MCO008-051008

Data Analytics in Supply Chain Management

Fall 2022

Prof. Dr.-Ing. Hendro Wicaksono

Group Project Work

Group Members

1	Abhishek Mathur	30006043
2	Mehr un Nisa	30005960
3	Meriel Wanyonyi	30006061
4	Mohammad Tanzil Alam	30006050

Date: 09th December 2022

Table of Contents

MCO008-051008 Data Analytics in Supply Chain Management Fall 2022 I
Prof. Dr.-Ing. Hendro Wicaksono I
Table of Contents II
List of Figures IV
List of Tables V
1 Supply Chain Scenario 1 : Online Retail 1
1.1 Scenario Description 1
1.2 Dataset Description 2
1.3 Exploratory Data Analysis 2
1.4 Data Preprocessing 5
1.5 Model Building and Result Description 7
1.6 Discussion 14
1.7 Business and Technological Implication 14
1.7.1 Business Implications 14
1.7.2 Business Disruptions 15
1.8 Business Solutions 16
1.9 Business Model Canvas 17
2 Supply Chain Scenario 2: DataCo SupplyChain Dataset 18
2.1 Scenario Description 18
2.2 Dataset Description 18
2.3 Exploratory Data Analysis 21
2.4 Data Preprocessing 25
2.5 Model Building and Result Description 27
2.6 Discussion 32
2.7 Business and Technological Implication 32
2.8 Business Challenges 33
2.9 Business Model Canvas 34
3 Supply Chain Scenario 3: CNC-Milling Machine_Production data 35
3.1 Scenario Description 35
3.2 Dataset Description 35
3.2.1 **Dependent Variables:** Processing time and average power consumption 36
3.3 Exploratory Data Analysis 36
3.4 Data Preprocessing 44
3.5 Model Building and Result Description 44
3.6 Discussion 48

3.7 Business and Technological Implication 49
3.7.1 Business Disruptions: 49
4 Conclusions Future Outlook, and Reflection 50
Bibliography 51

List of Figures

Figure 1. 10 Rows 3
Figure 2. 10 customers 3
Figure 3. Bottom five countries 4
Figure 4. Seasonality 4
Figure 5. Weekly Sales 5
Figure 6. Data Preporcessing 5
Figure 7. Preporcessed Data 6
Figure 8. No data with negative columns 6
Figure 9. Dataset Info 6
Figure 10. Dataset Info 7
Figure 11 : RFM values calculated per customer 9
Figure 12 RFM score(1-4) calculated per customer 11
Figure 13: Loyalty Levels defined 12
Figure 14: Bar Graph for count of customers as per Loyalty Level 12
Figure 15: Bar Graph for count of customers as per Loyalty Level 13
Figure 16: Graph for Elbow method 13
Figure 17 Kmeans clusters as result 14
Figure 18 Bussiness Model Canvas of Online Retail 17
Figure 19 21
Figure 20:Correlation Heatmap 22
Figure 21: Box plot of the variables of the dataset 23
Figure 22: Histogram plot of each variable 24
Figure 23: Scatter plot for sales and late delivery 25
Figure 24 : Category library 27
Figure 25: Confusion Matrix Logitstic Regression 28
Figure 26: XG Booster 30
Figure 27: Kernel SVM Confusion Matrix 31
Figure 28 Bussiness Model Canvas 34
Figure 29: Statistic about the data type of the dataset. 37
Figure 30: Statistic about missing values 38
Figure 31: Correlation heatmap of every variable. 39
Figure 32: Scatter plot of the target variables against some predictors. 40
Figure 33: Box plot of the independent and dependent variables. 41
Figure 34: Histogram 42
Figure 35: line plot to see evolution of average power consumption 42
Figure 36: Bar Plot between average power consumption and processing time 43
Figure 37: Average power consumption over time for the same material of 136659600 mm3 43
Figure 38: Feature Importance 47

List of Tables

Table 1: Data set description of Online Retail 2
Table 2: Reveal the output after encoding. 27
Table 3 30
Table 4: Kernel SVM Results 31
Table 5: Comparison Table 31
Table 6 CNC Machine dataset 35
Table 7: Different type of performance Metrics for Random Forest 45
Table 8: Different type of performance Metrics for Decision Tree 46
Table 9: Different type of performance Metrics for Support Vector 46
Table 10: Different type of performance Metrics for all models after cleaning 47
Table 11: Different type of performance Metrics for all models before cleaning 48

1 Supply Chain Scenario 1 : Online Retail

This is the model of the online retail. It consists of different sectors to ensure smooth run of buying, selling and delivering of products to end consumers.an online retail consists of key partners who are the online sellers and transportation companies to ensure delivery of demanded items to respective consumers. Day and time of demand can be an indicator ,the earliest date taken into consideration and deliver fast. Key activities are receivig all orders from customers and ensure efficient delivery to them.the channels to be used are mostly social media shopping apps and websites, consumers are taken into consideration to ensure their satisfaction is met. The dataset would be analysed to check the demand of the product and improve where necessary.

1.1 Scenario Description

Online retail deals with online sale of products according to their demand rate. The demand is focused on country wise demand, month and day of demand. Our main objectives are customer retention for customers who demand more. This can be manifested through gifts vouchers, a customer will be willing to purchase more to get the benefits associated with it. As every business every customer demands differently, there are the low demanding population. Marketing campaign through advertisement to educate people on the product, discounts can be offered for specific purchases. Businesses have an aim of a profit, and this can be boosted by more sales to customers. To understand this a dataset of online retail of 13 months was analysed to see its relevance. It was analysed on three main aspects frequency, recency and monetary. Frequency is the count of Invoice number of transactions carried out for the 13 months, monetary is the sum of total sales while recency is the difference between latest date and last invoice date.

1.2 Dataset Description

Table 1: Data set description of Online Retail

fields	description
Invoice number	Number assigned to customers to track payments.
Invoice Date	Represents time stamped time and date goods have been billed.
Invoice Time	Period which invoices are to be issued
Stock Code	It is an abbreviation that identifies a particular security on a product.
Description	It shows what the product is made up of.
Quantity	It shows the amount ordered.
Unit Price	It is the price of each amount in certain unit ordered
Total sales	It is the summation of product of unit sales and unit prices of each item.
Customer ID	It is the unique means of identification allocated by customer in relation of services offered.
Country	It shows countries demand of particular products.

1.3 Exploratory Data Analysis

The is having 541909 rows and 10 columns before pre processing.

	InvoiceNo	InvoiceDate	InvoiceTime	StockCode	Description	Quantity	UnitPrice	Totalsale	CustomerID	Country
0	536365	01-12-2010	08:26:00 AM	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2.55	15.30	17850.0	United Kingdom
1	536365	01-12-2010	08:26:00 AM	71053	WHITE METAL LANTERN	6	3.39	20.34	17850.0	United Kingdom
2	536365	01-12-2010	08:26:00 AM	84406B	CREAM CUPID HEARTS COAT HANGER	8	2.75	22.00	17850.0	United Kingdom
3	536365	01-12-2010	08:26:00 AM	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	3.39	20.34	17850.0	United Kingdom
4	536365	01-12-2010	08:26:00 AM	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	3.39	20.34	17850.0	United Kingdom
...	...	...	...	...	...	...	...	...	...	...
541904	581587	09-12-2011	12:50:00 PM	22613	PACK OF 20 SPACEBOY NAPKINS	12	0.85	10.20	12680.0	France
541905	581587	09-12-2011	12:50:00 PM	22899	CHILDREN'S APRON DOLLY GIRL	6	2.10	12.60	12680.0	France
541906	581587	09-12-2011	12:50:00 PM	23254	CHILDRENS CUTLERY DOLLY GIRL	4	4.15	16.60	12680.0	France
541907	581587	09-12-2011	12:50:00 PM	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	4.15	16.60	12680.0	France
541908	581587	09-12-2011	12:50:00 PM	22138	BAKING SET 9 PIECE RETROSPOT	3	4.95	14.85	12680.0	France

541909 rows × 10 columns

Figure 1. 10 Rows

Finding the top 10 customers who are recurrently purchase stuffs from the online retail.

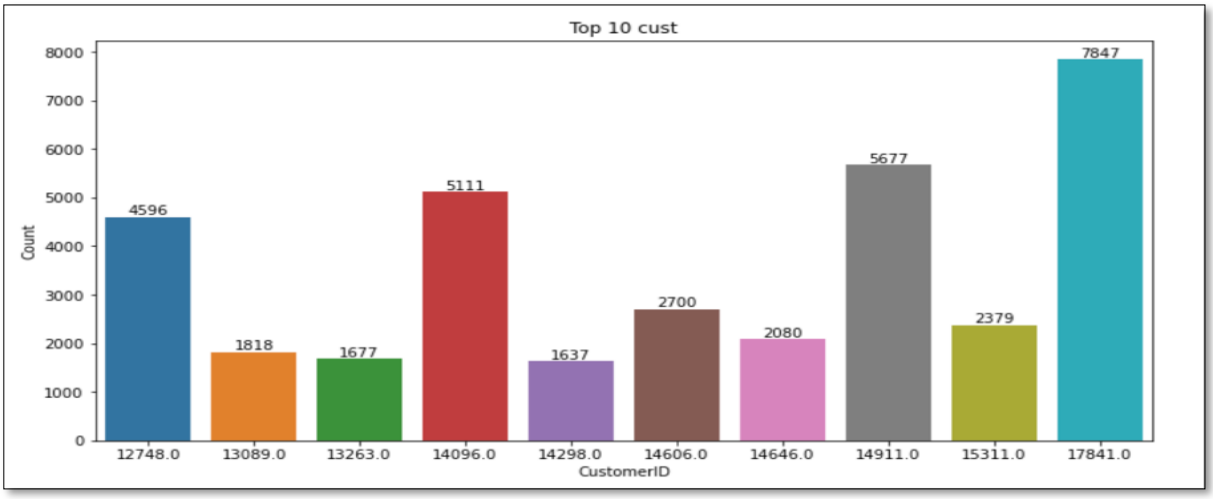


Figure 2. 10 customers

Bottom five countries having maximum customer counts.

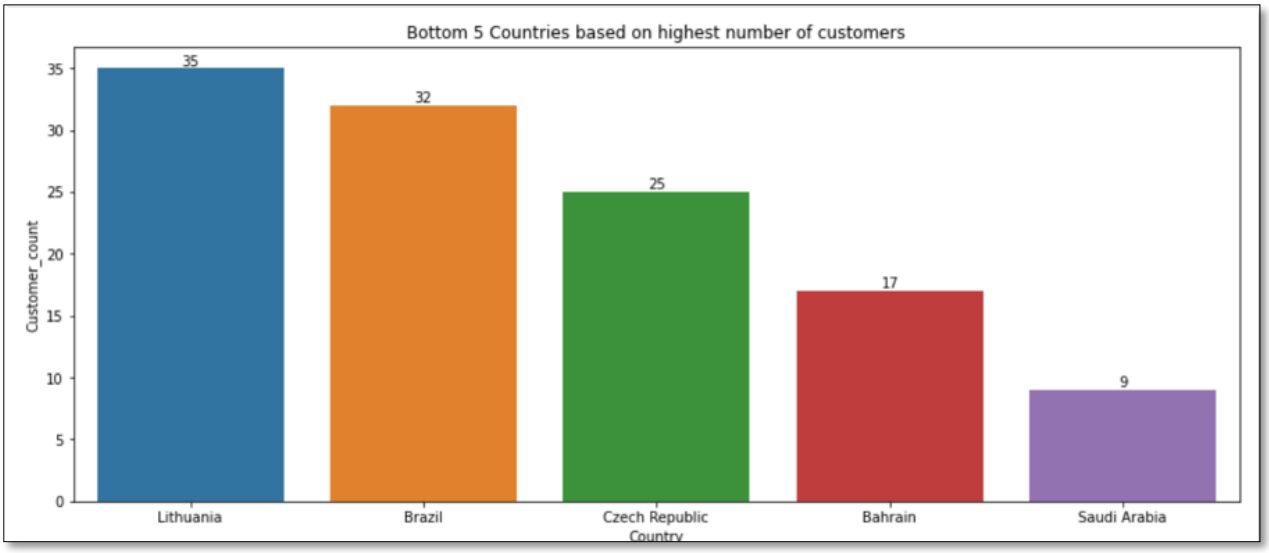


Figure 3. Bottom five countries

Plot of different months, indicating most of the sales is done in October and November and least is on January, February and December.

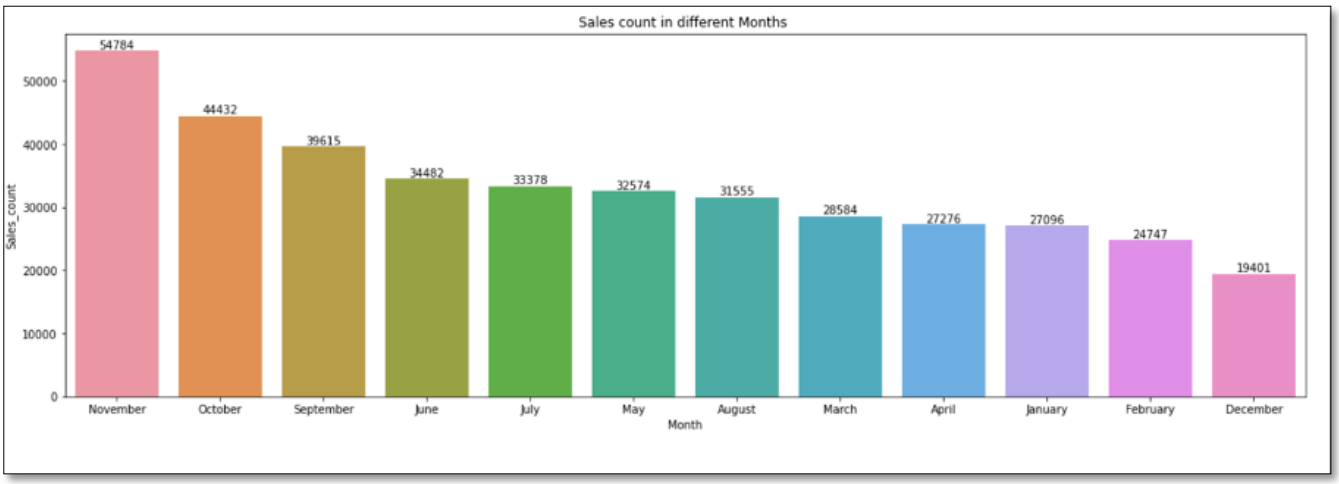


Figure 4. Seasonality

Sales count on different days.

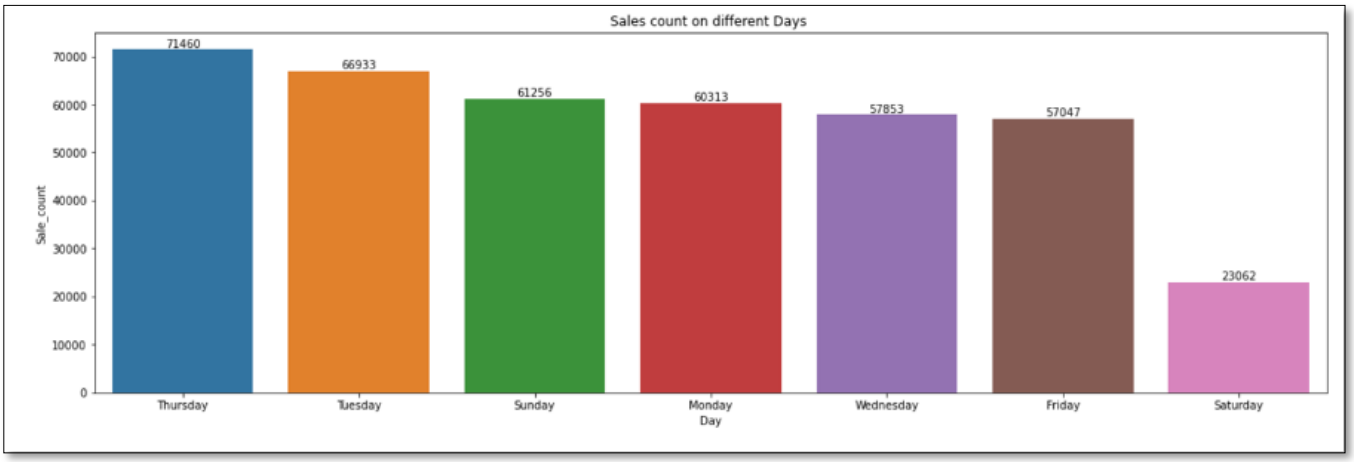


Figure 5. Weekly Sales

1.4 Data Preprocessing

For further execution of the project, we have joined two columns and merged them into one that is “InvoiceDate” and then we have to change the dtype of them to datetime. Then we have drop the column InvoiceTime, and extract the month and day out of the newly created InvoiceDate

```
# Using + operator to combine two columns
df["InvoiceDate"] = df['InvoiceDate'] +' '+ df["InvoiceTime"]
# df.head()

# Changing dtype of Invoicedate
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])

# Removing the InvoiceTime as it is merged with InvoiceDate
df=df.drop(['InvoiceTime'], axis=1)

# Deopping all null values
df = df.dropna()

# extracting month from the Invoice date
df['Month']=df['InvoiceDate'].dt.month_name()
# extracting day from the Invoice date
df['Day']=df['InvoiceDate'].dt.day_name()
```

Figure 6. Data Preporcessing

After preprocessing we are left with 406829 rows and 11 columns

	InvoiceNo	InvoiceDate	StockCode	Description	Quantity	UnitPrice	Totalsale	CustomerID	Country	Month	Day
0	536365	2010-01-12 08:26:00	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2.55	15.30	17850.0	United Kingdom	January	Tuesday
1	536365	2010-01-12 08:26:00	71053	WHITE METAL LANTERN	6	3.39	20.34	17850.0	United Kingdom	January	Tuesday
2	536365	2010-01-12 08:26:00	84406B	CREAM CUPID HEARTS COAT HANGER	8	2.75	22.00	17850.0	United Kingdom	January	Tuesday
3	536365	2010-01-12 08:26:00	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	3.39	20.34	17850.0	United Kingdom	January	Tuesday
4	536365	2010-01-12 08:26:00	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	3.39	20.34	17850.0	United Kingdom	January	Tuesday
...	...	...	...	...	...	...	...	...	...	...	...
541904	581587	2011-09-12 12:50:00	22613	PACK OF 20 SPACEBOY NAPKINS	12	0.85	10.20	12680.0	France	September	Monday
541905	581587	2011-09-12 12:50:00	22899	CHILDREN'S APRON DOLLY GIRL	6	2.10	12.60	12680.0	France	September	Monday
541906	581587	2011-09-12 12:50:00	23254	CHILDRENS CUTLERY DOLLY GIRL	4	4.15	16.60	12680.0	France	September	Monday
541907	581587	2011-09-12 12:50:00	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	4.15	16.60	12680.0	France	September	Monday
541908	581587	2011-09-12 12:50:00	22138	BAKING SET 9 PIECE RETROSPOT	3	4.95	14.85	12680.0	France	September	Monday

406829 rows × 11 columns

Figure 7. Preporcessed Data

Removing the negative columns from the Quantity to assuming that there is no negative quantity.

```
'''
As negative valued quantity is not of our concern for the project idea, so finding and dropping it from the dataset.
'''
#drop -ve values
a= df[df ['Quantity'] < 0].index
df.drop(a , inplace=True)
df[df ['Quantity'] < 0]
```

InvoiceNo	InvoiceDate	StockCode	Description	Quantity	UnitPrice	Totalsale	CustomerID	Country	Month	Day

Figure 8. No data with negative columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 397924 entries, 0 to 541908
Data columns (total 11 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    397924 non-null  object
 1   InvoiceDate  397924 non-null  datetime64[ns]
 2   StockCode    397924 non-null  object
 3   Description  397924 non-null  object
 4   Quantity     397924 non-null  int64
 5   UnitPrice    397924 non-null  float64
 6   Totalsale    397924 non-null  float64
 7   CustomerID   397924 non-null  float64
 8   Country      397924 non-null  object
 9   Month        397924 non-null  object
 10  Day          397924 non-null  object
dtypes: datetime64[ns](1), float64(3), int64(1), object(6)
memory usage: 36.4+ MB
```

Figure 9. Dataset Info

	Quantity	UnitPrice	Totalsale	CustomerID
count	397924.000000	397924.000000	397924.000000	397924.000000
mean	13.021823	3.116174	22.394748	15294.315171
std	180.420210	22.096788	309.055588	1713.169877
min	1.000000	0.000000	0.000000	12346.000000
25%	2.000000	1.250000	4.680000	13969.000000
50%	6.000000	1.950000	11.800000	15159.000000
75%	12.000000	3.750000	19.800000	16795.000000
max	80995.000000	8142.750000	168469.600000	18287.000000

Figure 10. Dataset Info

1.5 Model Building and Result Description

1.5.1 RFM

RFM is a technique which is used for analysing customer value. stands for Recency, Frequency and Monetary. It is usually used in direct marketing and database marketing and has received specific attention in retail and professional services industries.

RMF stands for:

Recency- How recently did the customer purchase?

Frequency- How often do they purchase?

Monetary- How much do they spend?

Customer purchases is represented by our dataset table with columns for the customer ID, date of purchase, Invoice Number and purchase value. One approach to RFM is to assign a score for each dimension on a scale from 1 to 10. In our approach, scale from 1-4 is chosen where 1 being the excellent value and 4 being the bad value. The minimum score represents the preferred behaviour, and a formula could be used to calculate the three scores for each customer. For example, a service-based business could use these calculations:

I) Recency = Latest date(2011, 12, 10) – Last Purchase date for each customer.
II) Frequency = the number of purchases by the customer in the last 12 months.
III) Monetary = the sum of values of all purchases by each customer.

Alternatively, categories can be defined for each attribute. For instance, Recency might be broken into three categories: customers with purchases within the last 90 days; between 91 and 365 days; and longer than 365 days. Such categories may be derived from business rules or using data mining techniques to find meaningful breaks.

To create RFM Modelling scores for each customer, following steps were taken:

I) Recency is calculated as Latest Date subtracting Last Inovice Data. Latest date is taken as '2011-12-10' as last invoice date was 2011-12-09. This is to calculate the number of days from recent purchase.
II) Frequency is calculated as count of invoice no. of transaction(s).
III) Monetary is calculated as Sum of Total amount of purchase by each customer.

First, converted the Latest_Date column to datetime datatype.

```
Latest_Date = dt.datetime(2011,12,10)
```

```
rfm_normalized = df.groupby('CustomerID').agg({'InvoiceDate': lambda x: (Latest_Date - x.max()).days, 'InvoiceNo': lambda x: len(x), 'Totalsale': lambda x: x.sum()})
```

Converted Invoice Date into type int.

```
rfm_normalized['InvoiceDate'] = rfm_normalized['InvoiceDate'].astype(int).
```

For simplicity, we have renamed column names to Recency, Frequency and Monetary and then normaloized the result.

```
rfm_normalized.rename(columns={'InvoiceDate': 'Recency', 'InvoiceNo': 'Frequency', 'Totalsale': 'Monetary'}, inplace=True)
```

```
rfm_normalized=rfm_normalized.reset_index()
```

In below screenshot, we can see the resulted RFM values for each customer.

	CustomerID	Recency	Frequency	Monetary	R	F	M
0	12346.0	325	1	77183.60	4	4	1
1	12347.0	39	182	4310.00	2	1	1
2	12348.0	75	31	1797.24	3	3	1
3	12349.0	18	73	1757.55	1	2	1
4	12350.0	310	17	334.40	4	4	3

Figure 11 : RFM values calculated per customer

As per the result above, we observed there were still some abnormalities in the dataset that needed to be treated. In recency, values are calculated as subtraction of last purchase date from the lasted date. That can be either zero or more than zero. However, some values were coming as negative values. Hence, we decided to delete those data because such data were very less. Same assumption and strategy were applied for monetary.

```
m=rfm_normalized[rfm_normalized['Monetary']<=0].index
rfm_normalized.drop(m , inplace=True)
mm=rfm_normalized['Monetary']<=0
mm.sum()

r=rfm_normalized[rfm_normalized['Recency']<0].index
rfm_normalized.drop(r , inplace=True)
rr=rfm_normalized['Recency']<0
rr.sum()
```

So as per the above logic, values below zero were eliminated from both Recency and Monetary.

1.5.2 RFM Model Analysis

To analyse the RFM features, we decided to divide them into quantiles and to label them as 1-4 score. So, we split the data into four segment using Quantile as below.

```
quantile = rfm_normalized.quantile(q = [0.25,0.50,0.75])
quantile = quantile.to_dict()
```

quantile

```
{
'CustomerID': {0.25: 13817.0, 0.5: 15303.0, 0.75: 16776.0},
 'Recency': {0.25: 22.0, 0.5: 63.0, 0.75: 163.0},
 'Frequency': {0.25: 17.0, 0.5: 41.0, 0.75: 99.0},
 'Monetary': {0.25: 305.78, 0.5: 663.65, 0.75: 1625.0500000000002}
}
```

Now here, for each customer, we have received quantile. Next, we have given them score. Lower the recency score, it is assumed to be good for the company because the customer has recently purchased something.

```
def RScoring(x,p,d):
    if x <= d[p][0.25]:
        return 1
    elif x <= d[p][0.50]:
        return 2
    elif x <= d[p][0.75]:
        return 3
    else:
        return 4
```

Higher value of frequency and monetary lead to a good consumer. Hence, here we given higher value as 1 in reverse way.

```
def FnMScoring(x,p,d):
    if x <= d[p][0.25]:
        return 4
    elif x <= d[p][0.50]:
        return 3
    elif x <= d[p][0.75]:
        return 2
    else:
        return 1
```

Once we have got the RFM score for each customer, we have calculated and added R,F and M segments values columns in the existing dataset to show R,F,M segment values.

```
rfm_normalized["R"] =
rfm_normalized['Recency'].apply(RScoring,args=('Recency',quantile,))

rfm_normalized["F"] =
rfm_normalized['Frequency'].apply(FnMScoring,args=('Frequency',quantile,))

rfm_normalized["M"] =
rfm_normalized['Monetary'].apply(FnMScoring,args=('Monetary',quantile,))
```

Below figure shows us the result using 5 rows from dataset.

	CustomerID	Recency	Frequency	Monetary	R	F	M
0	12346.0	325	1	77183.60	4	4	1
1	12347.0	39	182	4310.00	2	1	1
2	12348.0	75	31	1797.24	3	3	1
3	12349.0	18	73	1757.55	1	2	1
4	12350.0	310	17	334.40	4	4	3

Figure 12 RFM score(1-4) calculated per customer

To find loyalty level of each customer, we have used RFM scores. We have added RFM scores of recency, frequency and Monetary for each customer and created a new field for that. As per assumption, we have given importance to each recency, frequency, and monetary values. So as per logic create in above sections, if the total score comes less (minimum is 1+1+1 = 3), mean that customer is the most loyal one. Similarly, if the customer gets total RFM score of 12(maximum 4+4+4 = 12), mean that customer is of least important for the product companies. Also, we have assigned each customer with Labels as Platinum, Gold, Silver and Bronze.

Below is the resulted screenshot of the dataset for 5 customers.

	index	CustomerID	Recency	Frequency	Monetary	R	F	M	RFM_Group	RFM_Score	RFM_Loyalty_Level
0	0	12346.0	325	1	77183.60	4	4	1	441	9	Silver
1	1	12347.0	39	182	4310.00	2	1	1	211	4	Platinaum
2	2	12348.0	75	31	1797.24	3	3	1	331	7	Gold
3	3	12349.0	18	73	1757.55	1	2	1	121	4	Platinaum
4	4	12350.0	310	17	334.40	4	4	3	443	11	Bronz

Figure 13: Loyalty Levels defined

Below Graph shows the classification of customers on the basic of the four labels.

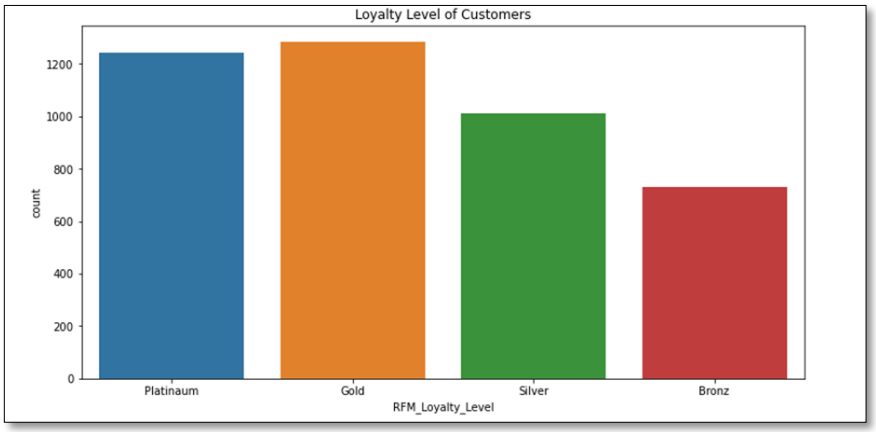


Figure 14: Bar Graph for count of customers as per Loyalty Level

It is important for any retail market to know their top best customers. So below we can see the top 10 customers that belongs to the Platinum category.

	index	CustomerID	Recency	Frequency	Monetary	R	F	M	RFM_Group	RFM_Score	RFM_Loyalty_Level
0	1690	14646.0	1	2080	280206.02	1	1	1	111	3	Platinaum
1	4202	18102.0	11	431	259657.30	1	1	1	111	3	Platinaum
2	3729	17450.0	2	337	194550.79	1	1	1	111	3	Platinaum
3	55	12415.0	24	716	124914.53	2	1	1	211	4	Platinaum
4	3772	17511.0	5	963	91062.38	1	1	1	111	3	Platinaum
5	2703	16029.0	29	242	81024.84	2	1	1	211	4	Platinaum
6	3177	16684.0	11	277	66653.56	1	1	1	111	3	Platinaum
7	1290	14096.0	11	5111	65164.79	1	1	1	111	3	Platinaum
8	997	13694.0	25	568	65039.62	2	1	1	211	4	Platinaum
9	562	13089.0	5	1818	58825.83	1	1	1	111	3	Platinaum

Figure 15: Bar Graph for count of customers as per Loyalty Level

1.5.3 KMeans Clustering

Before implementing the Kmeans Clustering algorithm we need to decide the number of clusters to put inside algorithm as input. So, we will be finding the minimum number of clusters required by using Elbow method.

After applying Elbow Method on Recency, Frequency and Monetary, below graph is what we get. And we can clearly see that an elbow is forming at point 3. So we will take that as our Kmeans values.

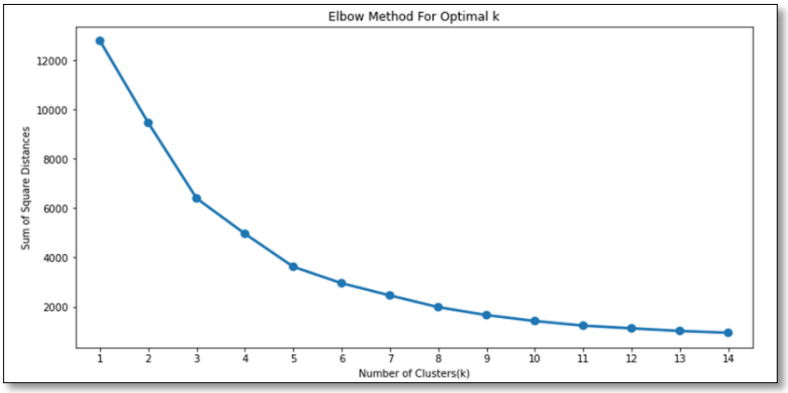


Figure 16: Graph for Elbow method

Once the algorithm is performed, below clusters we get.

	Recency			Frequency			Monetary			
	mean	min	max	mean	min	max	mean	min	max	count
Cluster_based_on_freq_mon_rec										
0	261.106438	145	696	29.792275	1	378	563.567279	3.75	77183.60	1165
1	48.210748	0	189	102.511816	1	1677	2065.431492	6.90	81024.84	3089
2	18.818182	1	88	1853.818182	3	5111	121841.620000	12156.65	280206.02	11

Figure 17 Kmeans clusters as result

As a result, a trend can be seen in the 3 respective clusters. Cluster 2 with low recency values, high frequency and monetary values is assumed to be the best cluster of customers. Whereas cluster 0 and 1 have comparatively high recency and low frequency and monetary values.

1.6 Discussion

As per the results and assumptions we had, taking all features of RFM into account, we have segregated the customers. With the help of the results, important and least important customers can be identified. Based on that, company can decide to lure them with gifts/discounts to reduce the churn rate. Using Kmeans cluster, we have divided customers into 3 clusters and based on it, business can decide varies strategies to retain its customers.

1.7 Business and Technological Implication

1.7.1 Business Implications

It is an effect of a policy or action that will have on operations and financial wellbeing of a Company. Online retail dataset noticed some of the activities which had shown its operations. They are as follows;

Recency and frequency have an effect on monetary gains.

The least recent and least frequent customer spend more. Customer ID 12346.0 has a recency of 325 and frequency of 1 but the spending is 77183.60. Comparing it to customer ID 12347.0 has a frequency of 182 and recency of 39 ,the monetary level is at 4310.0.

Combination of recency , monetary and, frequency assign loyalty levels to customers

Different customers have different purchase levels and through calculation of their recency, monetary, and frequency, loyalty levels are assigned accordingly. The loyalty levels is categorized into bronze, silver, gold and platinum.

1.7.2 Business Disruptions

This is the process by which a product becomes popular enough to replace a traditional or common product and services. Online retail has easened a lot of manpower with necessarily going to stores. This can be shown as follows;

- **Online Shopping**

This is whereby consumers directly buy goods, services over the Internet. Consumers visit different social apps from the comfort of their house. A variety is provided to cater different demands, this saves time that could have been used to go to online stores.

- **On demand delivery**

Customers have a choice to decide not just where they would like their products to be delivered but also when. This happens within the shortest delivery time possible.

- **E-commerce businesses**

After a consumer knows what he or she wants ,transfer of money and data is done for the transaction to be complete.Cancellation of orders can be made swiftly in case of consumer changes. There exist three types of E-commerce like business to business, business to consumer and consumer to consumer. Because of such services faster buying process, affordability in advertising and marketing, flexibility for customers , products and prices can be compared. The buyer and market demands are quickly responded to.

- **Personalisation and individuality.**

Consumers are always considered right and a business tries to satisfy what is demanded. This is the right quality,right quantity, right time and right price. Customers feedback is taken into consideration as it reflects the progress of the business.

1.8 Business Solutions

- **Gift vouchers for regular customers**

From analysis of the dataset cluster 2 has the highest monetary returns of 121841.620000, recency of 18.818182 and a frequency of 1853.818182. Since it has the highest monetary returns than cluster 0 and 1, gift vouchers can be given to the customers falling under that cluster to retain them and gain more revenue.

- **Advertisement and discounts for less demanding customers and churn rate reduction.**

Customers falling under cluster 0 have the lowest monetary returns .the business cannot loose on these customers so to boost their sales, more advertisement can be done for more product knowledge, satisfactory discounts can be offered for a specific purchase. Customers would spend more knowing the benefits associated with it.

The churn rate is the rate at which customers stop doing business with a company over a given period of time. It happens mostly for subscription goods where consumers end it after expiry of free trials. Subscription fees can be made cheaper for customers to ensure longer stay, the lower the churn rate the higher the profits, sales and growth of a company.

- **Aftersales Services**

These are supports provided to customers after purchase of products or services. This is a business strategy as it leads to brand loyalty, customer satisfaction and a possibility of reaching bigger markets.after sales services include warranty information, product or service training, repairs and upgrades, return and exchange policy details , issue of coupons and surveys.

1.9 Business Model Canvas

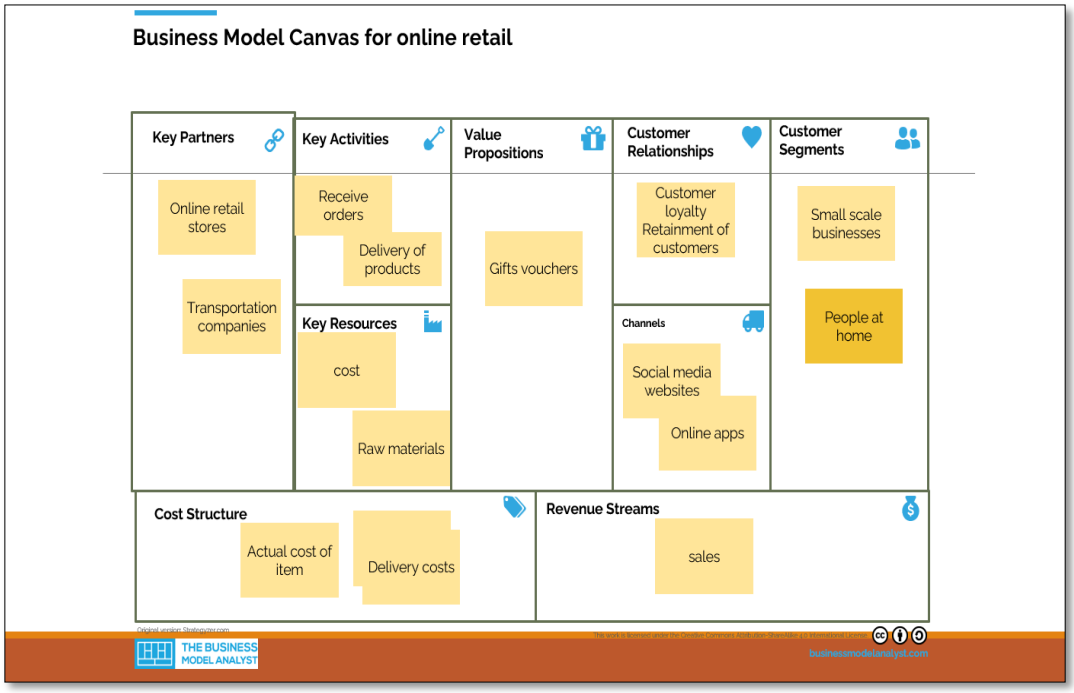


Figure 18 Bussiness Model Canvas of Online Retail

2 Supply Chain Scenario 2: DataCo SupplyChain Dataset

2.1 Scenario Description

DataCo SupplyChain is an e-commerce company deals with the products related to Footwear, apparel, fitness, fan shop, sports items, and electronics. The customers market for this company is global which means people from all over the world are ordering from this company. The company is facing two major issues, in order to tackle them the below mentioned data set will be analysed.

2.1.1 In order to improve sales in the months where the sale frequency is low. With the help of available data of three years ,firstly, the identification of the months with lower sales are required. Secondly the ideal time for advertisement has to be identified in order to boost the sales.

2.1.2 The company is facing an issue of late deliveries and in order to tackle the problem firstly , the identification of the factors which are causing late deliveries has to identified. Secondly, the decision for reducing the late delivery risk has to be taken.

2.2 Dataset Description

Table 2:Description of the dataset from DataCo SupplyChain(Constante et al., 2019)

FIELDS	DESCRIPTION
Type	Type of transaction made
Days for shipping (real)	Actual shipping days of the purchased product
Days for shipment (scheduled)	Days of scheduled delivery of the purchased product
Benefit per order	Earnings per order placed
Sales per customer	Total sales per customer made per customer
Delivery Status	Delivery status of orders: Advance shipping , Late delivery , Shipping canceled , Shipping on time
Late_delivery_risk	Categorical variable that indicates if sending is late (1), it is not late (0).

Category Id	Product category code
Category Name	Description of the product category
Customer City	City where the customer made the purchase
Customer Country	Country where the customer made the purchase
Customer Email	Customer's email
Customer Fname	Customer name
Customer Id	Customer ID
Customer Lname	Customer lastname
Customer Password	Masked customer key
Customer Segment	Types of Customers: Consumer , Corporate , Home Office
Customer State	State to which the store where the purchase is registered belongs
Customer Street	Street to which the store where the purchase is registered belongs
Customer Zipcode	Customer Zipcode
Department Id	Department code of store
Department Name	Department name of store
Latitude	Latitude corresponding to location of store
Longitude	Longitude corresponding to location of store
Market	Market to where the order is delivered : Africa , Europe , LATAM , Pacific Asia , USCA
Order City	Destination city of the order
Order Country	Destination country of the order
Order Customer Id	Customer order code
order date (DateOrders)	Date on which the order is made
Order Id	Order code
Order Item Cardprod Id	Product code generated through the RFID reader
Order Item Discount	Order item discount value
Order Item Discount Rate	Order item discount percentage
Order Item Id	Order item code
Order Item Product Price	Price of products without discount
Order Item Profit Ratio	Order Item Profit Ratio

Order Item Quantity	Number of products per order
Sales	Value in sales
Order Item Total	Total amount per order
Order Profit Per Order	Order Profit Per Order
Order Region	Region of the world where the order is delivered : Southeast Asia ,South Asia ,Oceania ,Eastern Asia, West Asia , West of USA , US Center , West Africa, Central Africa ,North Africa ,Western Europe ,Northern , Caribbean , South America ,East Africa ,Southern Europe , East of USA ,Canada ,Southern Africa , Central Asia , Europe , Central America, Eastern Europe , South of USA
Order State	State of the region where the order is delivered
Order Status	Order Status : COMPLETE , PENDING , CLOSED , PENDING_PAYMENT ,CANCELED , PROCESSING ,SUSPECTED_FRAUD ,ON_HOLD ,PAYMENT_REVIEW
Product Card Id	Product code
Product Category Id	Product category code
Product Description	Product Description
Product Image	Link of visit and purchase of the product
Product Name	Product Name
Product Price	Product Price
Product Status	Status of the product stock :If it is 1 not available , 0 the product is available
Shipping date (DateOrders)	Exact date and time of shipment
Shipping Mode	The following shipping modes are presented : Standard Class , First Class , Second Class , Same Day

2.2.1 Dependent Variable: Late Delivery Risk

2.2.2 Independent Variable:

- Days for shipping (real)
- Days for shipment (scheduled)
- Order Country Customer Id
- Product Card Id
- Product Price
- Shipping Mode
- Order Item Quantity

2.3 Exploratory Data Analysis

Figure 19

Before building any model it's important to analyze the raw data and make appropriate changes if it needed. In the next paragraphs, we will review more in details those steps.

Our dataset contains 180519 transactions using different type of payment and 53 variables. Most of the variables are non-numerical and the rest are in float and integer data type. To continue we checked if there were any missing values. In Fig 1. we present that information in percentage. It is noticed that 2 variables **Customer Zipcode** and **Customer Lname** have less than 0% of missing data points meanwhile **Order Zipcode** and **Product Description** present a high number of nonexistent records 86% and 100% correspondently.

We wanted to see how correlated each variable were between them, most importantly with our target attribute. To analyze that, we plotted a correlation heatmap (See Fig 2.). We couldn't find any feature that has a strong correlation with our target. Apart from **Days for shipping (real)** all of them possess a low relationship. Other interesting detail we noticed is that some variables got a perfect positive relationship. Meaning that they showed similar values.

Based on a Box plot (See Fig 3.) some variables such as **Benefit per order, Sales per customer, Order Item Profit Ratio** exhibit a high amount of outliers while **Customer Id, Order customer Id, Department Id, sales** have a low amount.

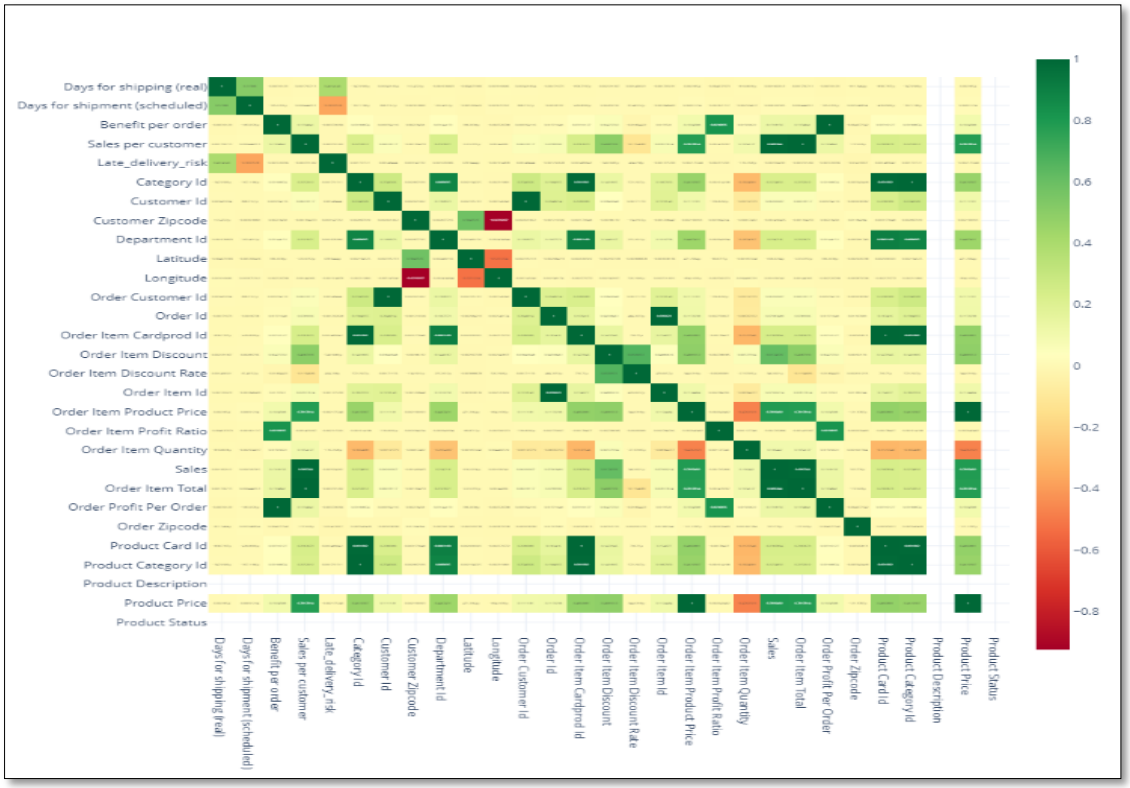


Figure 20:Correlation Heatmap

In order to see the distribution for each variable we plotted a histogram (See Fig 4.). Analyzing each plot, we could identify that some variables are right-skewed meaning that the extreme values affect the mean more than the median (Sales per customer, Order Item Discount, Order Item Total). Also, there were variables that are left-skewed, the median is more affected by the extreme values than the mean (Benefit per order, Order Item Profit Ratio, Order Profit Per Order). For others the distribution is approximately symmetrical meaning that the mean and median are equal. This is the case for Order Item Id which reveal a uniform distribution because there's no peak.

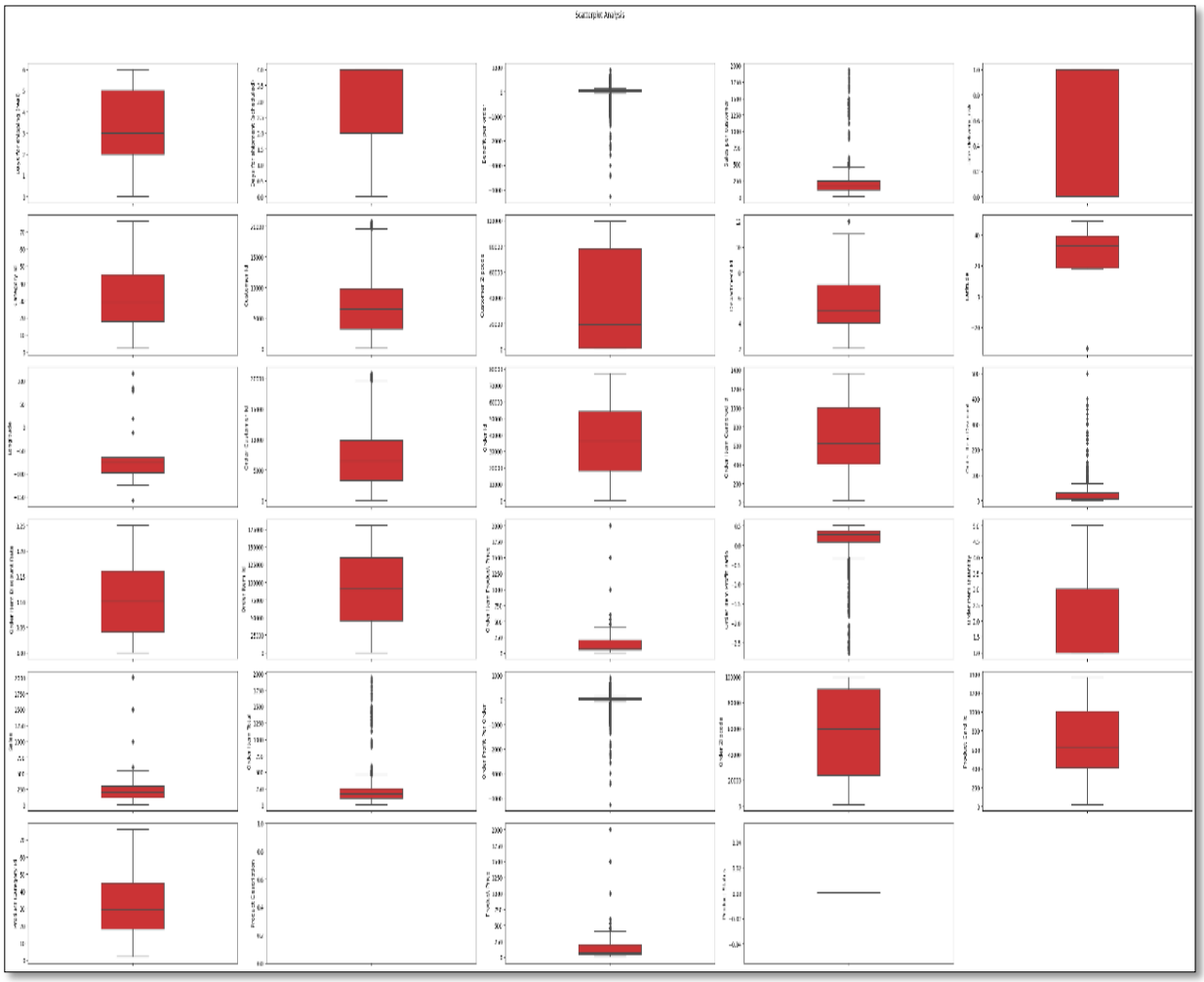


Figure 21: Box plot of the variables of the dataset

Figure 22: Histogram plot of each variable

To continue with the process of exploratory analysis, we went with a scatter chart to show the linear relation between our target and 3 other variables also between **Sales** and other variables. we can observe that (See Fig 5.) **Product Price, Order Item Product Price and Order Item Discount** have a positive linear relation with **Sales.** However, when we look at **Late_delivery_risk** there's no clear linear relation with the variables, because it doesn't cover the data points in the regression line.

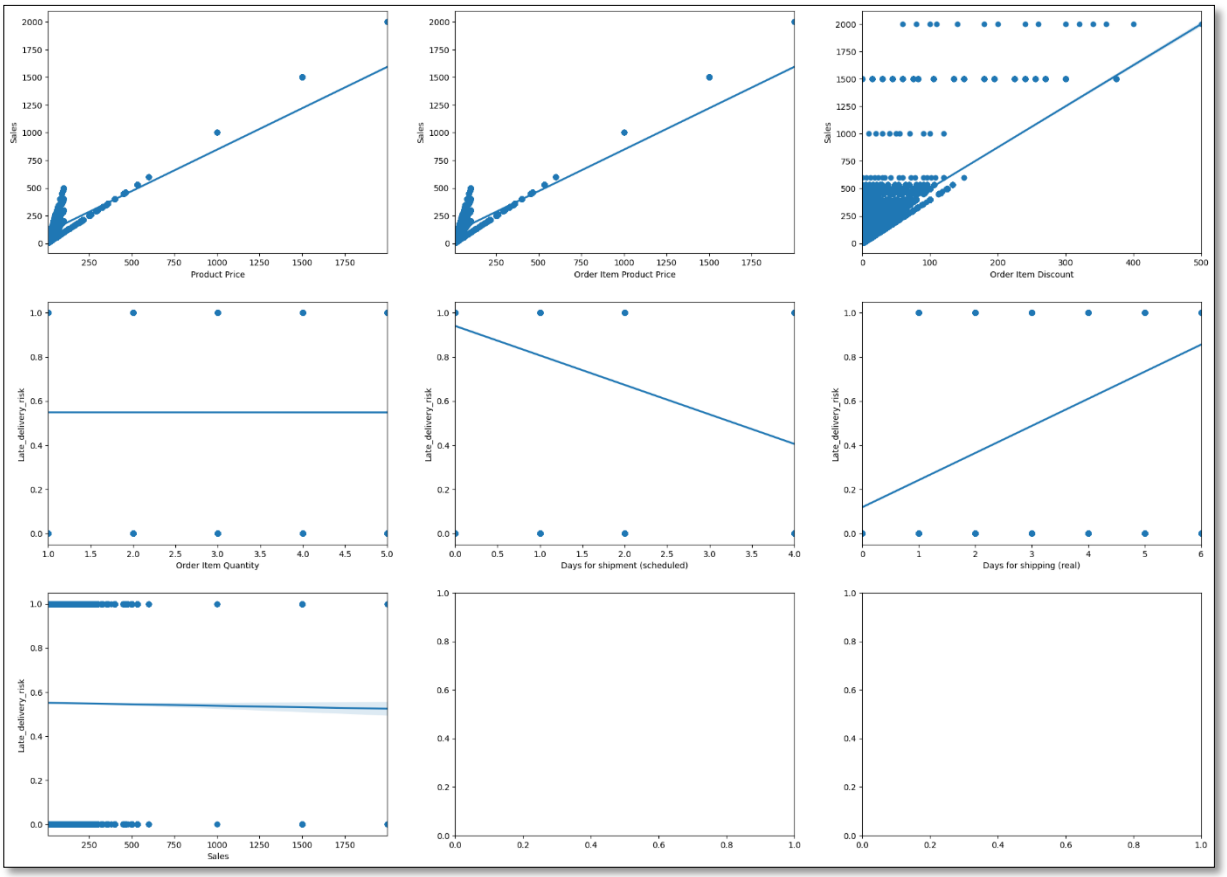


Figure 23: Scatter plot for sales and late delivery

2.4 Data Preprocessing

Now that we were done with the analysis, it was time to clean the dataset. First, we started with the missing values. As stated before, **Customer Zipcode** and **Customer Lname** comprise of a small number of missing values so we chose to drop the rows where those values were missing. Being that there were so many data points in **Order Zipcode** and **Product Description** both columns were dropped.

Product Status only had one unique value across samples, it was removed because it was non-informative and could have break the models.

Looking back at our heatmap graph (See Fig 23.) all the variables that have a Pearson correlation of 1 are identical (see the list below). So, in order to avoid harmful bias and make the learning algorithm faster we drop one between those 2 predictors and kept the other one.

Looking back at our heatmap graph (See Fig 23.) all the variables that have a Pearson correlation of 1 are identical (see the list below). So, in order to avoid harmful bias and make the learning algorithm faster we drop one between those 2 predictors and kept the other one.

Benefit_per_order and Order_Profit_Per_Order
Sales_per_customer and Order_Item_Total
Category_Id and Product_Category_Id
Customer_Id and Order_Customer_Id
Order_Item_Cardprod_Id and Product_Card_Id
Order_Item_Product_Price and Product_Price

We also renamed the following columns **order date (DateOrders) shipping date (DateOrders)** to **Order Date** and **Shipping Date** respectively just to make simpler to read and being that they were considered for the independent variables.

Being done with the cleaning part it was time to select the independent and dependent variables in order to build the model. For our model we took

Days for shipping (real), Days for shipment (scheduled), Order Country, Customer Id, Product Card Id, Product Price, Shipping Mode , Order Item Quantity, Type, as independent variables and **Late_delivery_risk** as the target variable.

As you can see some variables are categorical and for the machine learning to build the model those variables need to be converted in numerical.

In order to achieve that we used target encoding. This method consists in converting the categorical data points into the mean of the target variable in our case **Late_delivery_risk**. For each category it calculates the mean of the target variable and the value obtained replaced the categorical data point.
Furthermore, we used Category library to realize that as shown below.

```
from category_encoders.target_encoder import TargetEncoder

en_df=new_df.copy()
for col in en_df.select_dtypes(include='O').columns:
    te=TargetEncoder()
    en_df[col]=te.fit_transform(en_df[col],en_df.Late_delivery_risk)

en_df.head()
```

Figure 24 : Category library

Table 2: Reveal the output after encoding.

id	Days for shipping (real)	Days for shipment (scheduled)	Order Country	Customer Id	Product Card Id	Product Price	Shipping Mode	Order Item Quantity	Type	Late_delivery_risk
0	3	4	0.553784	20755	1360	327.75	0.380723	1	0.572175	0
1	5	4	0.561271	19492	1360	327.75	0.380723	1	0.485425	1
2	4	4	0.561271	19491	1360	327.75	0.380723	1	0.566323	0
3	3	4	0.542897	19490	1360	327.75	0.380723	1	0.572175	0
4	2	4	0.542897	19489	1360	327.75	0.380723	1	0.575332	0

2.5 Model Building and Result Description

At this stage we separate the dependent and independent variables in X and ***y.*** We split the dataset into training and testing set, using Sklearn library. We considered 75% of the dataset for training and the rest for testing.

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,
random_state = 0)
```

We started with multiple logistic regression algorithm; we first scaled the data using Sklearn then proceed to run the model into the training set. Then we predicted the test set.

In order to evaluate the model, we used 5 different types of metrics which are: Accuracy, Precision, Recall, F1 score and the Area Under Curve.

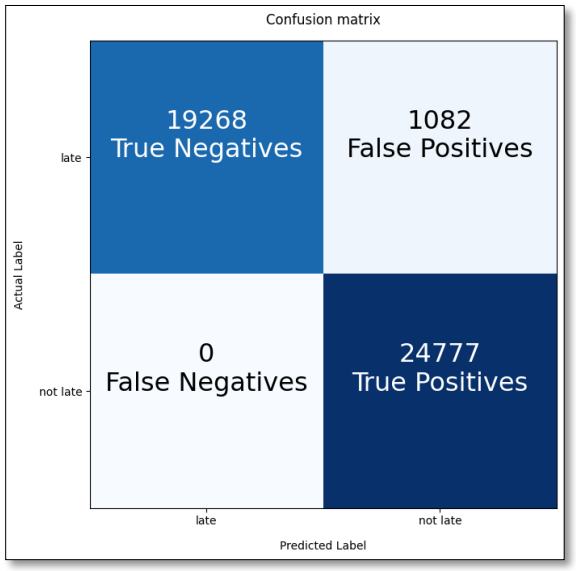


Figure 25: Confusion Matrix Logitstic Regression

Most of those metrics can also be obtained from the confusion matrix (See fig 28)

There are two class labels: late and not late.

The x-axis is labeled Predicted label - meaning our model's predictions given X_test.

The y-axis is labeled Actual label - meaning the class labels for our X_test observations.

The model made a total of 45127 predictions (equivalent to 24777+19268+0+1082) on if those 45127 shipments should be late or not

Out of the 45127 predictions, 19268 (equivalent to 0 + 19268) were for the late class.

Out of the 45127 predictions, 25859 (equivalent to 24777 + 1082) were for the not late class.

Let's explain each of the numbers in the boxes:

`For 19268 observations (from X_test) the model predicted late and the label was actually late.

For 0 observations (from X_test) the model predicted late but the label was actually not late.

For 1082 observations (from X_test) the model predicted not late but the label was actually late.

For 24777 observations (from X_test) the model predicted not late and it was actually not late.

Precision: for our predictions of the positive class (not late in our example), how often is it correct?

In sklearn, we used the precision_score method from the metrics module.

Precision: 0.958158

Recall: when the actual label is positive (not late in our example), how often did we predict correctly?

Recall: 1.0

In sklearn, we used the recall method from the metrics module.

F1 Score: combines precision and recall into one metric by calculating the mean between those two.

F1 Score: 0.978632

Accuracy: It measures how many observations, both positive and negative, were correctly classified.

Accuracy: 0.976023

Area under the ROC Curve (AUC): is the probability that the model ranks a random positive example more highly than a random negative example.

ROC AUC: 0.973415

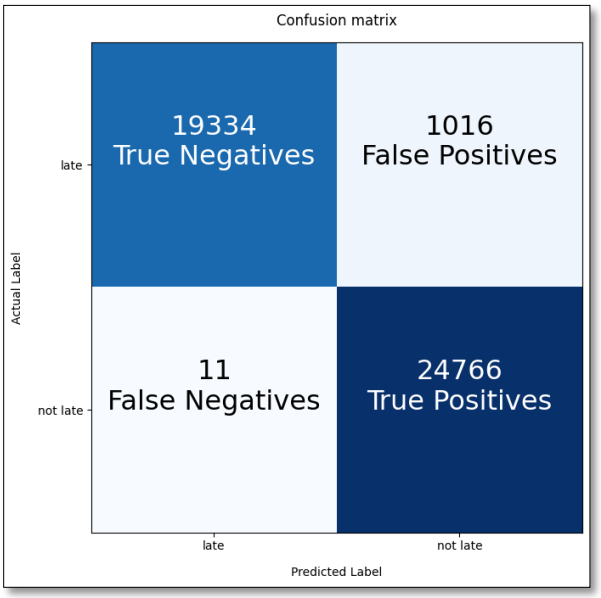


Figure 26: XG Booster

Next method was XGBooster . There's no need for feature scaling, however we did label encoding on the training dataset using scikit learn, then we run the model on training set. In Fig 29 we show the confusion matrix. It's the same interpretation that we did for Logistic Regression.

We got the following results

Table 3

Accuracy	Precision	Recall	F1 score	ROC AUC
0.975846	0.958074	0.999758	0.978472	0.973245

The final model used was Kernel SVM

Same as Logistic Regression we did Standard Scaling then run the model in the training set.

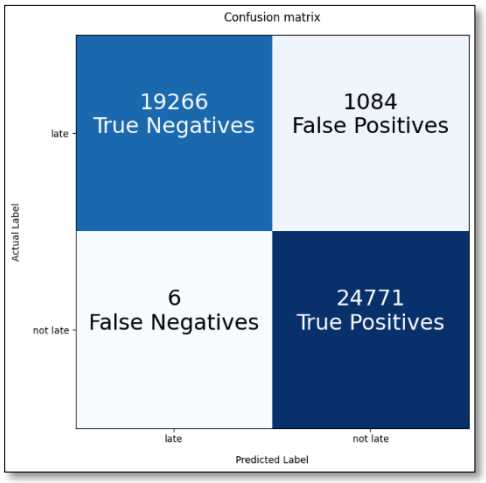


Figure 27: Kernel SVM Confusion Matrix

We got the following results for the metrics.

Table 4: Kernel SVM Results

Accuracy	Precision	Recall	F1 score	ROC AUC
0.975846	0.958074	0.999758	0.978472	0.973245

Table 5: Comparison Table

Models	Accuracy	Precision	Recall	F1 score	ROC AUC
Logistic Regression	0.976023	0.958158	1.0	0.978632	0.973415
Kernel SVM	0.975846	0.958074	0.999758	0.978472	0.973245
XGBooster	0.977242	0.960593	0.999556	0.979687	0.974815

Based on Table 5 we can say that XGBooster performed the best in all metrics except Recall, where Logistic Regression shine with a score of 1.0. While Kernel SVM performed the worst but still a good score in general.

2.6 Discussion

It have been observed that when compared with other classification machine learning models XGBooster did a good job of identifying orders with later delivery. For further study, all the machine learning models can be compared with different datasets to confirm whether the same machine learning models are performing better or not. We could also try to tune the parameters in order to get a better result. Something interesting would be to combine different features to see if we could boost the performance.

2.7 Business and Technological Implication

With the help of the current analysis, businesses can gain valuable insights into their operations, customers, and markets. This information can be used to inform strategic decisions, identify potential areas of improvement, and develop effective marketing strategies. Additionally, businesses can use the analysis to determine customer needs and preferences, optimize pricing and product offerings, and develop new products and services. By leveraging the data gathered through analysis, businesses can gain a competitive edge and increase their profitability based on the following:

Firstly, by carefully analysing the right times for marketing and advertising, the declining revenue can be increased. This analysis can help to determine the best times to reach out to potential customers and create more awareness of the product or service. With the right marketing strategy, businesses can gain more customers and increase their revenue.

Secondly, through the investigation of the late delivery reasons with classification model, it has been found that the standard type of shipment and the payment methods are the two primary causes of late shipments. This is due to the fact that the standard type of shipment is not as efficient as the express type, and the payment methods can cause delays in the processing of the order. As a result, the customer's order may not be delivered on time. To address this issue, companies must remodel their internal processing by improving the promised delivery date more accurately so that the actual delivery shall not vary from that. Moreover, company cannot rely on the express shipment considering the sustainability and Co2 emission factors. Hence, less Late deliveries results in customer satisfaction

Lastly, it has been observed that the majority of deliveries are getting delayed due to the debit card payment method. This is because it takes time for the payment to be processed in the system. To tackle this issue, businesses can include more payment methods such as Paypal or Klarna, which can increase customer ease of access and ultimately improve business , in turn, improve the business. By doing this, customers can have more faster payment options, thus leading to an improved delivery process and a better customer experience.

2.8 Business Challenges

Data Co. can face a variety of challenges when implementing the given solutions to their problems. These challenges can include difficulty in understanding the technical aspects of the solution, a lack of resources to properly implement the solution, or an inability to effectively communicate the solution to stakeholders. Additionally, it may struggle to stay within budget since its revenue is already declining. With fewer resources available, it must make difficult decisions about where to allocate their limited funds. This could mean cutting back on employee benefits, reducing staff, or reducing the amount of money spent on marketing and advertising. It could also mean reducing the amount of money spent on research and development or cutting back on the number of products and services offered.

Furthermore, DataCo. must also consider how to best manage their cash flow. In addition, companies must consider how to best use their resources to ensure their long-term success because in order to remain competitive in their industry, as well as to adapt their existing processes to the new solution. Finally, there may be resistance from employees to the new solution, which could impede the success of the implementation. All of these challenges can be difficult to overcome, but with the right preparation and resources, businesses can successfully implement the solutions they need.

However, considering the shipment method if Data Co. focus solely on same day delivery to overcome late delivery issues, this might not be beneficial in the long run due to the Sustainable Development Goals (SDGs). It must take them into consideration when making decisions about their operations. Same day delivery may be beneficial in the short-term, but in the long-term, businesses must consider the impact that their operations have on the environment and society. Therefore, focusing only on same day delivery to address late delivery issues may not be the most sustainable solution in the long run but adjusting their internal timeline can be.

Lastly, company is doing business in many countries in all continents, it is important to consider the payment partners. In some countries, certain payment partners may not be available, such as Paypal. This means that one must look for alternative payment partners in order to ensure that customers in those countries can make payments without any issues. Therefore, it is important to research the payment partners available in each country before deciding which ones to partner with.

2.9 Business Model Canvas

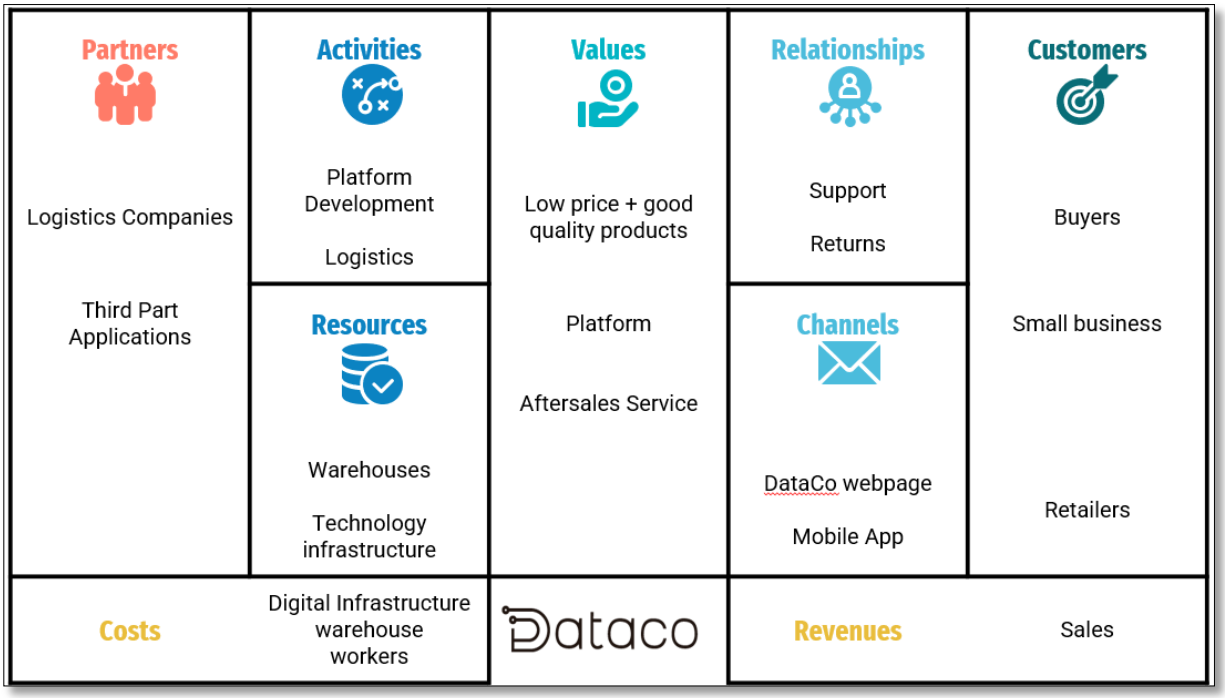


Figure 28 Bussiness Model Canvas

3 Supply Chain Scenario 3: CNC-Milling Machine_Production data

3.1 Scenario Description

The given dataset is a dataset of the power consumption of a programmable CNC milling machine with the corresponding process features. A row represents a process with its properties, timestamps and power consumption

3.2 Dataset Description

Table 6 CNC Machine dataset

Feature	Description
start_time	Timestamp (UTC) when the process starts.
end_time	Timestamp (UTC) when the process finishes.
processing_time	Duration of the process [s].
average_power_consumption	Average power consumption during the process [Watt].
number_of_missing_datapoints	Number of missing power measurement points during the process. The power data is collected with 1 minute interval.
raw_volume	Volume of the processed material [mm3].
number_of_lines_of_code	Number of lines of G/NC code that execute the process.
number_tool_changes	Number of tool changes during the process, e.g. drill head -> milling head or larger drill head -> smaller drill head. It does not refer to replacement through wear, just different tools for different tasks.
number_of_travels_to_machine_zero_point_in_rapid_traverse	Number of travels/movements of the tools from the initial state (zero point of the machine's coordinate system) to rapid repetitive/traverse process.

number_axis_rotations	Number of changes in the position of the axis of rotation of the tool (e.g. drill head).
speed	Rotation speed of the tool, e.g. drill head [1/min]
tool_diameter	Diameter of the tool used for a production step [mm].
cutting_length	Length of the cutting edges of the tool used for a production step [mm].
number_of_cutting_edges	Number of cutting edges of the tool used for a production step.
cutting_speed	Speed at which the cutting edge of the tool used for a production step cuts into the material [m/s].
feed_per_tooth	Portion of distance during feed into the component per cutting tooth of the tool used for a production step per revolution [mm].
feedrate	Speed at which the tool cuts through/into the material in [mm/min].

3.2.1 **Dependent Variables:** Processing time and average power consumption

3.2.2 **Independent Variables:**

Same as mentioned in the data description except Processing time and average power consumption.

3.3 Exploratory Data Analysis

In order to understand the structure of the dataset and make the machine learning process easier and leads to better result we used statistics and visualization to analyze and identify trends in data sets.

First looking at the shape of the dataset we have 220 observations and 17 columns. All the variables are numerical values which is convenient reducing on the preprocessing task (Fig 29).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 220 entries, 0 to 219
Data columns (total 17 columns):
 #   Column                                                Non-Null Count  Dtype
---  ------                                                --------------  -----
 0   start_time                                            220 non-null    int64
 1   end_time                                              220 non-null    int64
 2   processing_time                                       220 non-null    int64
 3   average_power_consumption                             220 non-null    int64
 4   number_of_missing_datapoints                          220 non-null    int64
 5   raw_volume                                            220 non-null    int64
 6   number_of_lines_of_code                               220 non-null    int64
 7   number_tool_changes                                   220 non-null    int64
 8   number_of_travels_to_machine_zero_point_in_rapid_traverse  220 non-null  int64
 9   number_axis_rotations                                 220 non-null    int64
 10  weighted_speed                                        220 non-null    int64
 11  weighted_tool_diameter                                220 non-null    int64
 12  weighted_cutting_length                               220 non-null    int64
 13  weighted_number_of_cutting_edges                      220 non-null    int64
 14  weighted_cutting_speed                                220 non-null    int64
 15  weighted_feed_per_tooth                               220 non-null    int64
 16  weighted_feedrate                                     220 non-null    int64
dtypes: int64(17)
memory usage: 29.3 KB
```

Figure 29: Statistic about the data type of the dataset.

Going on with the analysis we used different visualization to help us see any relation, trends between the target variables and independent variables.

```
start_time                                                    0
end_time                                                      0
processing_time                                               0
average_power_consumption                                     0
number_of_missing_datapoints                                  0
raw_volume                                                    0
number_of_lines_of_code                                       0
number_tool_changes                                           0
number_of_travels_to_machine_zero_point_in_rapid_traverse     0
number_axis_rotations                                         0
weighted_speed                                                0
weighted_tool_diameter                                        0
weighted_cutting_length                                       0
weighted_number_of_cutting_edges                              0
weighted_cutting_speed                                        0
weighted_feed_per_tooth                                       0
weighted_feedrate                                             0
dtype: int64
```

Figure 30: Statistic about missing values

Going on with the analysis we used different visualization to help us see any relation, trends between the target variables and independent variables.

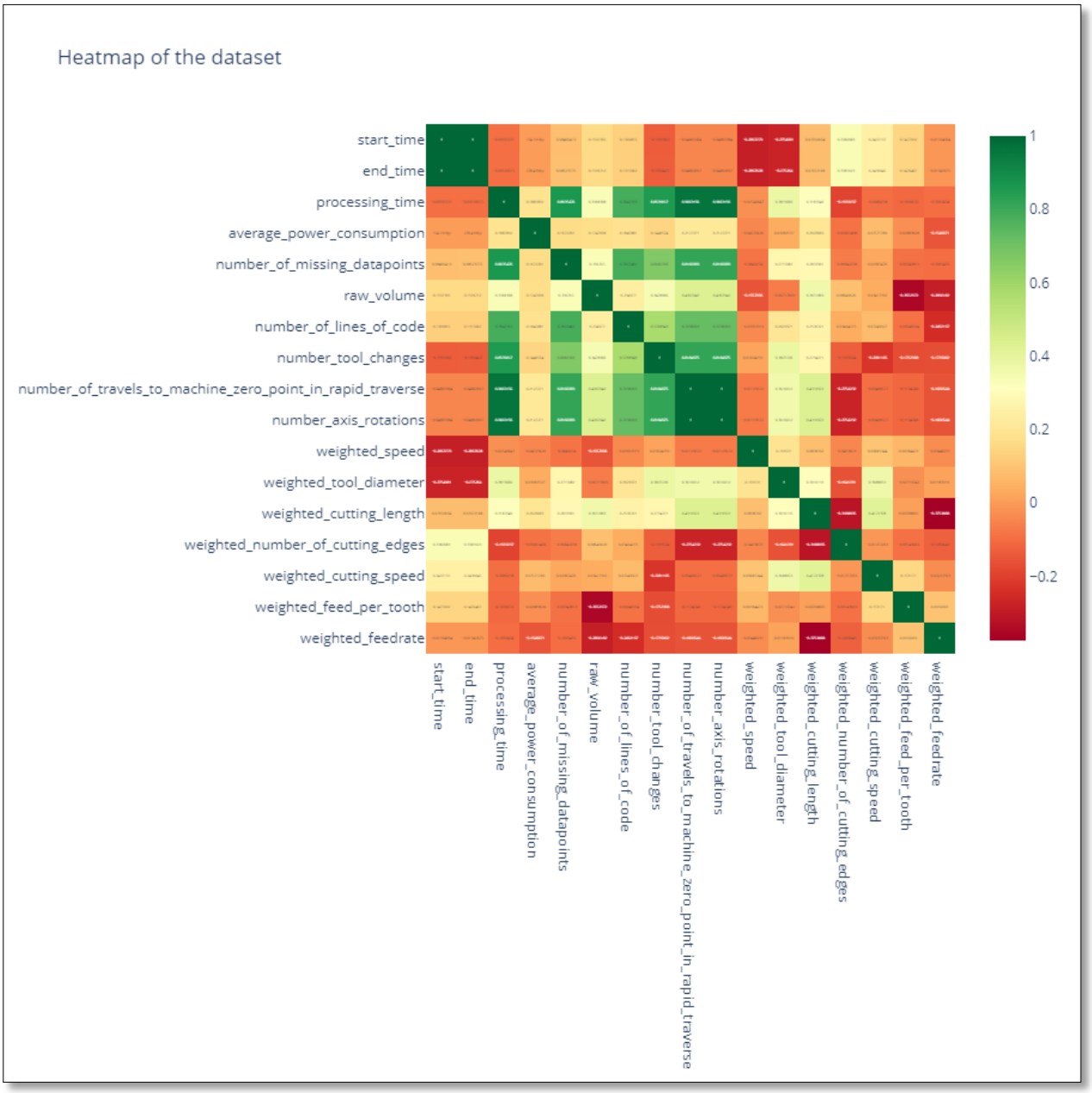


Figure 31: Correlation heatmap of every variable.

Initially we checked the correlation between the attributes using correlation heatmap

Looking at the figure (Fig 31) We can observe that **start_time** and **end_time** have a perfect positive relationship, same for **number_of_travels_to_machine_zero_point_in_rapid_traverse** and **number_axis_rotations** taking into account the dependent variable we can say that **processing_time** has a strong positive relationship with **number_axis_rotations**,

number_of_travels_to_machine_zero_point_in_rapid_traverse, number_tool_changes and **number_of_missing_datapoint**. Meanwhile **average_power_consumption** has a weak correlation with all the other variables.

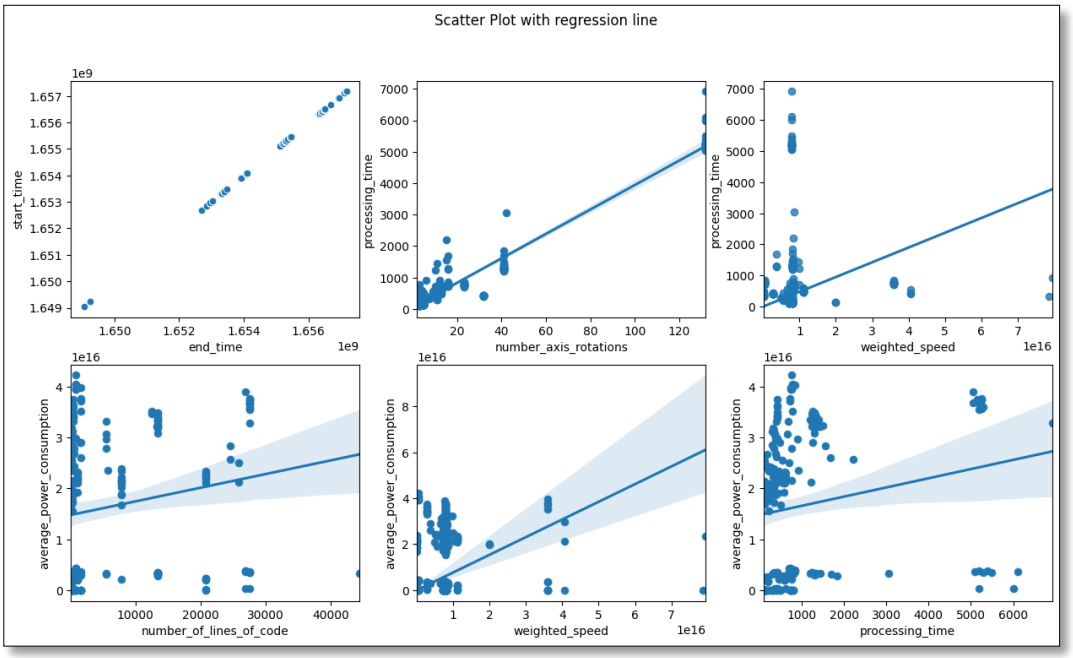


Figure 32: Scatter plot of the target variables against some predictors.

Here we notice that there's a linear relationship between **processing_time number_axis_rotations** and **weighed_speed**. However, there's no linear relationship between **average_power_consumption weighed_speed**, **processing_time** and **number_of_lines_of_code.** Finally start_time and end_time are completely linear.

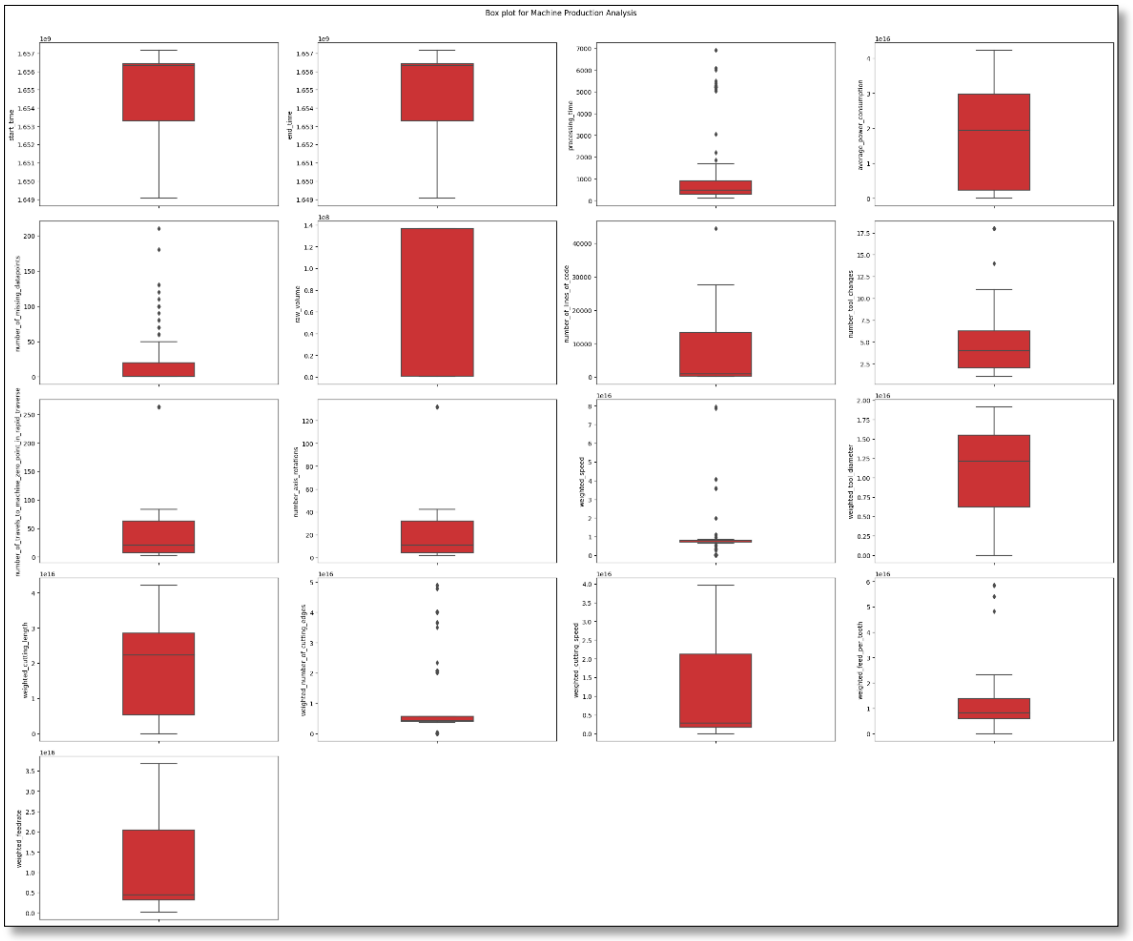


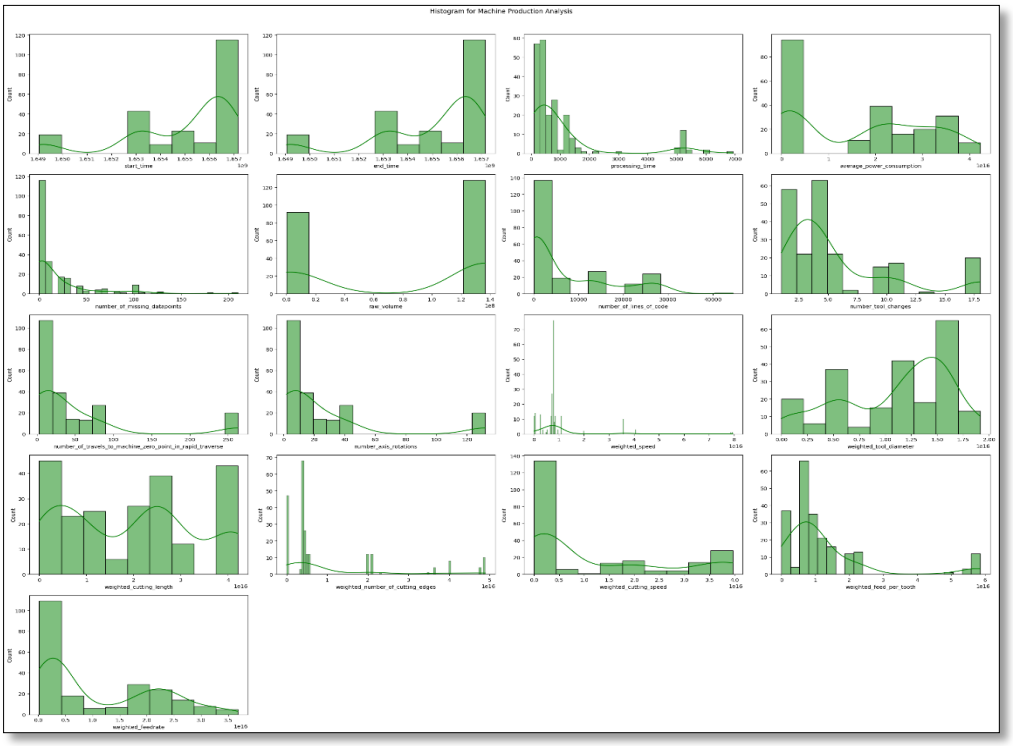
Figure 33: Box plot of the independent and dependent variables.

It can be seen that some variables present some outliers like **weighted_speed, weighted_number_of_cutting_edges, number_of_missing_data_points**. Also values in **raw_volume** doesn't vary that much.

Histogram Plot

We wanted to see the distribution for some variables so we used histogram plot for that

In (**Fig 34**) some variables are right skewed like **number_missing_data_points**, **weighted_feedrate**, **processing time**. **start_time** and **end_time** are left skewed. There are no variables with normal distribution. **Weighted_cutting_length** is non-symmetric bimodal distribution.



Line Plot

We intended to see how **average_power_consumption** would change during those 4 months

Figure 34: Histogram

from April to July. There were no clear patterns but the 2 highest energies consumed was on May and June.

Figure 35: line plot to see evolution of average power consumption

Bar plot

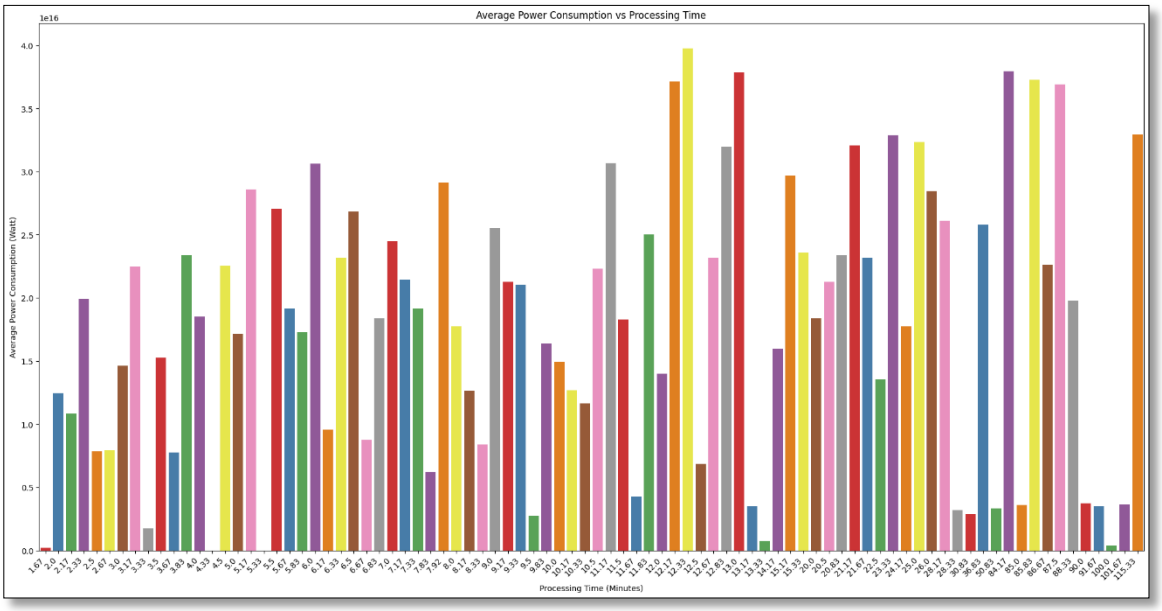


Figure 36: Bar Plot between average power consumption and processing time

With a bar plot we tried to look if the energy would raise with the increasement of time. However, it wasn't the case as you can see in Fig 36.

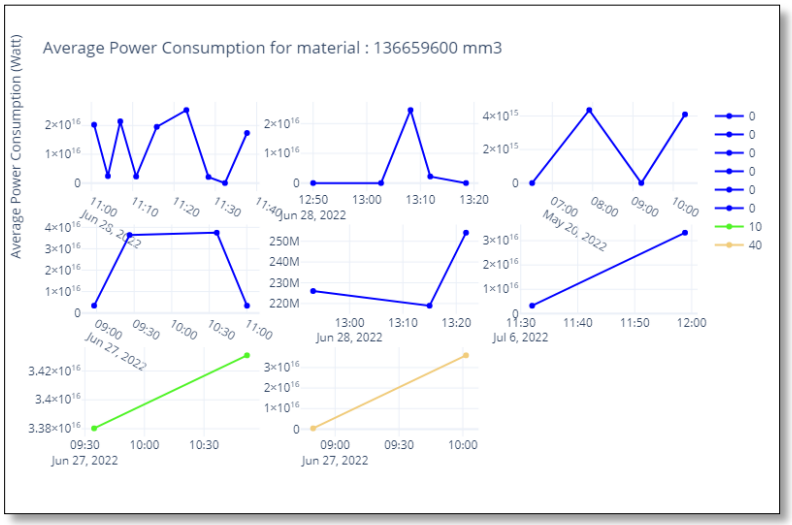


Figure 37: Average power consumption over time for the same material of 136659600 mm3

around 14 minutes it produced less than 0.5 x 10^5 Watts and in only 2 minutes it produced more than 1 x 10^5 Watts which is more in less time. Now we took a different approach by grouping same material with the exact same parameters and same time. Still there was no clear relation between the energy over time as we can see in Fig 37.

We have gained enough insights with those plots and our data was ready for modeling.

3.4 Data Preprocessing

Being that the data was already pre-processed there wasn't that much to do in this area however we did some pre-processing step in the next section by dropping some columns after doing feature importance.

3.5 Model Building and Result Description

We first decided to do the modeling by considering all the variables.

After selecting the independent and target variables. We split the dataset into training set and test set.

The first model we used was Random Forest. With RF there was no need for feature scaling, so we went straight on training the model into the training set and selecting 10 for the number of trees. We used mutli-target regression to predict both targets simultaneously. The method was done using scikit learn.

```
# Training the Random Forest Regression model on the whole dataset
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators = 10, random_state = 0)
regressor.fit(X_train, y_train)
```

In order to evaluate our model, we used 4 different type of performance metric. Which were R square, Mean square error (MSE) and Root mean square error (RMSE) and Mean absolute error (MAE)

R-squared determines how strong the linear relationship is between two variables. It's represented by value between 0 and 1, where 1.0 indicates a perfect fit for the model meanwhile a value of 0.0 would indicate that it didn't correctly model the data (Andrew, 2021).

$$R^2 = (\frac{1}{N})\frac{\sum[(x_i - \bar{x}) \times (y_i - \bar{y})]}{(\sigma x \times \sigma y)^2} \quad (1)$$

Mean square error (MSE) calculate the average squared distinction between the actual and predicted values. The larger the number the larger the error. (Jim, n.d.)

$$MSE = \frac{1}{N}\sum_{i=1}^{b} (y_i - \bar{y}_i)^2 \qquad (\mathbf{2})$$

Root mean square error is calculated as the square root of the average squared distance between the actual and the predicted values (Zack, 2021)

$$RMSE = \frac{1}{N}\sqrt{\sum_{i=1}^{b} (y_i - \bar{y}_i)^2} \qquad (\mathbf{3})$$

Mean absolute error (MAE) score is measured as the average of the absolute error values.

$$MAE = \frac{1}{N}\sum_{i=1}^{b} |y_i - \bar{y}_i| \qquad (\mathbf{4})$$

Using those metrics, we obtained the following result for Random Forest

Table 7: Different type of performance Metrics for Random Forest

Model	**R-2- 1**	**R- 2**	**MAE 1**	**MAE 2**	**MSE 1**	**MSE 2**	**RMSE 1**	**RMSE 2**
Random Forest	0.956491	0.139179	181.719697	9.396825e+15	156558.553030	1.515796e+32	156558.553030	1.515796e+32

Next model was Decision Tree. Same as Random Forest we split the data set into training and set and test set and run the multi-target regression on both dependent variables. We obtained the following result after evaluation.

Table 8: Different type of performance Metrics for Decision Tree

Model	R-square 1	R-square 2	MAE 1	MAE 2	MSE 1	MSE 2	RMSE 1	RMSE 2
Decision Tree	0.931603	-0.130642	191.818182	9.419117e+15	246113.636364	1.990918e+32	246113.636364	1.990918e+32

Lastly, we used Support Vector Regression. After splitting into training and test set. We proceed with feature scaling and then run the model in the training set by using a multi-output regression. In the following table we can see the result we obtained.

Table 9: Different type of performance Metrics for Support Vector

Model	R-square 1	R-square 2	MAE 1	MAE 2	MSE 1	MSE 2	RMSE 1	RMSE 2
Support Vector	0.947969	0.117954	212.293465	8.414137e+15	187223.166361	1.553171e+32	187223.166361	1.553171e+32

So far, we can say for processing time Random Forest performed the best among all models and metrics, support vector comes in second place with Decision Tree as worst. But overall, they all give a good performance.

For **average_power_consumption** Random Forest also performed the best among all models however it still a really poor score, same for the other two models.

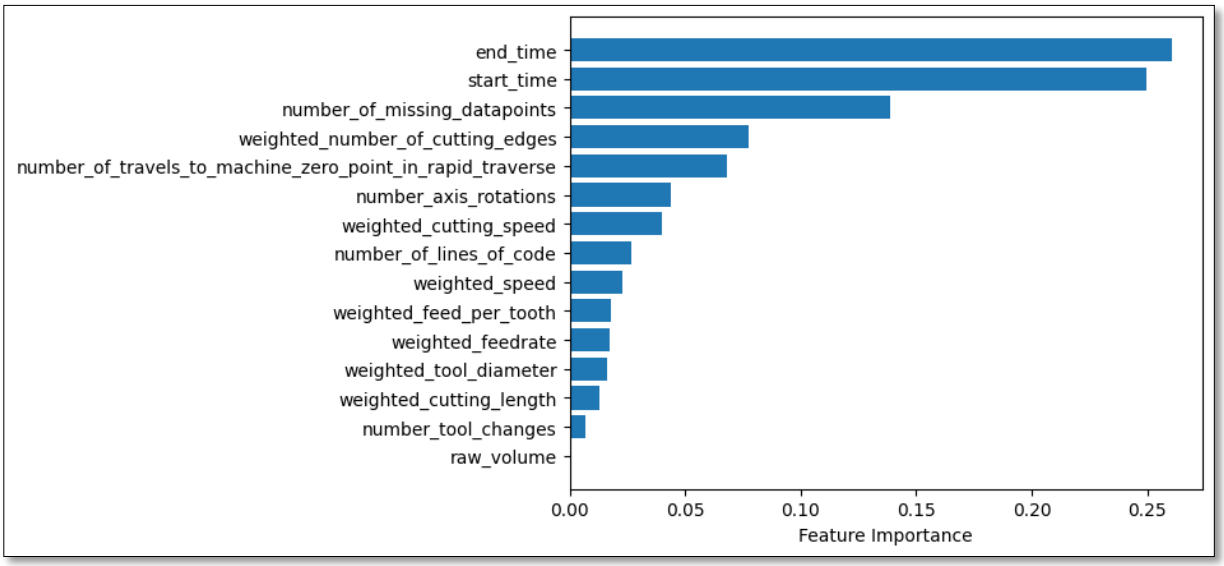


Figure 38: Feature Importance

At this stage we did feature importance, we can see that **raw_volume** doesn't bring too much to the target variables. So, we decided to drop it. Also **end_time** and **start_time** present similar values analyzing the correlation heatmap. Saying that we drop end_time.

With those 2 columns dropped we proceed with the whole process of modeling. In table 4 we show the results for the metrics.

Table 10: Different type of performance Metrics for all models after cleaning

	R-square 1	R-square 2	MAE 1	MAE 2	MSE 1	MSE 2	RMSE 1	RMSE 2
Random Forest	0.961658	0.124216	167.436364	9.522955e+15	137965.578182	1.542144e+32	137965.578182	1.542144e+32
Support Vector	0.938621	0.129229	221.311828	8.574613e+15	220859.223265	1.533317e+32	220859.223265	1.533317e+32
Decision Tree	0.931729	-0.130443	189.545455	9.404719e+15	245659.090909	1.990567e+32	245659.090909	1.990567e+32

Table 11: Different type of performance Metrics for all models before cleaning

	R-square 1	R-square 2	MAE 1	MAE 2	MSE 1	MSE 2	RMSE 1	RMSE 2
Random Forest	0.956491	0.139179	181.719697	9.396825e+15	156558.553030	1.515796e+32	156558.553030	1.515796e+32
Support Vector	0.947969	0.117954	212.293465	8.414137e+15	187223.166361	1.553171e+32	187223.166361	1.553171e+32
Decision Tree	0.931603	-0.130642	191.818182	9.419117e+15	246113.636364	1.990918e+32	246113.636364	1.990918e+32

Comparing Table 10 and Table 11 we can say that after doing the cleaning only Random Forest gained a considerable improvement. However, Support Vector went down in performance and came at last behind Decision Tree. There was minimum improvement for Decision Tree but not really impactful. Based on that we can say that there isn't that much difference in performance after cleaning.

3.6 Discussion

Using a regression model for a CNC machine can make it easier to predict which product will have a higher processing time. This prediction can help to identify which products may require more resources or more time to process, allowing for better scheduling and planning. By having this information, it can help to reduce the amount of time and resources spent on processing each product, resulting in a more efficient and cost-effective production process.

Moreover, with the right data and analysis, predicting power consumption can be a valuable tool for optimizing energy consumption. By understanding the power consumption of CNC machines, their timings can be adjusted to take advantage of dynamic power rates. This can help to reduce energy costs and ensure that the machines are running efficiently. Additionally, understanding the power consumption of CNC machines can help to identify potential problems before they become major issues, allowing for proactive maintenance and repairs.

3.7 Business and Technological Implication

In order to decrease processing time, code optimization is required. This means that the number of travels should be reduced, which will result in axis rotation optimization. By optimizing the code and reducing number of line of code for short time, the processing time can be significantly reduced, making the program more efficient. Moreover, when purchasing a machine, it is important to consider the various models available and determine which one is best suited for the task. Applying this type of ML model can help in making an informed decision. Different machines may have different features, so it is important to compare the models to determine which one is the most suitable. By using this type of ML model, one can compare the features of different machines and select the one that is most suitable for the task.

3.7.1 Business Disruptions:

CNC machines are becoming increasingly popular in the manufacturing industry, as they can provide a more efficient and cost-effective production process. They are much easier to program than NC (Numerical Control) machines, as changes can be made directly in the program line. This is not possible with NC machines, as editing is not possible. This makes CNC machines much more flexible and easier to use than NC machines. Moreover, they are able to produce parts with a higher level of accuracy and precision than traditional NC machines, which leads to lower production costs. This cost savings is one of the main reasons why CNC machines are replacing standard NC machines in many industries. Also, they offer a higher degree of automation, which makes them more efficient and reduces the need for manual labour. This leads to further cost savings, which makes CNC machines a more attractive option for businesses looking to reduce their production costs.

4 Conclusions Future Outlook, and Reflection

We have gained a lot of insight and understanding from the combination of supply chain and data analytics. By combining the two, we are able to gain insights into the entire supply chain process, from the problem statement, identify areas of improvement and providing solutions. Through this research we understood that data analytics can help us identify trends, problems, and opportunities for improvement, allowing us to make better decisions and create more efficient supply chain processes. Additionally, it can help us better understand customer needs and preferences, allowing us to create better products and services. Overall, the combination of supply chain and data analytics has been invaluable in helping us gain a better understanding of our operations and how to make them more efficient.

Lastly, We have learned through this course that Machine Learning (ML) models are powerful tools that can be used to achieve a variety of results. Depending on the task at hand, different ML models can be used to achieve the desired outcome. For example, if the goal is to classify data, a supervised learning model such as a Support Vector Machine (SVM) or a Decision Tree can be used. If the goal is to predict a continuous value, a regression model such as Linear Regression or a Random Forest can be used. For clustering tasks, an unsupervised learning model such as K-Means or Hierarchical Clustering can be used. Additionally, for tasks such as natural language processing (NLP), a deep learning model such as a Recurrent Neural Network (RNN) or a Convolutional Neural Network (CNN) can be used. No matter the task, there is likely an ML model that can be used to achieve the desired result.

Bibliography

Constante, F., Silva, F., Pereira, A., 2019. DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS 5. https://doi.org/10.17632/8gx2fvg2k6.5

Frost, J., 2021. Mean Squared Error (MSE). Stat. Jim. URL https://statisticsbyjim.com/regression/mean-squared-error-mse/ (accessed 12.10.22).

Online retail dataset [WWW Document], n.d. URL https://www.kaggle.com/datasets/lakshmi25npathi/online-retail-dataset (accessed 12.10.22).

payment not received icon - Google Search [WWW Document], n.d. URL https://www.google.com/search?q=payment+not+received+icon&tbm=isch&chips=q:payment+not+received+icon,online_chips:payment+failure:G8_UhLinejc%3D&hl=en&sa=X&ved=2ahUKEwiE9vmz3tP7AhVthP0HHXcGBhcQ4lYoBHoECAEQLQ&biw=1519&bih=722#imgrc=kJ0EWH4Kie5iIM (accessed 11.29.22).

Schneider, P., Xhafa, F. (Eds.), 2022. Contents, in: Anomaly Detection and Complex Event Processing over IoT Data Streams. Academic Press, pp. vii–xii. https://doi.org/10.1016/B978-0-12-823818-9.00005-5

Schoppe, I., 2020. Heute im Sonder-Livestream: So baust du 2021 dein Online Business auf. Gründer.de. URL https://www.gruender.de/gruendung/sonder-livestream-online-business/ (accessed 11.28.22).

Zach, 2021. How to Interpret Root Mean Square Error (RMSE). Statology. URL https://www.statology.org/how-to-interpret-rmse/ (accessed 12.10.22).