum())

By printing the ‘counts’, all empty lists, dictionaries and strings were found.

Since, the missing data from several columns were higher in number, those columns were removed from the dataframes.

Removed columns based on null values:

1. Review Data – [‘style’, ‘vote’]
2. Meta Data - Only the following columns are used ["category","title","brand","main\_cat","asin","rank"]
3. Rating Data – No columns has been removed.

**Data Cleaning:**

This step includes tasks such as removing duplicates, correcting errors, and standardizing data. For example, you can check for outliers, remove or correct invalid data, and convert data into a consistent format.

Following steps are performed to clean the Review data.

1. Dropping duplicate rows - df[df.duplicated(keep=False)]
2. For cleaning textual data in the ‘reviewText’ column, regular expressions were used.
3. The following cleaning function is used to clean the data.

def cleaning(text):

text = re.sub("[^0-9A-Za-z\-]+", " ", text)

text = re.sub("(?<!\w)\d+", "", text)

text = re.sub("-(?!\w)", "", text)

text = " ".join(text.split())

text = text.lower()

return text

1. Stopwords were also removed from the text column.

df['reviewText'] = df['reviewText'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop\_words)]))

Following steps are performed to clean the Meta data.

1. Since we need only Video Games data, other categories in ‘main cat’ column were removed.
2. Duplicated rows were removed from the dataframe.

**Feature extraction:**

The goal of feature extraction is to identify and extract the most relevant and informative features from the raw data, while discarding irrelevant or redundant information. It is performed on the main data since category column was in list which contained important features.

1. While exploring, three main categories were found inside lists in the ‘category’ columns. Extracting these categories would help in EDA for answering business questions. Following functions were used to extract these categories and assign these values in a new column.

a=m[m["category"].apply(lambda x:"Games" in x)]

b=m[m["category"].apply(lambda x:"Accessories" in x)]

c=m[m["category"].apply(lambda x:"Consoles" in x)]

a["category"]="Games"

b["category"]="Accessories"

c["category"]="Consoles"

m1=pd.concat([a,b,c])

1. Platform information was also available in the lists, so another function is used to extract these data. Research

l1=["PlayStation 3","PlayStation 4","PlayStation 2","Sony PSP","PlayStation","PlayStation Vita"]

l2=["Xbox","Xbox 360","Xbox One"]

l3=["Super Nintendo","Nintendo DS","Sega Genesis","Wii","Game Boy Advance","GameCube","Nintendo 64","Game Boy Color",

"Sega Dreamcast","NEOGEO Pocket","Sega Game Gear","Nintendo NES","Nintendo Switch","Nintendo 3DS & 2DS","Wii U"]

l=l1+l2+l3+["PC","Mac","3DO"]

To add all the dataframes into a single dataframe.

for i in l:

a=m[m["category"].apply(lambda x:i in x)]

a["platform"]=i

n=pd.concat([n,a])

**Save the clean data**:

It's important to save the cleaned data so that it can be easily reused later. This can be done by exporting the data to a file, such as a CSV or Excel file, or by pickling the data for use in Python.Both the cleaned files were saved using “.to\_csv” method.

**Merging Review and Meta data:**

Both Review and Meta data are merged using ‘pd.merge’. Duplicated and null values are removed from the merged dataframe. This merging allows us to perform extensive EDA to determine some business answers.

Preliminary Analysis:

**Sentiment Analysis:**

Sentiment analysis is the process of determining the emotional tone of a piece of text. It is a natural language processing task that can be used to classify text as positive, negative, or neutral. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media.

**Objective:**

The objective is to build a sentiment analysis model using VADER that can accurately classify reviews as positive, negative, or neutral sentiments.

Step 1: Installation

First, you need to install the VADER library. You can do this by running the following command:

Step 2: Importing the library and loading the data

Next, you need to import the VADER library and load your Amazon reviews data. The following code imports the library and loads a CSV file containing the reviews.

Step 3: Initializing the Sentiment Analyzer

Now, you need to initialize the Sentiment Analyzer by creating an instance of the SentimentIntensityAnalyzer class:

Step 4: Analyzing the Sentiment

Now, you can use the polarity\_scores method of the SentimentIntensityAnalyzer class to analyze the sentiment of each review. The method returns a dictionary containing the compound score, positive, negative and neutral scores.

Step 5: Extracting the Sentiment

You can extract the compound score, which ranges from -1 to 1, and represents the overall sentiment of the review. A score of 1 represents a positive sentiment, a score of -1 represents a negative sentiment and a score of 0 represents a neutral sentiment.

**Future Sentiment Forecasting:**

Future positive sentiments have been forecasted using Time Series Analysis. Below are the steps that are followed to achieve the future forecasts.

Step 1 : Resampling

Resampling of date column from days into months.

Step 2 : Stationarity check

Time series data is said to be stationary if its mean and variance remain constant over time. If the data is not stationary, it will be necessary to perform a transformation such as differencing or detrending to make it stationary.

Result : Data is stationary, so differencing is not required.

Step 3: Model Selection

Once the data is cleaned and understood, it is time to select a model to make predictions. There are several types of models commonly used for time series analysis, including moving averages, exponential smoothing, and ARIMA models.

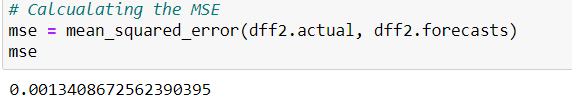
Result : Since data shows seasonality, SARIMA model is used.

Step 4 :Model Fitting

After selecting the model, it isto the data by estimating the model parameters. This step involves adjusting the parameters of the model to optimize its performance.

Step 5:Model Evaluation

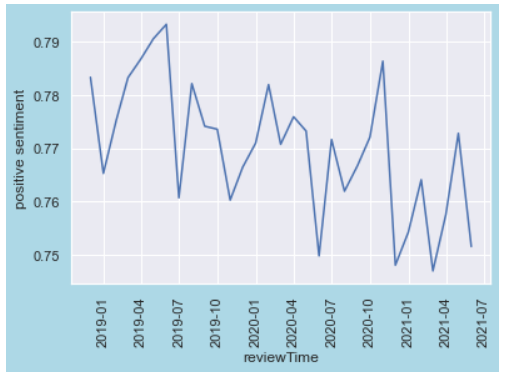
Once the model is fitted, it is evaluated to determine its accuracy and usefulness. This is done by comparing the model's predictions to the actual data and calculating metrics such as mean squared error or root mean squared error.

Result : 

Step 6: Forecasting

After the model has been evaluated and found to be useful, it can be used to make predictions about future data. These predictions can be used to make decisions, identify patterns, or simply understand how the data is likely to change over time.

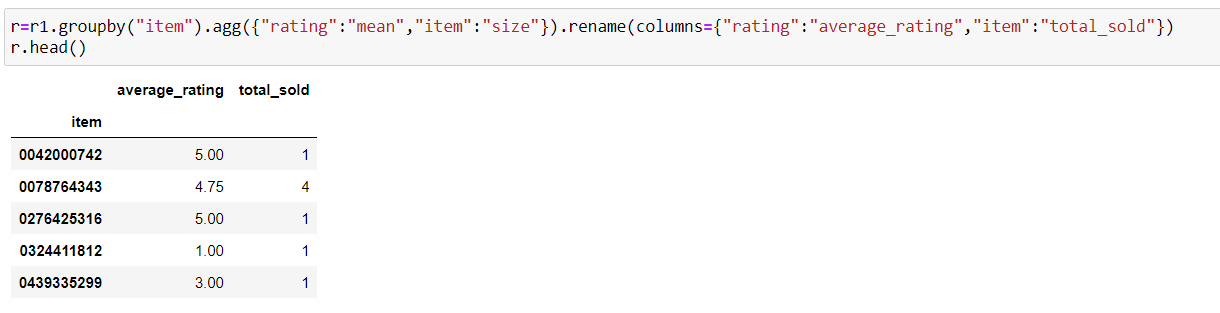
We can see the future forecasting for 10 weeks in the below graph.



**Product Segmentation:**

Product segmentation is the process of dividing a market into smaller groups of consumers with similar needs or characteristics. This allows companies to create targeted marketing campaigns and to develop products and services that meet the specific needs of each segment. It is a crucial step in developing an effective marketing strategy, as it allows companies to identify and focus on the most profitable segments of the market.

It is done with the rating data by taking the item column and grouping it to find the average rating and number of items rated for each product.



Step 1: Collect and preprocess data:

Data has been collected and preprocessed, to ensure that it is in a format that can be analyzed.

Step 2: Choose a clustering algorithm:

Select an appropriate clustering algorithm based on the characteristics of the data and the goals of the segmentation analysis. K-Means and Agglomerative Heirarchical clustering has been chosen to find which method best suits the data.

Step 3: Apply the algorithm:

1. K-Means Clustering
2. Scaling used – StandardScaler()
3. Elbow method is used to find the best number of clusters. (3 in our data)
4. Silhouette score is found for the model for evaluation metrics. (0.56 for this data)
5. Agglomerative Heirarchical Clustering
6. Scaling used – StandardScaler()
7. Dendrogram is used to find the best number of clusters. (3 in our data)
8. Silhouette score is found for the model for evaluation metrics.. (0.43for this data)

Step 4: Evaluate the results:

Evaluate the results of the clustering analysis to see if the segments make sense and are internally consistent.

Result: Based on the Silhouette score, K-Means algorithm is chosen for better clustering of products. Evaluation is also done and the clusters make sense with their rating and sales.

Step 5: Assign labels to segments:

Assign labels to each segment to make them easier to understand and interpret.

Cluster 0 – Not popular products – Low rating and low sales

Cluster 1 – Underrated products – High rating and low sales

Cluster 2 – Popular products – High rating and High sales

**Customer Segmentation:**

Customer segmentation is the process of dividing a customer base into smaller groups of individuals that have similar needs or characteristics. This allows companies to create targeted marketing campaigns and to develop products and services that meet the specific needs of each segment. Customer segmentation can be based on various factors such as demographics, behavior, and psychographics.

Behavioral segmentation divides customers based on their behavior and usage patterns, such as purchase history, brand loyalty, and benefits sought. Based on our data, customers are segmented based on their frequency, recency and satisfaction level.

Frequency, recency and satisfaction level are calculated with the help of the date and rating columns in the rating dataset. Same steps are followed as per product segmentation.

Step 1: Collect and preprocess data:

Data has been collected and preprocessed, to ensure that it is in a format that can be analyzed.

Step 2: Choose a clustering algorithm:

Select an appropriate clustering algorithm based on the characteristics of the data and the goals of the segmentation analysis. K-Means and Agglomerative Heirarchical clustering has been chosen to find which method best suits the data.

Step 3: Apply the algorithm:

1. K-Means Clustering
2. Scaling used – StandardScaler()
3. Elbow method is used to find the best number of clusters. (3 in our data)
4. Silhouette score is found for the model for evaluation metrics. (0.51 for this data)
5. Agglomerative Heirarchical Clustering
6. Scaling used – StandardScaler()
7. Dendrogram is used to find the best number of clusters. (3 in our data)
8. Silhouette score is found for the model for evaluation metrics.. (0.41for this data)

Step 4: Evaluate the results:

Evaluate the results of the clustering analysis to see if the segments make sense and are internally consistent.

Result: Based on the Silhouette score, K-Means algorithm is chosen for better clustering of products. Evaluation is also done and the clusters make sense with their satisfaction and frequency of ratings.

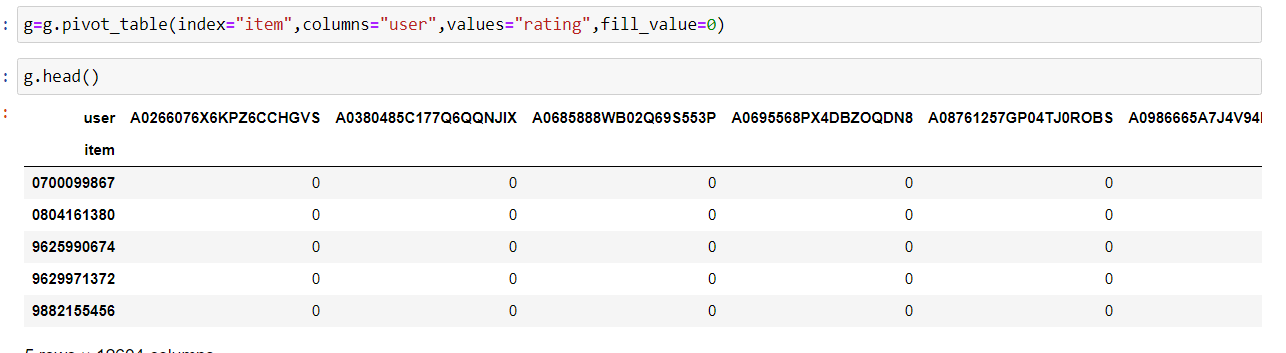
**Colloborative filtering:**

Collaborative filtering is a method of making recommendations to users based on their past behavior and the behavior of other users. It works by using the similarities between users or items to make predictions about the interests or preferences of a particular user. There are two main types of collaborative filtering: user-based and item-based.

**Item-based collaborative filtering:**

This method makes recommendations to a user based on the similarity between the items that the user has liked or interacted with and other items. It looks at the items that are most similar to the items that the user has liked and recommends those items to the user.

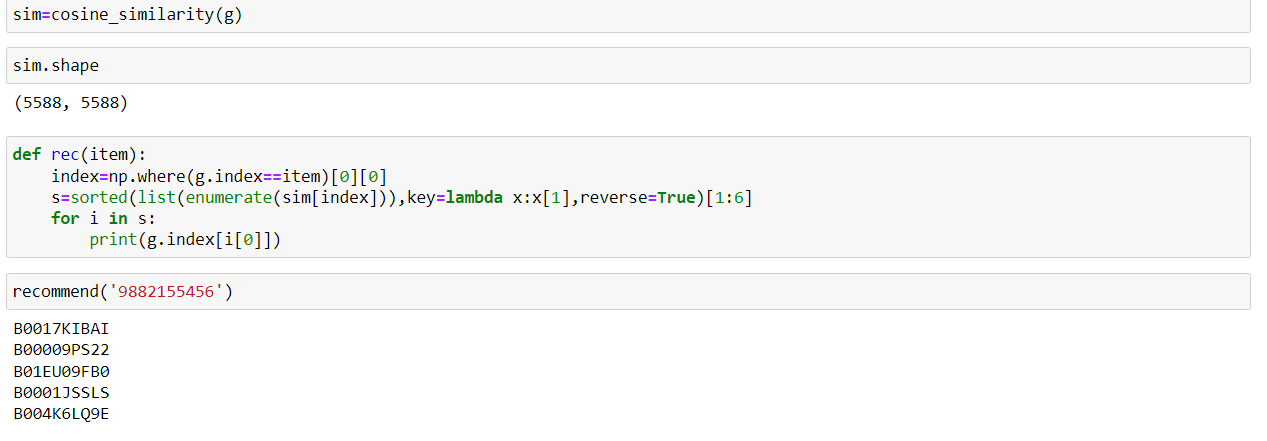
A cross table of the products along with their rating is created using the below code.



**Cosine similarity** is a measure used to determine the similarity between two non-zero vectors of an inner product space. It is commonly used in collaborative filtering to measure the similarity between users or items.

In item-based collaborative filtering, cosine similarity is used to measure the similarity between two items. The similarity score is calculated in a similar way to user-based collaborative filtering, by taking the dot product of the item vectors and dividing it by the product of the magnitudes of the vectors.

After using cosine similarity, we can find the top similar items to an item using simple code as below.



**Demand Forecasting :**

Demand forecasting is the process of estimating the future demand for a product or service. It is a crucial step in business planning and operations, as it allows companies to make informed decisions about production, inventory, and staffing. We are using Time series Analsyis to achieve the demand forecasting.

**Time-series analysis**: This method uses historical data on sales or demand to make predictions about future demand. It involves analyzing patterns in the data, such as trends, seasonality, and cyclical patterns, to make predictions about future demand. By following the below steps, we were able to forecast future video games demands.

Step 1 : Resampling

Resampling of date column from days into weeks.

Step 2: Visualisation of trend, seasonality, residuals :

This is achieved by decomposing the data using seasonal\_decompose() method and viewing the graphs using matplotlib library. This is done to check if any trends, seasonality or errors are present in the data.

Result: Seasonality is present in the data.

Step 3 : Stationarity check

Time series data is said to be stationary if its mean and variance remain constant over time. If the data is not stationary, it will be necessary to perform a transformation such as differencing or detrending to make it stationary. It is checked using adfuller() method.

Result : Data was not stationary, so differencing is required. After shifting once, data became stationary.

Step 4: Model Selection

Once the data is cleaned and understood, it is time to select a model to make predictions. There are several types of models commonly used for time series analysis, including moving averages, exponential smoothing, and ARIMA models.

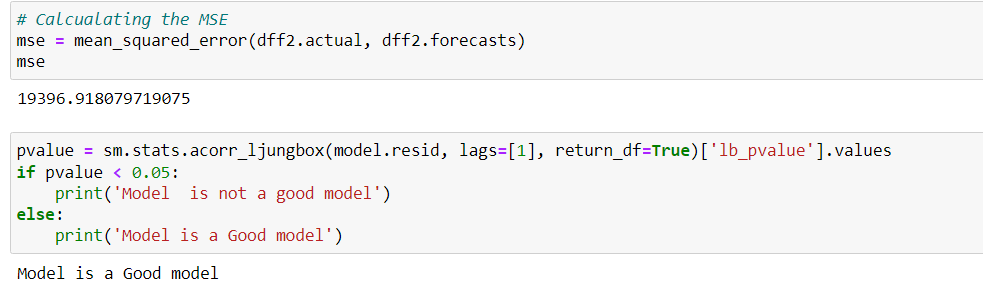
Result : Since data shows seasonality, SARIMA model is used. P and Q values are found with the help of ACF and PACF plots. (

Step 5 :Model Fitting

After selecting the model, it isto the data by estimating the model parameters. This step involves adjusting the parameters of the model to optimize its performance.

Step 6:Model Evaluation

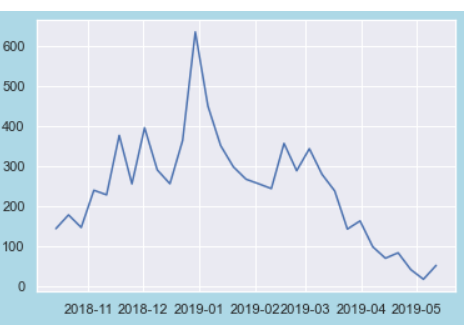
Once the model is fitted, it is evaluated to determine its accuracy and usefulness. This is done by comparing the model's predictions to the actual data and calculating metrics such as mean squared error or root mean squared error.

Result : By finding the RMSE and p\_value we can find that the model is a good model.

Step 6: Forecasting

After the model has been evaluated and found to be useful, it can be used to make predictions about future data. These predictions can be used to make decisions, identify patterns, or simply understand how the data is likely to change over time.

We can see the future forecasting for 10 weeks in the below graph.



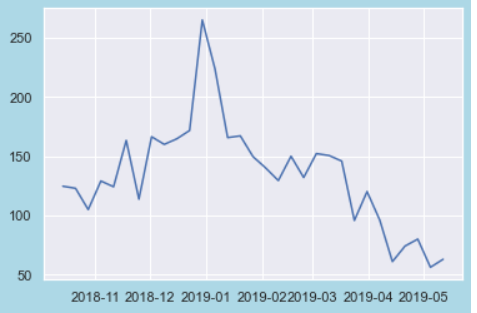
Same steps have been followed for all products belonging to different platforms, but all of them shows a similary trend, seasonality and future forecasts.

**Demand Forecasting on Underrated Products:**

Underrated products are products that have good reviews but could not perform well in sales. We have decided to analyse deeper on this and forecast to inform market strategy to boost the sales on these products.

Same steps have been followed as per demand forecasting on all the products. We can find the future forecast of the underrated products below.

**Future Forecast of 10 weeks for Underrated Products:**



**Conclusion:**

Some conclusions are found while doing the above analysis on the review data. Find some of the conclusion below.

* During the analysis, possibility of fake reviews by the reviewers should be kept in mind as about **35%** of the reviewer are non-verified customers.
* People tend of give mostly **Positive** or **5-star** reviews. This can mean that the products or services purchased are of good quality.
* Most of the customers tend to buy **Games** .
* Due change in customer behaviour, people are preferring digital copies from game distribution services.
* Number of Underrated Products are quite high.
* People tend to buys products near the holiday season.
* From forecasting we can expect increase in demand in future 10 weeks.

**BUSINESS PROPOSITIONS :**

Based on our analysis on the review data on Video Games category, we are hoping that the business propositions will be beneficial.

* Fake review monitoring system can be implemented.
* Lucrative deal or discounts can be promoted during the middle of the year to boosts sales.
* Focus should shift towards digital copies of the games instead of physical disk.
* Marketing campaigns maybe required for underrated products.
* Man power should be adjusted according to the demand forecast.
* Focus should be given on the existing customers

Appendix:

Working code book link:

<https://github.com/akshayaithal98/Amazon-Reviews-Analysis/blob/main/Capstone.ipynb>