Name – Himanshu Soni

Student ID – 29389089

Unit – FIT5141

Assignment 3

Table of Contents:

1. Introduction

E-commerce is one of the fastest growing industries in today's digital age amassing an estimated sales amount of $ 3.53 trillion which is projected to grow at $ 6.5 trillion in the next few years.

In 2018, the e-commerce industry had an estimated 1.8 billion active users which generates a massive amount of data with different formats such as numeric, textual, geospatial. As this market is extremely competitive, it is imperative that organizations utilize their data in making better business decisions.

For this assignment, I have decided to use the Olist public dataset published on Kaaggle. Olist is a Brazilian ecommerce giant and have released their data related to customers, product, sellers, orders, reviews and more.

Problem Statement:

In this scenario, I am a company executive who must make critical decisions in different spheres such as expansion of business as well as identify areas which require improvement. To accomplish this, I would use the insights from the data to make better decisions.

A detailed dataset explanation and their management is explained in the next section.

2.**Database Characteristics:**

The database in use is MongoDB and I am using it on my local machine (local host).

Some advantages that I found using MongoDB for this assignment were –

- It uses a NoSQL structure which eliminates the usage of multiple tables.
- Eliminates the concern regarding data integration.

Setup:

Problem statement – My task involves analyzing multiple files from the Brazilian e-commerce dataset with each file providing information about different aspects such as the customers, their orders, order details, all listed products and sellers.

1. To perform this, I created a single database called customer_records which can be accessed through the command

use customer_records.

```
> show databases
admin              0.000GB
config             0.000GB
customer_records   0.140GB
local              0.000GB
mydb               0.000GB
mydb1              0.000GB
> 
```

Collection information:

For this assignment I am using the following collections along with their local host names to answer the business questions which would be addressed in the Analysis part.

olist_customers_dataset.csv……cust_record

olist geolocation dataset.csv……geo records

olist order items dataset.csv……orders items

olist_order_payments_dataset.csv…….. payments

olist order reviews dataset.csv……reviews

olist orders dataset.csv……orders

olist products dataset.csv……products

product_category_name_translation.csv……products_category_english

The collections stored in mongoDB are:

For further details please refer this link –

https://www.kaggle.com/olistbr/brazilian-ecommerce

Data format issues such as date-time conversion and addition of fields will be discussed in the analysis part.

3. Analysis

Some of the business questions that I feel would be critical for our operations are.

1. The management has decided to expand our operation base in the next period. Is that a good decision?

2. How can we improve the satisfaction level of our customers? To answer this major question, we have to answer these subquestions

- What was the major causes of their dissatisfaction?

- Would it be beneficial to launch an 1 day delivery service like Amazon prime delivery?

- Which areas should be targeted (such as more stores, kiosks, customer interaction) more?

- Which product categories are performing well and which are performing poorly?

3.1 Forecasting analysis

To analyze if it's an ideal time for expansion, we should be able to forecast the sales for the next quarter as this help in making decision such as hiring, stock prices and other parameters.

To facilitate this, I have the following collections

1. Payments

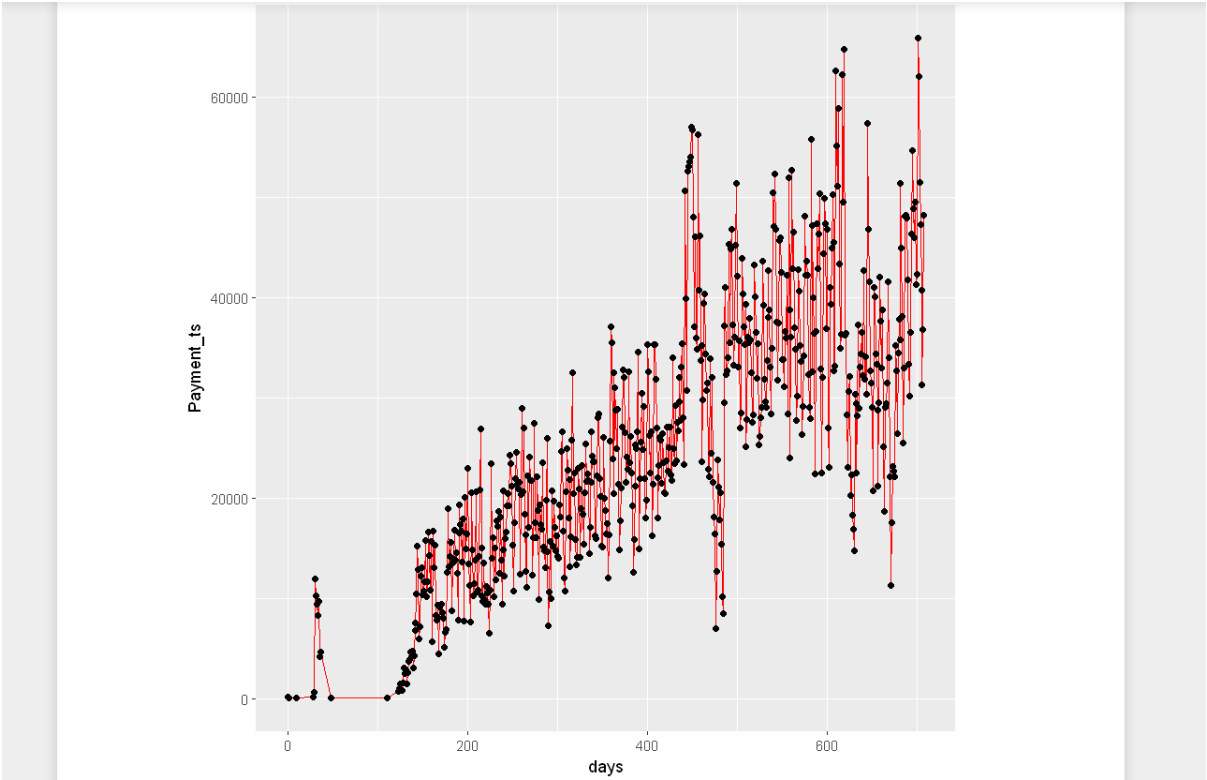
order_id	payment_sequential	payment_type	payment_installments	payment_value
ba78997921bbcdc1373bb41e913ab953	1	credit_card	8	107.78
b81ef226f3fe1789b1e8b2acac839d17	1	credit_card	8	99.33
42fdf880ba16b47b59251dd489d4441a	1	credit_card	2	128.45
298fcdf1f73eb413e4d26d01b25bc1cd	1	credit_card	2	96.12
771ee386b001f06208a7419e4fc1bbd7	1	credit_card	1	81.16

2. Orders

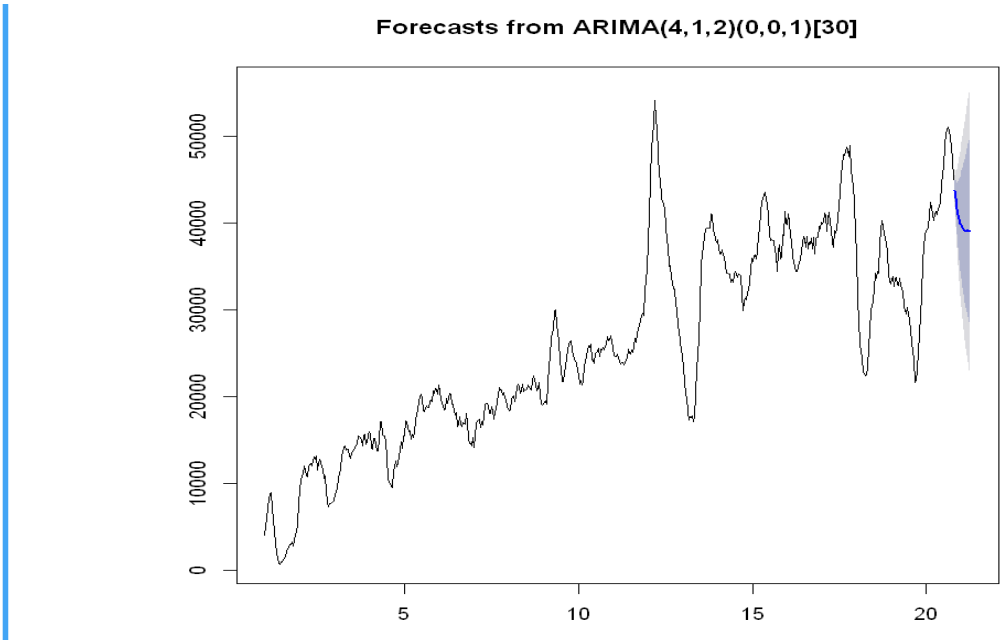
order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date
e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10-04 19:55:00
47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	2018-08-08 08:38:49	2018-08-08 08:55:23	2018-08-08 13:50:00
949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2017-11-18 19:28:06	2017-11-18 19:45:59	2017-11-22 13:39:59
a4591c265e18cb1dcee52889e2d8acc3	503740e9ca751ccdda7ba28e9ab8f608	delivered	2017-07-09 21:57:05	2017-07-09 22:10:13	2017-07-11 14:58:04
ad21c59c0840e6cb83a9ceb5573f8159	8ab97904e6daea8866dbdbc4fb7aad2c	delivered	2018-02-13 21:18:39	2018-02-13 22:20:29	2018-02-14 19:46:34

The two collections have are joined using the key order_id.

- From the initial analysis, it is evident that the sales(in terms of payments) has been increasing since the time of the launch (removed the outliers) as shown in the below figure.



The final analysis of this plot gives us an ARIMA model with the order 4,1,2 which can be used to forecast future sales. The plot is shown below



This plot includes the seasonality component and predicts the sales/revenue will be less than the previous sessions(indicated by the blue part).

3.2 Customer satisfaction and improvement analysis

To analyze the customer satisfaction levels, I am using the reviews collection which has the review scores ranging from 1(unhappy) to 5(happy).

The table count (discrete values) provide the following result.

```
    1     2     3     4     5
11858  3235  8287 19200 57420
```

This tells us that almost 12,000 people are unhappy with the service. The cause of this unhappiness and the measure taken to determine the appropriate response will be discussed further.

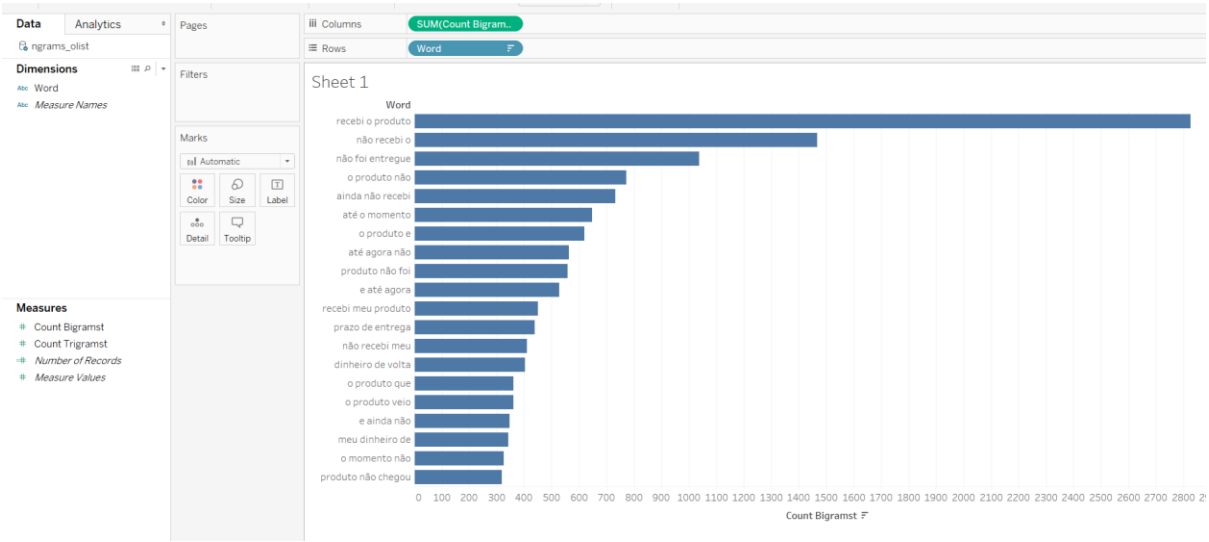
- Cause

The review_comment_message feature/column in the reviews collection has the comments posted by the users which will be used to understand the underlying cause of their dissatisfaction.

review_id	order_id	review_score	review_comment_title	review_comment_message	review_creation_date	review_answer_timestam
f80a40eba40	73fc7af87114b39712e6da79b0a377eb	4			2018-01-18 00:00:00	2018-01-18 21:46:5
39d5645fdfde	a548910a1c6147796b98fdf73dbeba33	5			2018-03-10 00:00:00	2018-03-11 03:05:1
l1322874b6f0	f9e4b658b201a9f2ecdecbb34bed034b	5			2018-02-17 00:00:00	2018-02-18 14:36:2
9ff8bb30750e	658677c97b385a9be170737859d3511b	5		Recebi bem antes do prazo estipulado.	2017-04-21 00:00:00	2017-04-21 22:02:0
c41a392bdeb	8e6bfb81e283fa7e4f11123a3fb894f1	5		Parabéns lojas lannister adorei comprar pela Internet seguro e prático Parabéns a todos feliz Páscoa	2018-03-01 00:00:00	2018-03-02 10:26:5

After subsetting the documents/rows which have a review score of 1, I was able to generate bigrams and trigrams from the corpus to understand the common concerns.

This diagram shows the 20 major causes of unhappiness



Using google translate, The Portuguese words were converted to English and the major causes were:

- Customers didn't receive the product(primary) – almost 80% of the cases
- Late delivery (few cases).

- Does delivery time affect the review scores?

1. To analyze this, I used the orders and the reviews datasets (mentioned earlier) and performed a join operation on the common key order_id

The order dataset has the following columns/features

order_purchase_timestamp - Shows the purchase timestamp.

order_approved_at - Shows the payment approval timestamp.

order_delivered_carrier_date - Shows the order posting timestamp. When it was handled to the logistic partner.

order_delivered_customer_date - Shows the actual order delivery date to the customer.

2. Based on this features, additional features were generated which are listed below:

time_to_approve – Time taken to approve

time_to_deliver – Time taken to deliver

diff_estimated_actual_delivery_date – Difference between the generated estimated time of delivery by Olist and the actual delivery date.

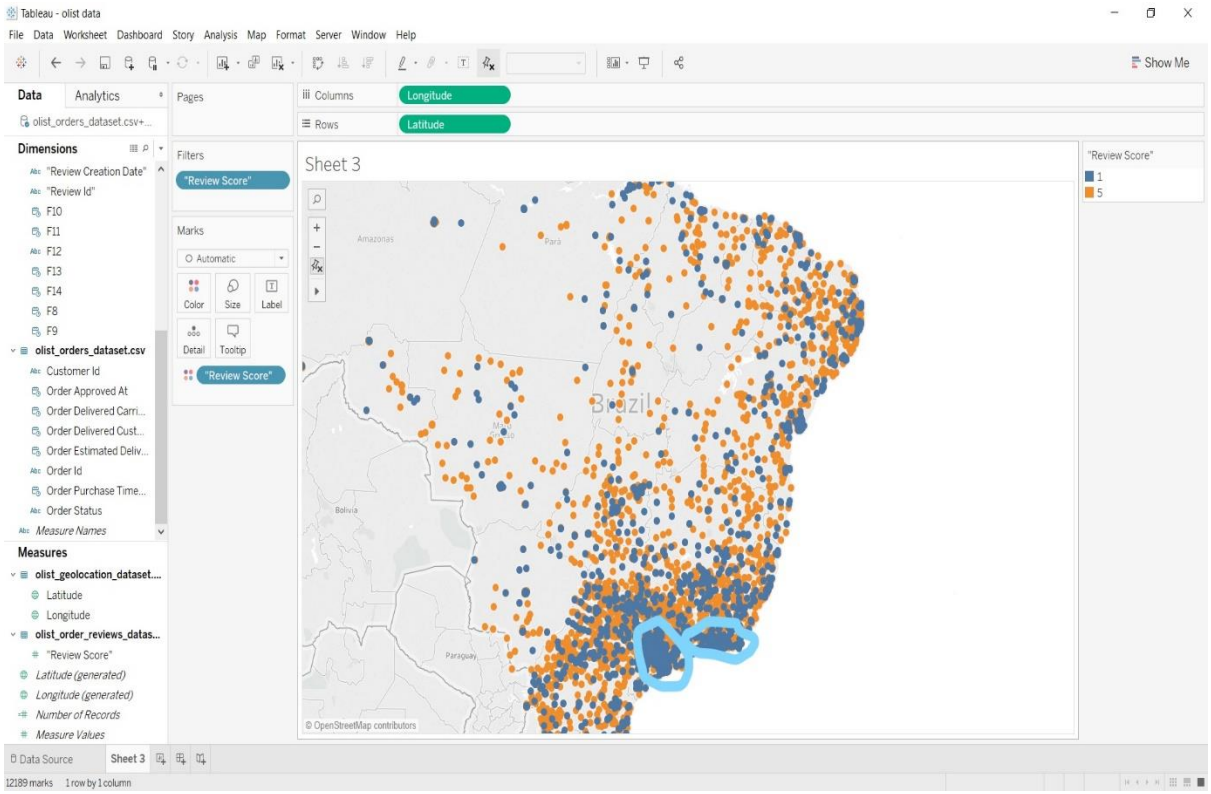
3. After using the correlation function between the new feature and review score (one at a time), the results indicate non-significant correlation below a value of 0.5 for each pair.

- Which areas should be targeted?

To accomplish this, **4** collections were joined using the MongoDB shell command

The four collections were reviews, orders, cust_record and geo_location.

Using the results from the join, I was able to plot the map of brazil with review scores using tableau.



Based on this map, I was able to mark the regions (with light blue marker) which have a concentration of negative scores.

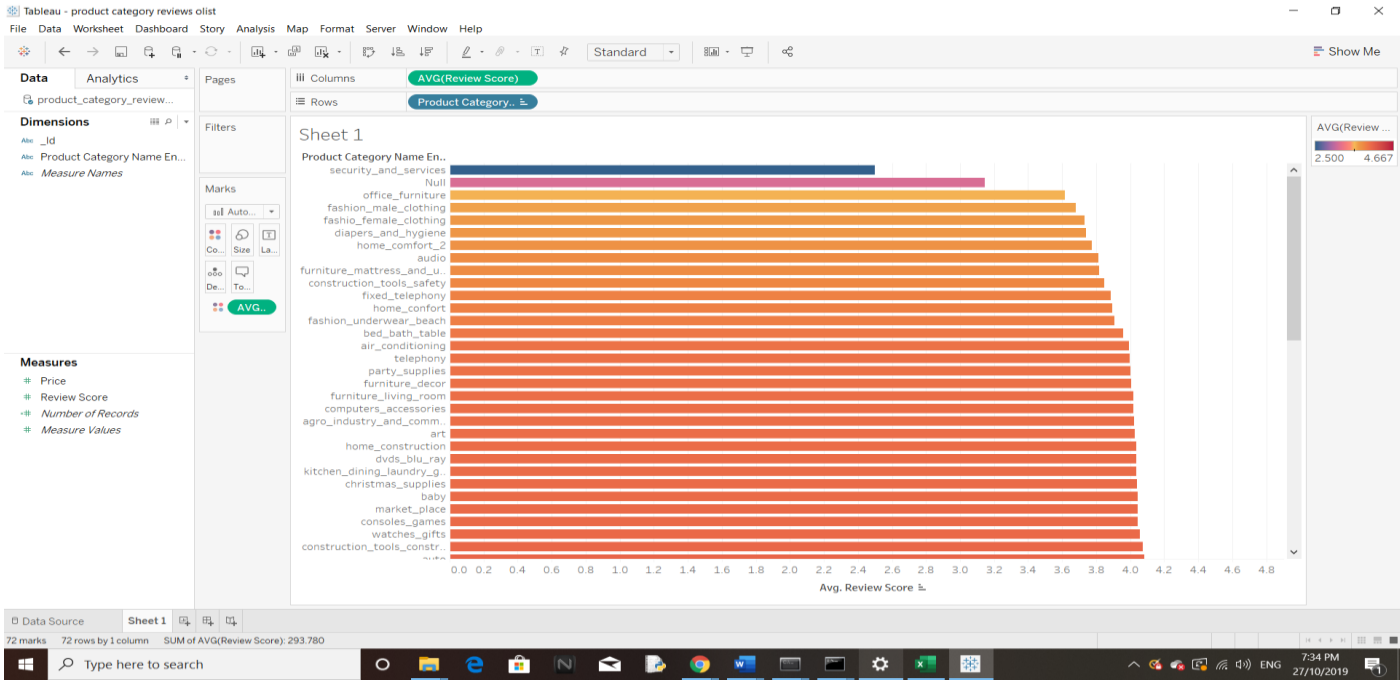
It could be beneficial to have more customer representatives in those areas to improve the service and satisfaction levels.

- Best and worst performing product categories

To accomplish this, I had to join **4** collections using the MongoDB shell command.

The collections used are reviews, orders_items, products and products_category_english

The results of the join were visualized using tableau as shown in the picture below.



The review_score vs product_category plot tells us that Security and services category receives the lowest score of 2.5 which is much lower than the other categories.

This information could be conveyed to the sellers of this category in order to improve their services.

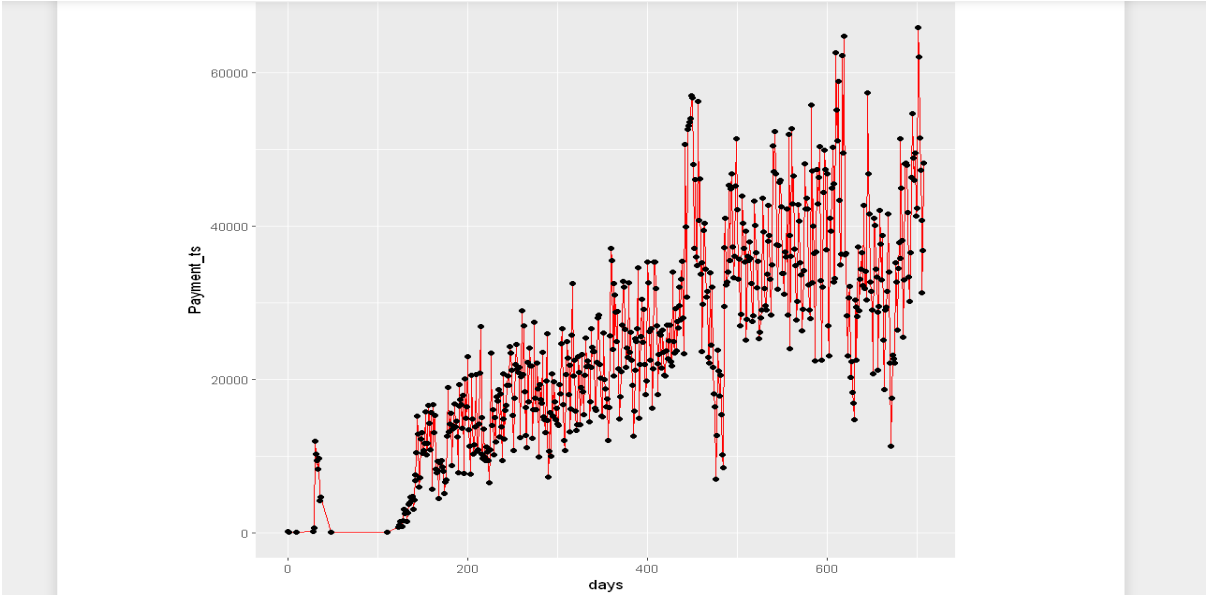
4. Methodology

This section describes the steps/methods used in obtaing the analysis results discussed in the previous section.

I have also added the reasons and inspiration for my choices

4.A Forecasting Methodology

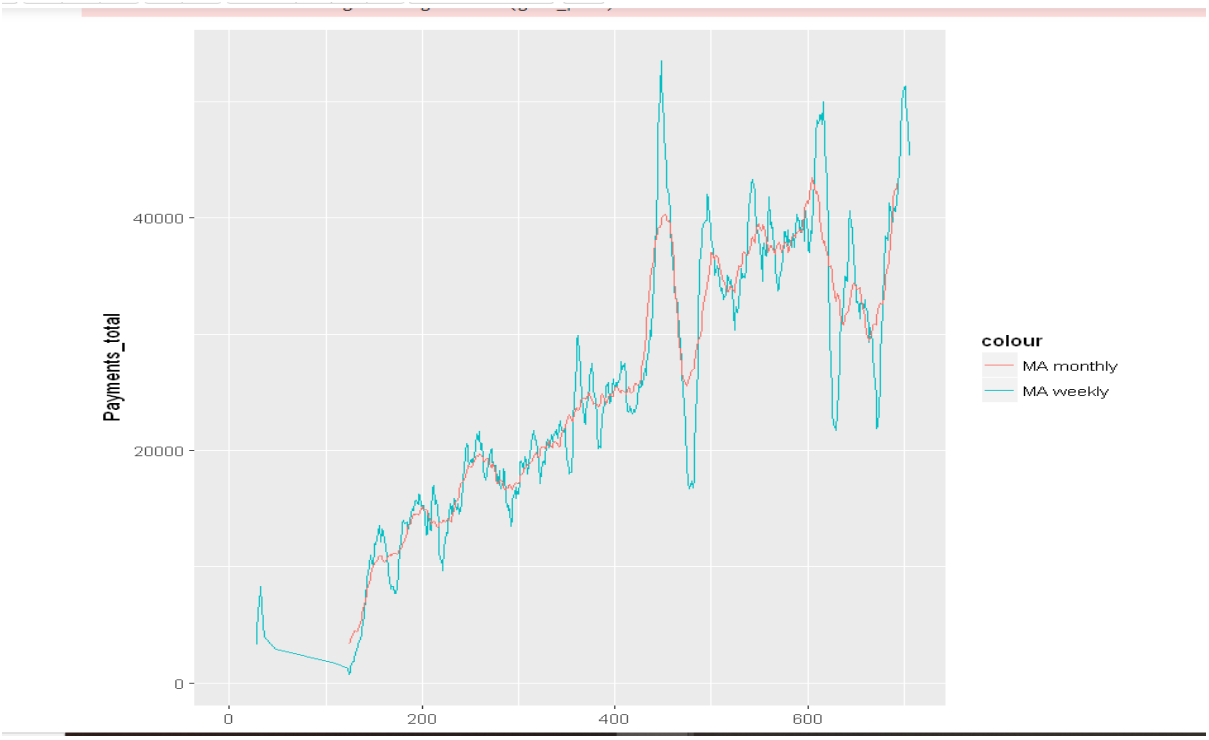
Before using the ARIMA model, there was some wrangling done as it the shape of the graph represented a non-stationary model



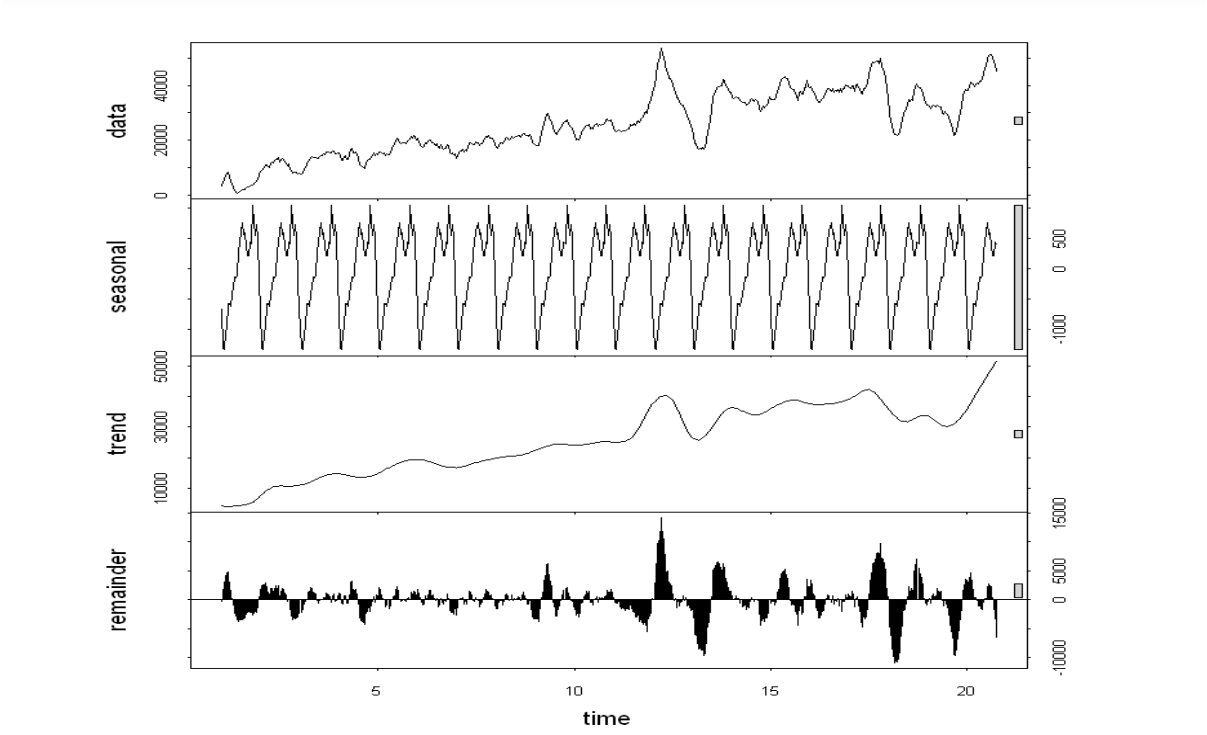
The following steps were taken –

1. Removal of outliers using the tsclean() function.

2. Using the moving average function ma() with weekly setting to smooth the data.

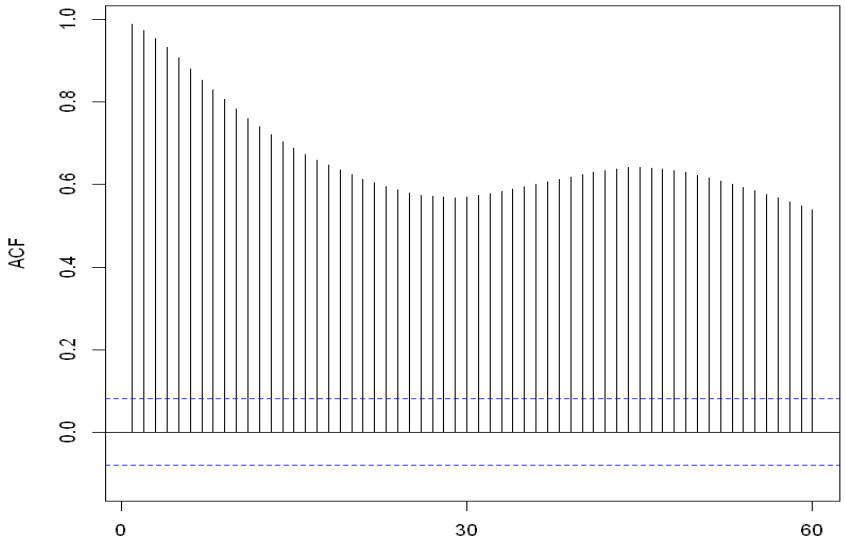


3. Decomposing the data to find trends, seasonality.

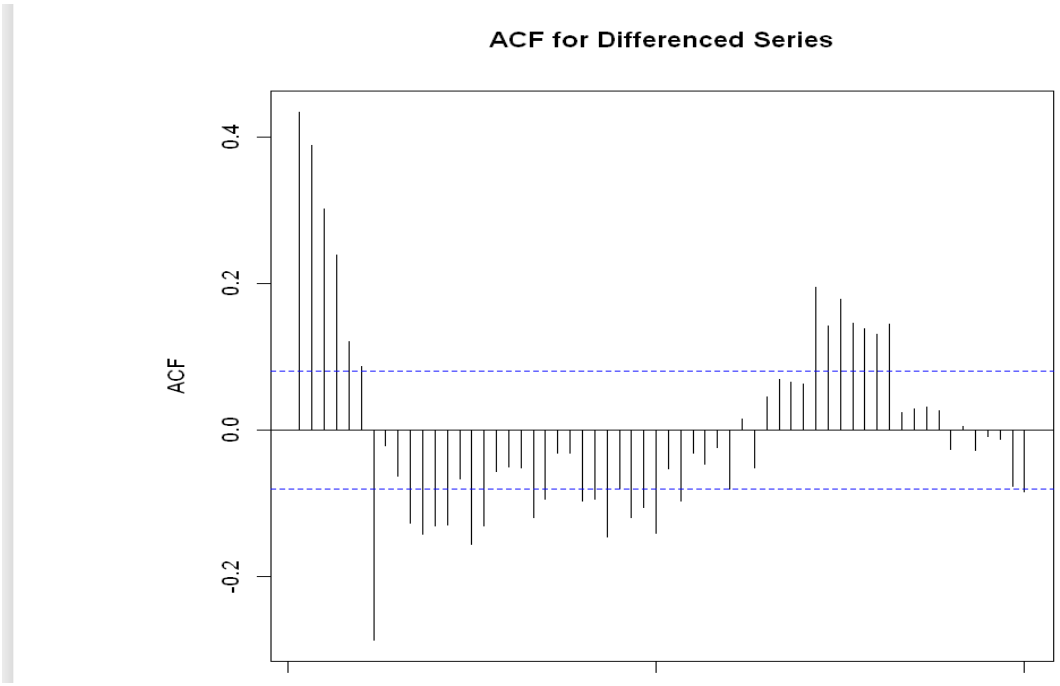


4. Using the de-seasonalized data and making it stationary by differencing the series.

ACF before differencing

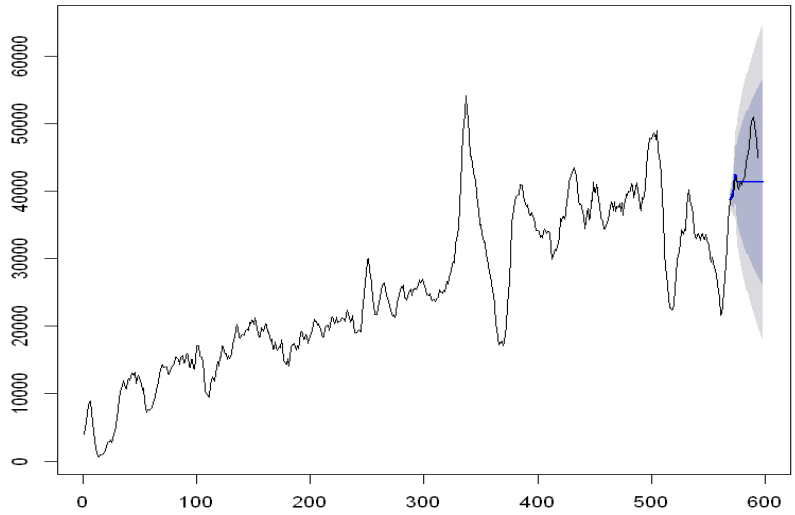


ACF after differencing



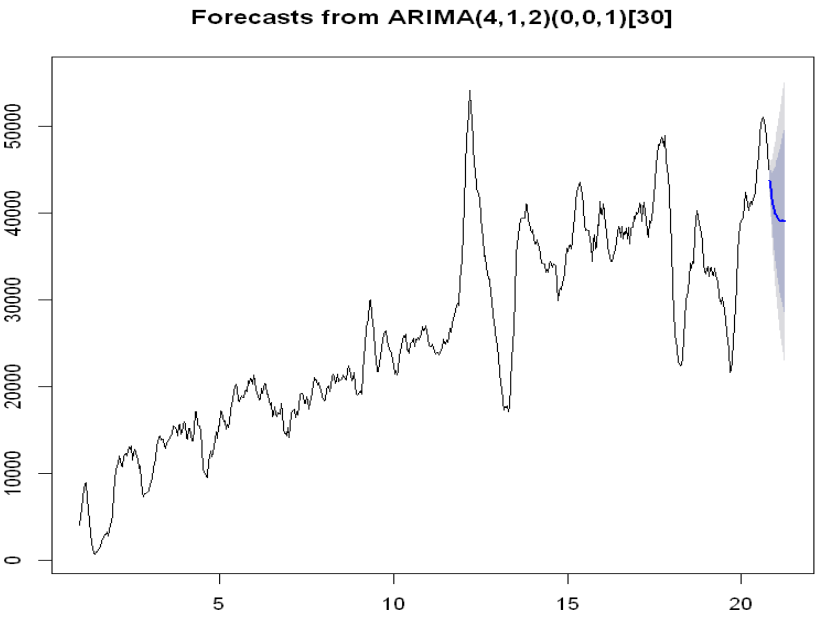
Spike in every 7th value indicates weekly trend.

5. Using auto-arima to calculate the order and forecast the sales (order – 1,1,7 result)



The model does determine the shape of the trend but flattens out pretty quickly(as seen by the blue line becoming straight)

6. **Optimizing** the plot by reintroducing the seasonality aspect (model – (4,1,2))



Just by visually inspecting, this shape/forecast look better.

I introduced the seasonality aspect because without it the shape and the forecast doesn't make much sense.

Why ARIMA?

- I used ARIMA because there's a seasonality and cyclic trend involved with the graph which would make models like Linear regression to ineffective for forecasting.
- Also, models like neural networks or regression models of higher order could **overfit** the data which would provide inaccurate forecasts.

4.B Determining the causes

The steps involved where

1. Subsetting the dataframe with rows having 1 as review_score and some comment associated with it.

Command :

```
x = subset(df5,df5[,10] == 1 & nchar(df5[,12]) > 0)
```

2. Using the quanteda package in R to perform stopword and punctuation removal as well as ngram generation.

Command :

```
trigrams <- quanteda::dfm(z, ,remove = stopwords("portuguese"),ngrams=3, concatenator=" ",remove_punct = TRUE)
```

Why Quanteda?

- Quanteda is an amazing package for text analysis in R and has support for languages like **Portuguese** which is vital for this dataset.
- Ngram generation and data wrangling is pretty straightforward.

4.C Effect of delivery time on reviews

1. Explicitly converting the columns with date-time features to date-time object for easier calculation like finding the difference

Command:

```
df5$order_approved_at <- as.POSIXct(df5$order_approved_at, format='%Y-%m-%d %H:%M:%S')
```

2. Introducing new features by taking the integer difference in time for two columns by the command below

Command:

```
df5$time_to_approve <- as.integer
(difftime(df5$order_approved_at,df5$order_purchase_timestamp,units="mins"))
```

3. Finding correlation between these newly created features/columns with the review score.

Command:

```
cor(df5$time_to_deliver,df5$review_score,use = "complete.obs")
```

Inspiration:

- Many E-commerce websites provide one day delivery like Amazon which has proven to be commercially successful as well as increased customer satisfaction.

4.D Determining target areas:

1. To analyze this, I had to join the four collections(mentioned in analysis) using the **mongoshell** command prompt and not the dataframe in R. The code I wrote for this is shown below

This is to display the logic, the code was executed in mongoshell as shown in the next image.

```
In [ ]: db.reviews.aggregate([
    {$lookup:
     {from: "orders",
      localField: "order_id",
      foreignField:"order_id",
      as: "order_reviews"}
    },
    {$replaceRoot:
     { newRoot:
      { $mergeObjects: [ { $arrayElemAt: [ "$order_reviews", 0 ] }, "$$ROOT" ] } }},
    { $project: { order_reviews: 0 } },

    {"$lookup":
     {from: "cust_record",
      localField: "customer_id",
      foreignField:"customer_id",
      as: "cust_data"}},
    {$replaceRoot:
     { newRoot: { $mergeObjects: [ { $arrayElemAt: [ "$cust_data", 0 ] }, "$$ROOT" ] } }},
    { $project: { cust_data: 0 } },

    {"$lookup":
     {"from": "geo_records",
      "localField": "geolocation_zip_code_prefix",
      "foreignField":"geolocation_zip_code_prefix",
      "as": "geo_data"}},
    {$replaceRoot: { newRoot: { $mergeObjects: [ { $arrayElemAt: [ "$geo_data", 0 ] }, "$$ROOT" ] } }},
    { $project: { geo_data : 0 } },

    {"$project":
     {"order_id": 1,
      "customer_id": 1,
      "review_score": 1,
      "customer_city": 1,
      "customer_state": 1,
      "geolocation_zip_code_prefix" : 1,
      "geolocation_lng" : 1,
      "geolocation_lat" : 1}},

    { $out : "review analysis" }])
```

Command in Mongoshell

```
> db.reviews.aggregate([{$lookup:{from: "orders_items",localField: "order_id",foreignField:"order_id1",as: "order_reviews"}},{$replaceRoot: { newRoot: { $mergeObjects: [ { $arrayElemAt: [ "$order_reviews", 0 ] }
, "$$ROOT" ] } }}, { $project: { order_reviews: 0 } },{"$lookup":{from: "products",localField: "product_id1",foreignField:"product_id",as: "prod_data"}},{$replaceRoot: { newRoot: { $mergeObjects: [ { $arrayElemA
t: [ "$prod_data", 0 ] }, "$$ROOT" ] } }},{ $project: { prod_data: 0 } },{"$lookup":{"from": "products_category_english","localField": "product_category_name","foreignField":"product_category_name1","as": "trans
lated_name"}},{$replaceRoot: { newRoot: { $mergeObjects: [ { $arrayElemAt: [ "$translated_name", 0 ] }, "$$ROOT" ] } }},{ $project: { translated_name : 0 } },{"$project": {"product_category_name_english":1,"revi
ew_score": 1,"price":1}},{ $out : "product_category_reviews" }])
```

Replaceroot – To convert the nested features to normal features/columns of a dataframe. Without this, all the columns/features of the second table would be fit in a nested feature structure making it difficult for analysis.

Project – Considers only the mentioned fields and rejects the other fields.

2. This new collection review_analysis was then sent to the tableau for visualization using the **MongoDB BI connector.**

Why this approach and not the dataframe join/relational join

- Firstly, since we are using MongoDB, it seems counterintuitive to use a relational join as the purpose of MongoDB is the use **non-relational** database.
- In one to one relation the results would be identical no matter which structure is in use, but for one to many relations the result changes as the non-relational structure will have a nested structure and the relational structure will have the more rows.

4.E Determining worst product category

Same working as 4.D with different collections and a change in code.

5.Conclusion:

Based on the data analysis and plots, I have reached at the following conclusions.

- Regarding expansion goals – From the plot it is evident that the next cycle is predicted to have a decrease in sales. Hence, it would be advisable to wait a few cycles when the sales figure increase again.

- Regarding the cause – Customers not receiving their product is a major challenge that needs to be overcome to improve the customer satisfaction level. Additional measures such as logistics tracking could be introduced to improve on this issue.

- Regarding the one-day delivery program – Since the correlation between the delivery time and he reviews is not strong, it suggests that the prime-delivery may not be a very successful model in Brazil. This could be because Brazil is still under the developing category and people could prioritize prices over delivery.

- Regarding the target area – Based on the analysis, location seems to be a factor that could determine the customer satisfaction level. Before planning more kiosks or stores, it is vital to study if it's due to the high population levels in that area or a genuine correlation with those areas.

- Since the product category Security and services has substantially low rating (2.5), an internal investigation needs to be performed in order to determine the cause.

6. Appendix

1. Performing a join operation in MongoDb shell is a very expensive operation which took me **5 hours** to perform whereas in tableau it takes close to 10 seconds.

In order to fix this issue, I had to research ways of improving this issue.

- **Optimizing** the performance time

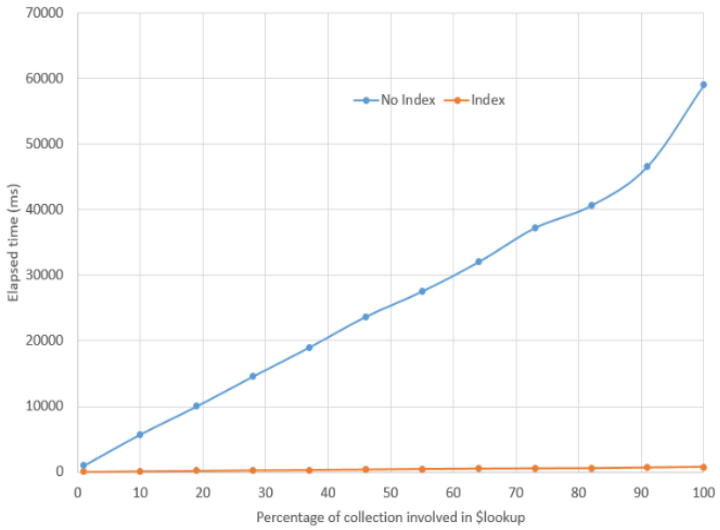
Using the createIndex, command creates an index (similar to id) which then facilitates hash join which is exponentially faster than a normal join.

Command:

db.products_category_english.createIndex({ product_category_name1: 1 })

where products_category_english is the collection name and the product_category_name1 is the foreign field

The difference after optimizing is shown below and I had the similar experience with my performance time reducing from 5 hours to 10 seconds



2. Socket- timeout error.

The default time-limit for operations in R using MongoDB's Mongolite package is 20 mins.

To optimize the process and improve the time limit , I had to modify the socke connection limit by the command below(check the URL part below)

Command:

dmd7 = mongo(collection="payments",db="customer_records",url = "mongodb://localhost?sockettimeoutms=10000000")