2019

Data Analytics for Supply Chain Management

MACHINE LEARNING APPLICATIONS IN E-COMMERCE, DELIVERIES & PRODUCTION

ATIT BASHYAL, TANASORN CHINDASOOK, JANDRA FISCHER, HAMZA INTISAR, MARK KOERNER, PETER-SLEIMAN MANSOUR

JACOBS UNIVERSITY BREMEN

28. NOVEMBER 2019

Contents

List of Figures

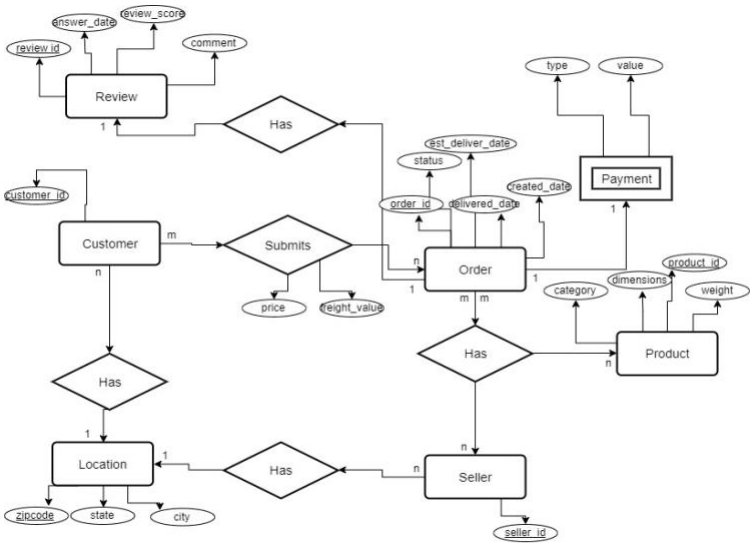
1. Introduction

In the context of supply chain and in order to get familiar with supply chain related datasets, we were asked to develop use case scenarios based on certain open dataset. The first step of this project was to propose datasets that were publicly available, easy to understand, interesting to work on, challenging to analyze and concern real world scenarios. Therefore, and in order to be able to cover more than one topic of the supply chain, different analysis methods and various scenarios three datasets were chosen from the online platform Kaggle. While the first dataset covers the E-commerce business and shells data regarding sales. The second dataset lists details about cashew truck deliveries and focuses on the procurement part of the supply chain. Finally, the third dataset covers the production topic and catalogues real world data obtained from a floatation plant. This report is divided into three sections, in which we will take an extended look at each dataset, put it into the right context and attempt to provide solutions.

2. E-Commerce Data Analytics: OList Brazil

Olist is a Brazilian e-commerce platform founded in 2015 that sells a wide variety of products from different shops on the main online marketplaces in Brazil. The dataset comes from the Kaggle website and concerns the sales part of Olist's Supply Chain. It lists data of more than 100,000 orders made in the years 2016 until 2018 and contains a total of 9 tables. The table descriptions for the Olist dataset can be found in Appendix 1.

In total, these 9 tables contain 51 variables. However, some variables are duplicates in different tables. For instance, all three variables Zip Code, City and State appear in the three tables Customers, Sellers and Geolocation. This poses a data integrity issue in storage, as the values in all three tables would have to be updated if a seller or customer changed locations. A suggested improvement is that each location be stored by a unique identifier (location ID) and the duplicate locations in the other tables should be referred to by their location ID, thus creating a foreign key reference in both tables instead of duplicate data issues. The general ER diagram for the Olist dataset is shown in Figure 2.1.



1 Figure 2.1 ER diagram for the Olist dataset

.

2.1. Supply Chain Context and Relevant Features

The Olist dataset can be applied to multiple Supply Chain scenarios. As the tables include different aspects, such as freight performance, prices, order status, customer and seller locations, product attributes and order reviews, one can analyse a variety of scenarios. The following sections describe how various methods of data analysis on the Olist dataset can be beneficial in the Supply Chain context.

2.1.1. Demand Forecasting

Sales data is often used to forecast demand, as it aims to estimate based on a model the number of products sold to customers, considering different factors, such as product type, customer and region[1]. Using the Olist dataset, one could forecast demand with a linear regression model or a decision tree. Possible scenarios could be to forecast sales of products for different seasons, for example the sales for the period before Christmas. Also, it would be possible to forecast the demand per state, city, seller or customer. Relevant attributes from the dataset can be, depending on the exact scenario, price, order_purchase_timestamp, customer_city, customer_id, customer_state or seller_id.

2.1.2. Market Basket Analysis (Association Mining)

With Market Basket Analysis sellers try to understand which products are bought together. This is especially useful for industries that carry a large amount of products where links are not very obvious[2]. Not only does this help marketers promote the right products together, but it is also helpful for the Supply Chain department, as they understand for example which products are often shipped together.

For Olist's dataset it can be seen that there are orders containing two or more products. It can therefore be analysed in order to determine which products or product categories get often sold together. Relevant attributes are therefore order_item_id, product_id and product_category_name.

2.1.3. Customer Segmentation (Clustering)

Businesses can use clustering to create customer segments to tailor their marketing strategies to each segment[3]. With the Olist dataset, we could cluster our customers based on the location (customer_city, customer_state), the seller they buy from or the location of the seller (seller_id, seller_city/state), the price or freight cost of their orders, the review we receive from them (review_score), the types of product they buy from Olist (product_category_name), payment type or delivery time (order_delivered_customer_date minus order_purchase_timestamp).

2.2. Scenario Development

Currently, many customers only buy one product when ordering through Olist as 90% of all orders contain only one product. This information indicates that there is significant potential in advertising associated products together to increase sales. Not only can this increase the overall revenue, but also reduce costs per product in transaction costs, freight costs or packaging costs. In order to achieve this goal, complementary products that are frequently bought together by other

[1] (Islek & Ögüdücü, 2015)
[2] (Blattberg, et al., 2008)
[3] (Carnein & Trautmann, 2019)

customers should be advertised once an item is added to the basket. Therefore, a market basket analysis or association mining is conducted on our past orders that contained two or more products.

Initially, category-wise association mining will be performed on the 72 product categories to see whether overall synergies exist between categories. Subsequently, a second market basket analysis will then be conducted on the individual products between pairs of categories with hidden relationships. This implementation of product-wise association mining is due to the fact that there are a total of 32,951 unique products available on Olist, which would cause a row overflow in computation if all products were considered in the market basket analysis. Therefore, we have chosen to only conduct association mining for products within categories that contain hidden synergies as a workaround to this limitation. However, if better hardware is available, one should conduct association mining across all 32,951 unique products for optimal results.

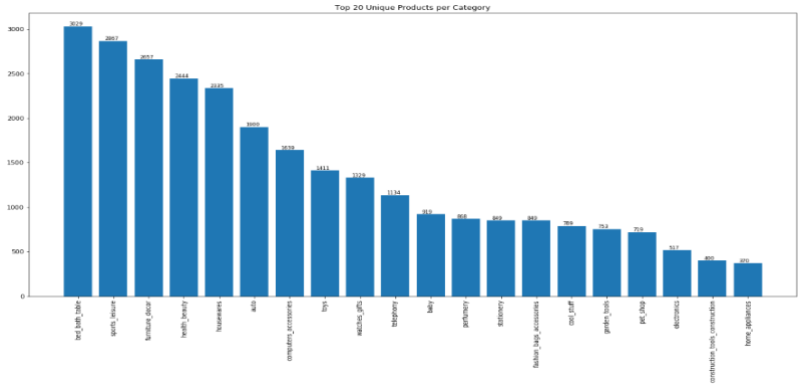
As a result, Olist will be able to market randomly selected products from associated categories or specifically associated products to a customer that is shopping with Olist, leading to more sales per customer order. The analysis will also help the Supply Chain department to understand which products are sold together and hence could be shipped together. It can also help to get the Supply Chain activities ready, once these new advertisements for relevant products start.

2.3. Data Exploration and Preprocessing

Data exploration and pre-processing often occur before performing data analysis. The first step of data pre-processing is to translate the category names, as the data is originally in Portuguese. A translation map is created and applied in order to translate the data into English.

Subsequently, missing values were handled by dropping products without categories from the products dataset. Out of the 32,951 unique products, 623 products did not have any information available for their accompanying product category. This step is applicable to both the category-wise association mining, as well as the product-wise association mining, as product category is crucial in determining which categories have hidden synergies for category association mining, and which product category to focus on in product association mining.

Data exploration is then performed on the products dataset in order to see how many categories are relevant in the analysis. Overall, there are 72 distinct product categories amongst the remaining 32,328 unique products. The distribution of the top 20 categories by available product on the Olist platform can be seen in Figure 2.2.



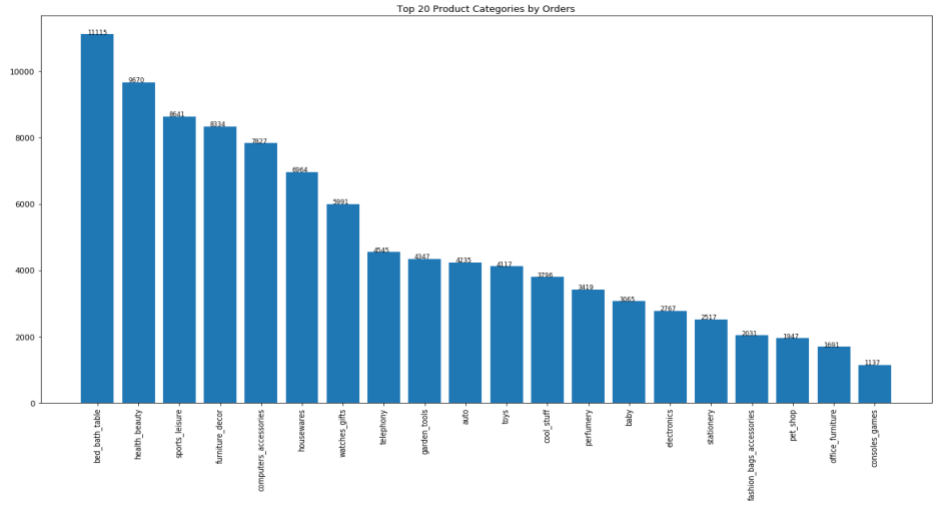
2 **Figure 2.2. Distribution of Olist orders amongst the top 20 product categories**

The next step in pre-processing is joining the tables together in order to create a working dataset. With regards to the category wise association mining, the products table was joined to the orders table via the product_id field to create the order_products working dataset. By joining these two tables, we can see which orders include products from more than one category, and perform association mining on the categories based on the orders. Data exploration is then performed on the order_products dataset in order to see which product category can be attributed to the highest number of orders. An example of the initial order_products dataset can be seen in Figure 2.3 and the distribution of the top 20 product categories by number of orders is illustrated in Figure 2.4.

	product_category_name	order_id
0	cool_stuff	00010242fe8c5a6d1ba2dd792cb16214
1	cool_stuff	130898c0987d1801452a8ed92a670612
2	cool_stuff	532ed5e14e24ae1f0d735b91524b98b9
3	cool_stuff	6f8c31653edb8c83e1a739408b5ff750
4	cool_stuff	7d19f4ef4d04461989632411b7e588b9
...	...	...
111018	garden_tools	ffebd80e3291e811c308365936897efd
111019	furniture_decor	ffee31fb4b5e35c9123608015637c495
111020	watches_gifts	fff7c4452f050315db1b3f24d9df5fcd
111021	sports_leisure	fffa82886406ccf10c7b4e35c4ff2788
111022	bed_bath_table	fffe41c64501cc87c801fd61db3f6244

111023 rows × 2 columns

3 **Figure 2.3. An example of the raw order_products dataset after the missing values for categories have been excluded**



4 **Figure 2.4. A column chart showing the distribution of orders by product category.**

We can see that there is some similarity between the number of unique products offered on the website by each category, and the number of orders placed for each category.

Successively, the dataset must be transformed into the correct format so that it can be analyzed. The dataset is spread into a wider format with each row representing an order and each column representing a product category. The values are then encoded into ones and zeroes denoting

whether or not a product category is present in a specific order. Furthermore, orders that contain only one type of product category are excluded from the analysis at this step. We have chosen to exclude orders with only one product category they are irrelevant to the analysis and the extremely high number of single product orders would significantly affect the minimum support of the algorithm. An example of the transformed dataset is shown in Figure 2.5.

product_category_name	agro_industry_and_commerce	air_conditioning	art	arts_and_craftmanship	audio	auto	baby	bed_bath_table	books_general_interest	books_imported	...	security_and_services	signaling_and_security
order_id													
00010242fe8c5a6d1ba2dd792cb16214	0	0	0	0	0	0	0	0	0	0	...	0	0
00018f77f2f0320c557190d7a144bdd3	0	0	0	0	0	0	0	0	0	0	...	0	0
000229ec398224ef6ca0657da4fc703e	0	0	0	0	0	0	0	0	0	0	...	0	0
00024acbcdf0a6daa1e931b038114c75	0	0	0	0	0	0	0	0	0	0	...	0	0
00042b26cf59d7ce69dfabb4e55b4fd9	0	0	0	0	0	0	0	0	0	0	...	0	0

5 **Figure 2.5. An example of the transformed order_products dataset for category-wise association mining**

The data pre-processing steps for product-wise association mining is identical to category-wise association mining, except that the working dataset is reduced to two associated categories and relationships are considered through the product_id variable. Apart from the product category, the Olist dataset does not disclose the name or nature of the product being sold. Therefore, product_id is used to differentiate between different products. An example of the transformed working dataset for product-wise association mining can be seen in Figure 2. 6.

product_id	0009406fd7479715e4bef61dd91f2462	001b237c0e9bb435f2e54071129237e9	001b72dfd63e9833e8c02742adf472e3	001c5d71ac6ad696d22315953758fa04	002959d7a0b0990fe2d69988affcbc80
order_id					
000229ec398224ef6ca0657da4fc703e	0	0	0	0	0
0006ec9db01a64e59a68b2c340bf65a7	0	0	0	0	0
000e63d38ae8c00bbcb5a30573b99628	0	0	0	0	0
00125cb692d048878096806618a2a145f	0	0	0	0	0
0013503b13da1eac686219390b7d641b	0	0	0	0	0

6 **Figure 2.6. An example of the transformed order_products dataset for product-wise association mining**

2.4. Data Analysis and Results

After the data has been transformed into the correct format, association mining is performed using the Apriori algorithm. The Apriori algorithm identifies relationships between frequent individual items in the entire itemset by observing the frequency in which these subset of items occurs in each transaction[4]. This method was chosen for our market basket analysis as it is devised to work well with datasets with a large number of transactions, such as E-commerce.

For our category-wise association mining, the itemset is defined as $I = \{set\ of\ all\ product\ categories\}$. Each transaction is then defined as an order that a customer makes with the unique identifier being the order_id. The dataset is then fit to an Apriori algorithm with a minimum support of 0.01. The support is defined for the itemset and measures the frequency that an item occurs in a dataset. It is defined by the following formula[5]:

$$\mathbf{supp(X)} = \frac{\text{Count of Transactions Including X}}{\text{Total Transactions}}$$

The association rules are then determined by a confidence metric with a minimum threshold of 0.1. The confidence metric measures the probability of observing the consequent Y in an order, given that the order also contains the antecedent X. It is defined using the following formula[6]:

[4] (Agrawal & Srikant, 1994)
[5] (Hahsler, 2005)
[6] (Hahsler, 2005)

$$\mathbf{conf}(\mathbf{X} \rightarrow \mathbf{Y}) = \frac{\mathrm{supp}(\mathrm{X} \cup \mathrm{Y})}{\mathrm{supp}(\mathrm{X})}$$

Another metric that can be used to determine whether or not rules can be derived is lift, which takes into account the popularity of both item sets. It is defined as[7]:

$$\mathbf{lift}(\mathbf{X} \rightarrow \mathbf{Y}) = \frac{\mathrm{supp}(\mathrm{X} \cup \mathrm{Y})}{\mathrm{supp}(\mathrm{X}) * \mathrm{supp}(\mathrm{Y})}$$

. If the lift is greater than 1, it means that item set Y is likely to be bought with item set X. If lift is less than 1, it means that the presence of item set X could hurt the chances of item set Y being bought. As category association is rather general and there are a large number of categories to be covered, we have chosen to lower the minimum support to 0.01 and also set the confidence threshold to 0.1 so that more results can be returned. After that, we removed all associations with a lift of less than 1. The results of the analysis can be seen in Figure 2.7. By changing the minimum support to 0.05, the results vary drastically and do not seem very useful to Olist as very few associations are returned as a result. These results are shown in Figure 2.8.

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
(cool_stuff)	(baby)	0.089532	0.128099	0.027548	0.307692	2.401985	0.016079	1.259412
(baby)	(cool_stuff)	0.128099	0.089532	0.027548	0.215054	2.401985	0.016079	1.159912
(baby)	(toys)	0.128099	0.068871	0.026171	0.204301	2.966452	0.017349	1.170203
(toys)	(baby)	0.068871	0.128099	0.026171	0.380000	2.966452	0.017349	1.406292
(furniture_decor)	(bed_bath_table)	0.279614	0.272727	0.096419	0.344828	1.264368	0.020160	1.110048
(bed_bath_table)	(furniture_decor)	0.272727	0.279614	0.096419	0.353535	1.264368	0.020160	1.114347
(home_confort)	(bed_bath_table)	0.068871	0.272727	0.059229	0.860000	3.153333	0.040446	5.194805
(bed_bath_table)	(home_confort)	0.272727	0.068871	0.059229	0.217172	3.153333	0.040446	1.189443
(computers_accessories)	(garden_tools)	0.070248	0.100551	0.012397	0.176471	1.755036	0.005333	1.092188
(garden_tools)	(computers_accessories)	0.100551	0.070248	0.012397	0.123288	1.755036	0.005333	1.060498
(construction_tools_lights)	(furniture_decor)	0.024793	0.279614	0.015152	0.611111	2.185550	0.008219	1.852420
(home_construction)	(furniture_decor)	0.035813	0.279614	0.017906	0.500000	1.788177	0.007893	1.440771
(housewares)	(garden_tools)	0.140496	0.100551	0.015152	0.107843	1.072522	0.001025	1.008174
(garden_tools)	(housewares)	0.100551	0.140496	0.015152	0.150685	1.072522	0.001025	1.011997
(health_beauty)	(perfumery)	0.096419	0.035813	0.016529	0.171429	4.786813	0.013076	1.163674
(perfumery)	(health_beauty)	0.035813	0.096419	0.016529	0.461538	4.786813	0.013076	1.678079
(health_beauty)	(sports_leisure)	0.096419	0.092287	0.019284	0.200000	2.167164	0.010386	1.134642
(sports_leisure)	(health_beauty)	0.092287	0.096419	0.019284	0.208955	2.167164	0.010386	1.142263
(housewares)	(sports_leisure)	0.140496	0.092287	0.015152	0.107843	1.168569	0.002186	1.017437
(sports_leisure)	(housewares)	0.092287	0.140496	0.015152	0.164179	1.168569	0.002186	1.028335

7 **Figure 2.7. Results of the market basket analysis for categories with support set to 0.01 and lift greater than 1.**

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(furniture_decor)	(bed_bath_table)	0.279614	0.272727	0.096419	0.344828	1.264368	0.020160	1.110048
1	(bed_bath_table)	(furniture_decor)	0.272727	0.279614	0.096419	0.353535	1.264368	0.020160	1.114347
2	(home_confort)	(bed_bath_table)	0.068871	0.272727	0.059229	0.860000	3.153333	0.040446	5.194805
3	(bed_bath_table)	(home_confort)	0.272727	0.068871	0.059229	0.217172	3.153333	0.040446	1.189443

8 **Figure 2.8. Results of the market basket analysis for categories with support set to 0.05**

Now that we have extracted information about which product categories are frequently purchased together, product-wise association mining can be performed on categories with relationships so that we can recommend specific frequently purchased together products to customers. However,

[7] (Hahsler, 2005)

since there are a very large variety of unique products, we have relaxed the minimum support to 0.005 to accommodate for the fact that many products may not show up often in transactions. We have chosen to perform an example of product-wise association mining on the two categories with the highest confidence: Home Comfort and Bed Bath Table. For this, we have set the minimum threshold of the confidence to 0.5, as product recommendations should be more specific. The results of this analysis can is shown in Figure 2.9.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(fb783e3e545937820b57fe539b2c5a6c)	(0fa81e7123fd0ebe03adbbe99d912827)	0.006024	0.013554	0.006024	1.000000	73.777778	0.005942	inf
1	(35afc973633aaeb6b877ff57b2793310)	(99a4788cb24856965c36a24e339b6058)	0.054217	0.072289	0.043675	0.805556	11.143519	0.039755	4.771084
4	(64fb265487de2238627ce43fe8a67efc)	(84f456958365164420cfc80fbe4c7fab)	0.007530	0.010542	0.006024	0.800000	75.885714	0.005945	4.947289
3	(4d0ec1e9b95fb62f9a1fbe21808bf3b1)	(9ad75bd7267e5c724cb42c71ac56ca72)	0.013554	0.019578	0.009036	0.666667	34.051282	0.008771	2.941265
7	(ad0a798e7941f3a5a2fb8139cb62ad78)	(9463446971569 47d846d27fe0d503033)	0.013554	0.015060	0.009036	0.666667	44.266667	0.008832	2.954819
2	(99a4788cb24856965c36a24e339b6058)	(35afc973633aaeb6b877ff57b2793310)	0.072289	0.054217	0.043675	0.604167	11.143519	0.039755	2.389347
6	(9463446971569 47d846d27fe0d503033)	(ad0a798e7941f3a5a2fb8139cb62ad78)	0.015060	0.013554	0.009036	0.600000	44.266667	0.008832	2.466114
8	(f2e53dd1670f3c376518263b3f71424d)	(99a4788cb24856965c36a24e339b6058)	0.015060	0.072289	0.009036	0.600000	8.300000	0.007947	2.319277
5	(84f456958365164420cfc80fbe4c7fab)	(64fb265487de2238627ce43fe8a67efc)	0.010542	0.007530	0.006024	0.571429	75.885714	0.005945	2.315763
9	(c211ff3068fcd2f8898192976d8b3a32)	(f4d705aa95ccca448e5b0deb6e5290ba)	0.010542	0.010542	0.006024	0.571429	54.204082	0.005913	2.308735
10	(f4d705aa95ccca448e5b0deb6e5290ba)	(c211ff3068fcd2f8898192976d8b3a32)	0.010542	0.010542	0.006024	0.571429	54.204082	0.005913	2.308735

Figure 2.9. Results of the market basket analysis for products in the home_comfort and bed_bath_table categories

Product-wise association mining can then be performed on all pairwise product category relationships. Figure 2.7 and 2.8 show the categories that are often bought together, such as Home Comfort and Bed Bath Table with the highest confidence value of 0.86. Overall, there are 20 pairwise category associations with a lift of greater than 1. In Figure 2.9, one can see the specific products that have a high probability to be bought together within the Home Comfort and Bed Bath Table category. As Olist does not provide actual product names, but instead masks them with a Product ID, we cannot determine what the exact products are. This analysis on one pair of associated categories can be extended to all the pairs categories with relationships. The results can then be combined into a master table of product associations across various categories.

Olist's marketing team can now implement marketing strategies to encourage customers to add another item to their purchase. Furthermore, for products without specific associations with other products, Olist could randomly select a product from an associated category with the highest confidence.

Ultimately, this section of the report has shown how market basket analysis can be implemented on an E-commerce dataset in order to derive relationships between products to provide customers with improved product recommendations in hopes of increasing sales. With this market basket analysis, Olist will be able to provide suitable specific product recommendations for products with existing product-wise relationships, and general product recommendations in categories of interests for others with category-wise relationships. The Kaggle link to the Python code for data pre-processing and analysis for the Olist section can be found in Appendix 2.

3. Supplier Analysis and Price Prediction: Cashew Truck Arrivals

Cashew truck arrivals is a dataset found on Kaggle and updated 10 months ago. This dataset shells the deliveries of cashew nuts from bush to port warehouse. It covers a period of 2 years from 2015 to 2017 and has about 200 observations per year. The dataset presents the different deliveries indexed by the date of the delivery. It lists 670 deliveries from 139 different suppliers. This table

contains 16 columns that display the different characteristics of the deliveries. The attribute descriptions can be found in Appendix 3.

3.1. Supply Chain Context

The dataset presents the quality of the nuts, the size of the nuts can be computed by comparing the weight and the nut count in the bag and finally the quality of the supplier. Therefore, this dataset can be useful in different application area of the supply chain. Delivery optimization, scheduling, conditioning, quality prediction, forecasting and supplier selection.

3.1.1. Delivery Optimisation and Scheduling

A well optimized supply chain means more profits and less costs. While reviewing the data presented to us, we have noticed that our deliveries are seasonal due to the seasonality of the cashew nuts. Therefore, optimizing the deliveries and scheduling drop of times would reduce the stress that is created on the warehouse. The need of flexible manpower due to overflow of the laboring power is reduced thus driving the cost of warehousing down[8].

3.1.2. Quality Prediction

"Product quality prediction would allow a manufacturer to make better choices of system parameters at the early design stage and, hence, enhance competitive-ness through achieving higher quality levels"[9]. By analyzing the data, we can predict the different quality of nuts depending on the supplier, origin of the nuts, date of delivery, etc. Moreover, we can further develop the study to create a pricing system according to quality forecast.

3.1.3. Forecasting and Order Generation

Delivery forecasting and order generation is generally used in order to make sure that the customers never run out of products. In a vendor management inventory (VMI) environment, the vendor which in our case is the company responsible of running the warehouse. By analyzing and forecasting the demand, one can control the inventory and the scheduled deliveries. Using this dataset and by applying specific methods, we can offer VMI to our different customers and control the demand and order generation and thus optimizing the deliveries[10].

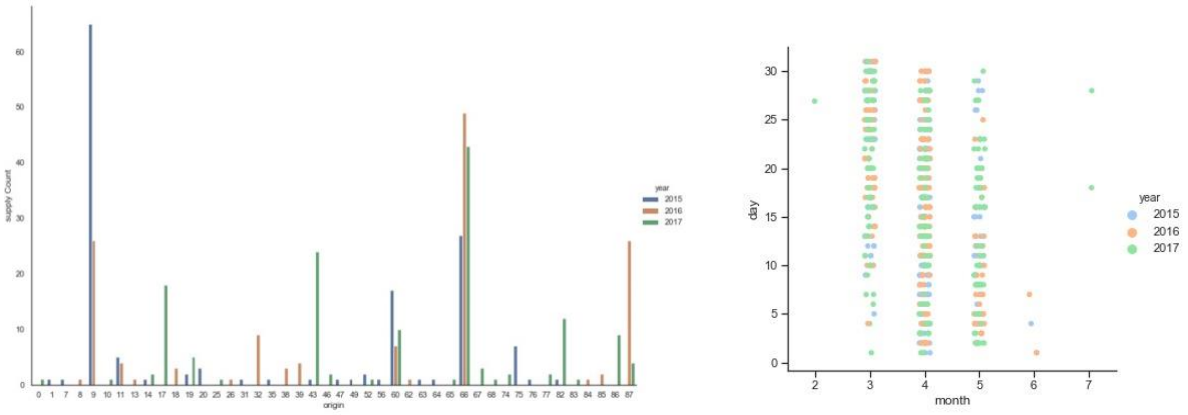
3.1.4. Supplier Selection

Supplier segmentation is usually used to allow the company to define the level of engagement with each supplier depending on certain variabilities. This segmentation is usually aligned with the strategy of the company. Generally, segmentation is dividing the suppliers into three different groups. The first and highest level is usually limited to three or four suppliers. The lowest section is usually the biggest and groups the occasional suppliers. The middle sector groups the suppliers that would need some management and those that have potential to become long term partners. This last group creates competition and pushes the top suppliers for continuous improvement[11].

[8] (Johnson, 2019)
[9] (Omayma A.Nada, 2006)
[10] (ORTEC, 2019)
[11] (Nanncy, 2017)

3.2. Data Exploration

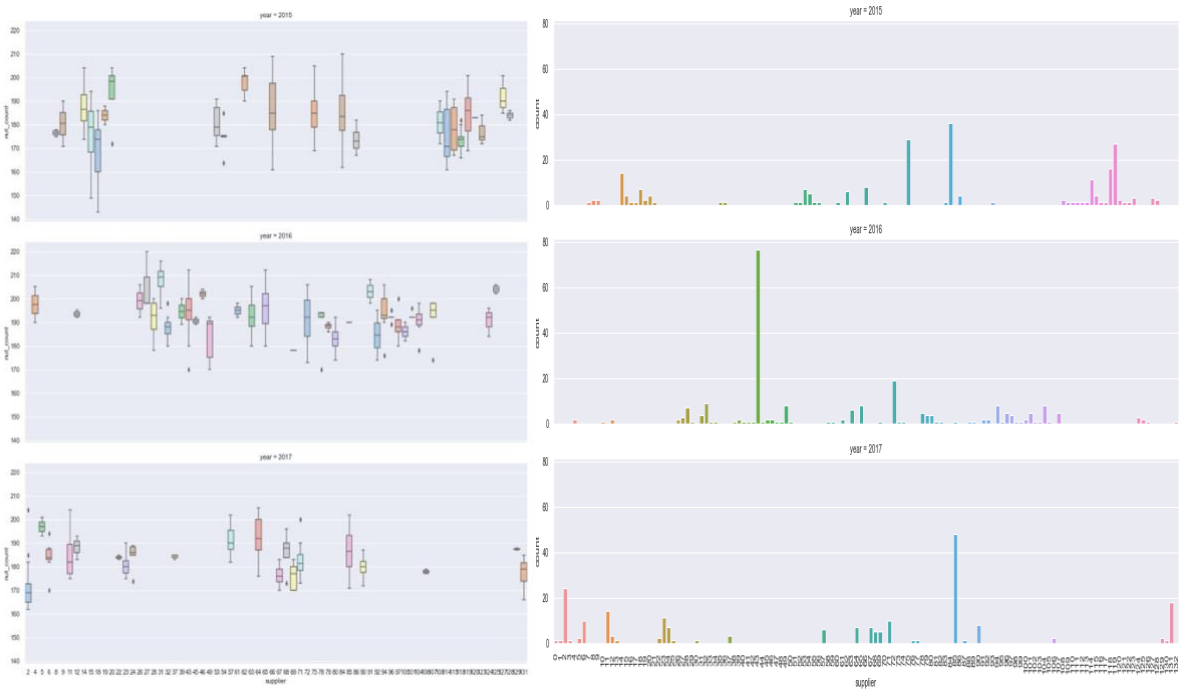
Data exploration was helpful in order to determine how the dataset was spread and what scenario can be developed and analyzed. Figure 1 shows the count of deliveries according to the origin. Out of the 87 origins, only one is predominant per year. For instance, in 2015, most of the shipments originated from location number 9 while in 2016 and 2017 origin 66 was leading with more than 40% of shipments. Moreover, while visualizing the date of the deliveries, a pattern was established. Few shipments start arriving at the beginning of March, the vast influx of raw nuts starts in the first third of March and ends at the end of May. Some shipments will still arrive in the first third of June. This finding is aligned with the research presented by Bhaskara Rao, director of the National Research Center for Cashew, India[12].



10 Figure 3.1: Number of supplies by origin and year & Figure 3.2: distribution of deliveries' date

Figure 3.3 shows the box plot of the nut counts per supplier per year. From this figure, we were able to conclude that, since the figure is differentiated by year, not all suppliers deliver every year. Therefore, the scenario development and the aim of the project which is to develop strategic and long-term relationships with suppliers are aligned with the dataset. Having no permanent supplier, the cashew nut distributor will not be able to rely on constant quality and quantity.

[12] (Rao, 1998)



11 Figure 3.3: distribution of nut count (left) shipment count(right) per supplier per year

3.3. Scenario Development

Exploration of the raw data provided to us, was helpful for us to interpret basic behavior shown by our suppliers and pose a question on how we can filter suppliers and improve business decisions. From our exploratory findings, we hypothesized the following scenarios:

- Cashew nut harvest is seasonal and not all suppliers supply each year. For the suppliers that supply each year, the number of delivery and the distribution of the nut_count varies each year. This trend supports the idea that "these dynamics must create competition for the top suppliers and push them to innovate and improve their product quality and quantity". As for the other suppliers even though deliveries are usually limited to one-year, better product quality and quantity would potentially overcome any loss incurred by limitation on the number of deliveries. Therefore, we decided to cluster suppliers and based on the clustering results create labels for suppliers indicating good or bad suppliers. Creating such labels is aimed at performing supplier selection so that we can create strategic alliances with good suppliers. These strategic alliances would mean a significant increase in information sharing including cost information and processes transparency. Moreover, this high level of supplier contact would also mean easier cost prediction and delivery schedule. Overall these alliances would influence our business directly by being able to promise our customers the right quality with high confidence in our suppliers.

- Missing values were not found in the dataset except for the 2015 prices. Instead of filtering those data points, the next step of our project would be to create a price prediction model. Where the model would be trained, and predictions evaluated on the 2016 and 2017 prices while the generalization of the model prediction would also be explored using the 2015 prices. The aim of building this model at this stage will be limited to exploring mainly what

variables are important for price prediction and secondly which algorithm we can potentially use to predict the shipment prices. With improved information sharing with our suppliers, we can focus on collecting and improving data on these variables that our models will indicate as important.

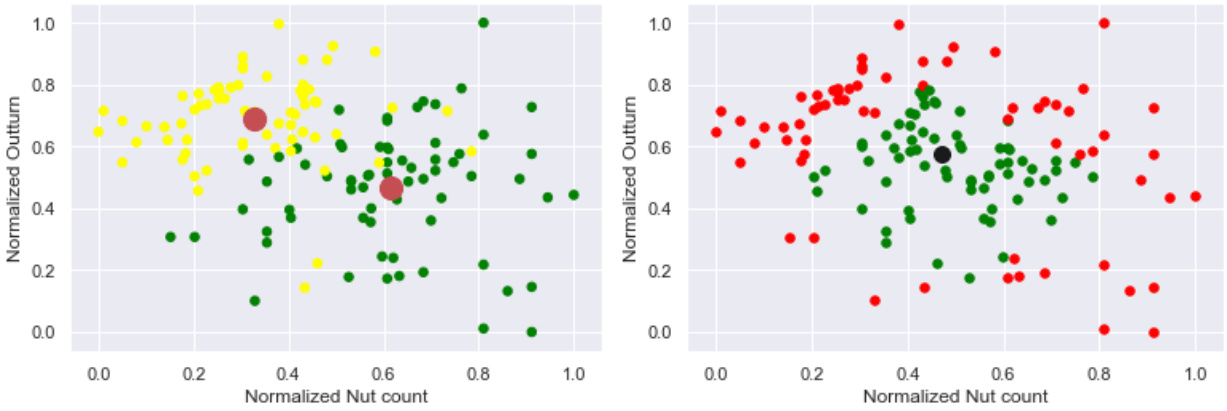
3.4. Data Preprocessing

Each supplier in our dataset delivers multiple times in one year, in order to start our clustering and to efficiently group the different suppliers, we group observations in the dataset by the supplier and collapse the dataset by calculating the mean of each variable for the supplier. Also as the aim of clustering for us is to filter suppliers based on quality and quantity of the suppliers it makes sense to train the clustering algorithm based the following variables: 'nbags', 'net_weight', 'moisture', 'nut_count', 'outturn', 'defective' and 'avg_wpb'. This pre-processing step, therefore, results in a dataset with 132 points for each supplier containing columns with the average value of each of these variables. Which is not much of a problem for training a clustering algorithm but, would not be sufficient for our prediction algorithms to learn from. Therefore, to train the prediction algorithms we use the original dataset, by excluding information on the supplier and using the variables: 'nbags', 'net_weight', 'moisture', 'nut_count', 'outturn', 'avg_wpb'. We also decided to include a new variable 'label' which would indicate if the supplier was filtered as good or bad from our labeling. We then divided the data set into divisions based on the years 2016/17 and 2015. The 2016/17 dataset would be our training dataset and the dataset with information present from 2015 would then be used to see how our model generalizes to new datasets.

3.5. Data Analysis and Results

3.5.1. K-Means Clustering

As discussed in the scenario development, we started by clustering the suppliers according to the attributes earlier. We implemented the k-means function present in the sklearn package in Python, build 2 clusters of the suppliers based on the variables used. After the clustering, we plotted each supplier color-coded with the cluster they belong to in a phase plane with the Normalized Outturn in the y-axis and Normalized Nut Count on the x-axis. The resulting plot is shown in Figure 3.4. The "yellow" cluster has the suppliers with higher outturn but also tends to have a low nut count which means that these particular suppliers provide shipments with high quality but low quantity. The second cluster "Green" group suppliers who have high quantity but low quality. This allowed us to determine the existence of a tradeoff between quantity and quality when we choose suppliers. Following this observation, we decided to find the point in the phase plane where the tradeoff would be minimum. We decided that this point on the phase plane would be the midpoint between the 2 centers that we determined earlier (black point in Figure 3.5). This midpoint theoretically represents the optimum tradeoff between quality and quantity. By calculating the distances of each supplier in the phase plane from this optimum center, we classify(label) the suppliers as good (low tradeoff) or bad (high tradeoff). The suppliers are labeled as good if their distance from the optimum point is less than the median of the distances for all suppliers from this point. Figure 3.5 plots the color-coded suppliers (Green=Good and Red=Bad) in the same phase plane of Nutcount Vs Outturn.



12 Figure 3.4: Supplier clustering & Figure 3.5: supplier classification

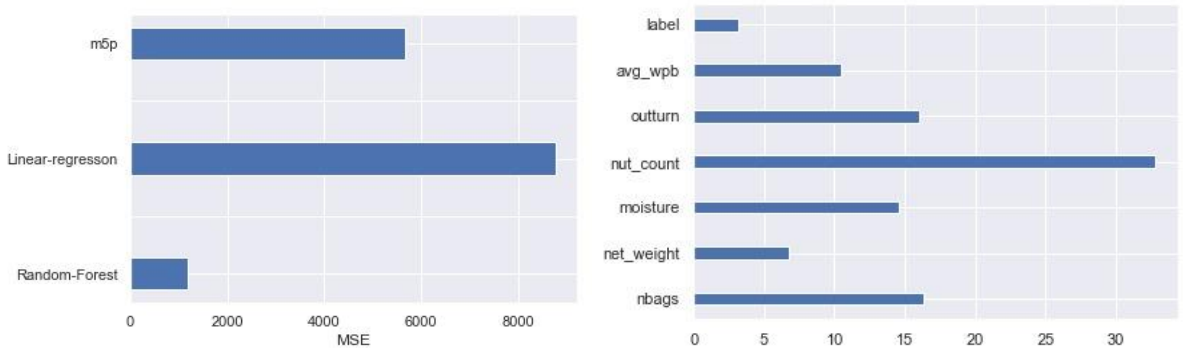
3.5.2. Price Prediction Model

After labeling the suppliers as good or bad based on the clustering results, attaching the label in the original dataset and splitting the dataset into training (year = 2016/17), evaluation (year = 2016/17), and generalization (year=2015) as discussed in the scenario development section we built three different models to predict the prices. These three models were built using the following algorithms:

- Linear Regression
- Random Forest (with 500 trees)
- M5p decision tree by appending a regression model to each node of the tree

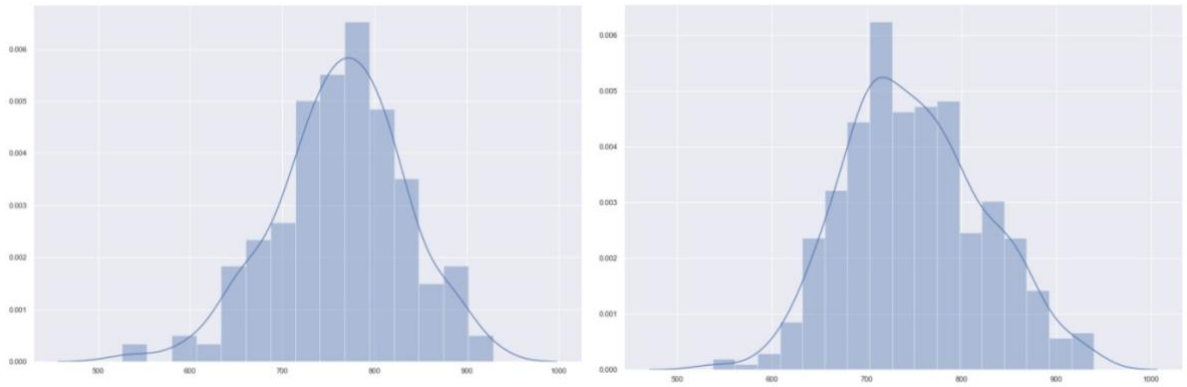
The Linear Regression and Random Forest model were built using the algorithm functions present in the sklearn library of python. While we built the m5p model from scratch based on the m5p function available in R package Cubist.

Figure 3.6 shows the Training error calculated by using Mean Square Error (MSE) as our loss function for each of these models. As seen in the figure the Random-Forest model gives us the least training error i.e. the least MSE. Therefore, for the purpose of looking at the Variables that influence the price the most, we plotted the importance of each variable (based on averaging the decrease in impurity over trees) (Figure 3.7) given by the Random Forest model. As seen in figure nut_count is the variable that is most important to predict the price while the 'label' variable we created does not influence the price as much. Other variables that we used to train the model also show significant effects in decreasing the average impurity over the trees built.

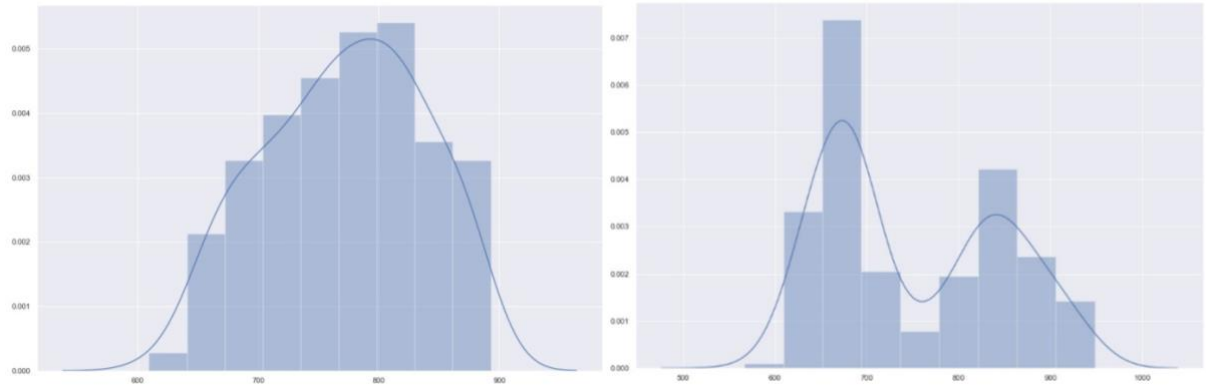


13 Figure 3.6.: Error rate of 3 models & Figure 3.7: Variable of importance random forest

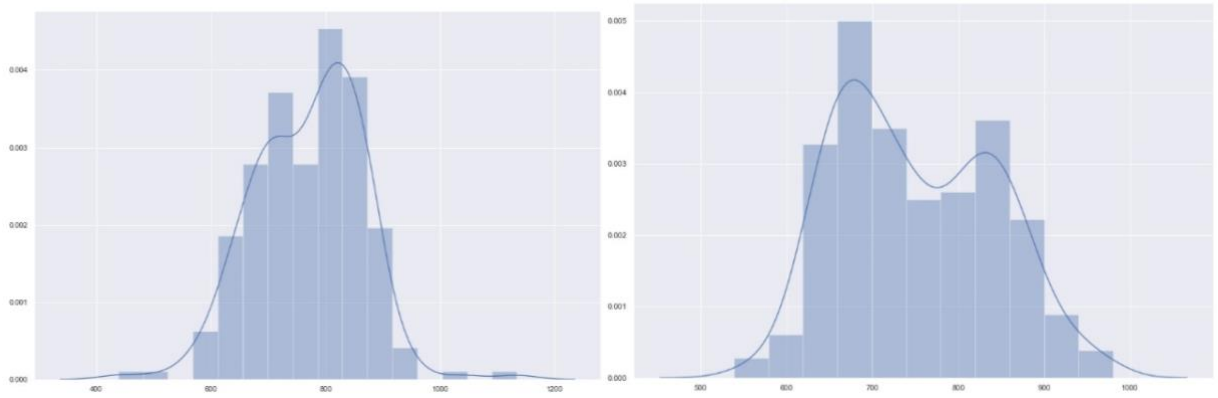
The next step for us then was to see how this model generalizes to the data from 2015. In this case, we did not have a straight way to evaluate the models using a loss function as we were missing the prices for 2015. We decided to plot the distribution of the prices from the model predictions for the training data and the 2015 data to compare them with the actual distribution of prices so that we can see if the models were overfitting the training data. Figure 3.8 through 3.13 shows these distributions and Figure 3.14 shows the actual distribution of the prices on which the models were trained.



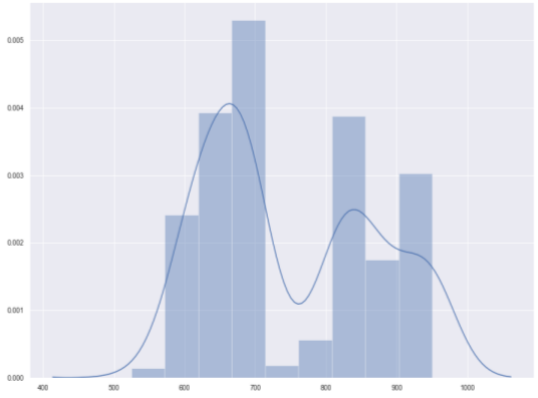
14 Figure 3.8: Prediction of the linear model 2015 data & Figure 3.9: Prediction of the linear model train data



15 Figure 3.10: Prediction of the random forest 2015 data & Figure 3.11: Prediction of the random forest train data



16 Figure 3.12: Prediction of the M5P model 2015 data & Figure 3.13: Prediction of the M5P model train data



17 Figure 3.14: Distribution of Prices in the Training dataset

From the plots, it is evident that the distribution plot of predictions made by the random forest model is the closest one to the actual distribution. This hints us to the fact that the random-forest model is overfitting in our training data. The m5p model also mimics the distribution but not as close as the random forest model, which indicates that in terms of overfitting the m5p model does better than the random forest. The distribution for the predicted price in the 2015 dataset has a similar distribution predicted by all three models. The predictions range from similar minimum value and maximum value. These evidences are not enough to decide on the best model, until we evaluate the loss function using a validation data-set which was not possible for us dur to the lack of data-points present in our dataset.

3.6. Proposal for Improvement

The dataset used is not stored in a database. Therefore, it was fine to include all the observations in one table. Nevertheless, in order to store it in a database, it has to be more structured. The proposed ER diagram in Appendix 4 splits the dataset into 3 different entities (Product, Truck, Supplier). They are linked together with relationships such as the product is delivered by a truck that has an ID, net weight of the shipment and number of bags delivered by this particular truck.

Finally, we suggested adding three attributes related to the date the supplier issued the shipment, the date the truck actually delivered the shipment and the classification of the supplier.

By looking at the outputs of our models, we can assume that none of them predicts the output accurately. Nevertheless, by creating strategic alliances with our top suppliers, we can expect a constant data flow and maybe develop new features to measure. Finally, with a better structured dataset and more information about the shipments, the model created will be able to better predict the expected outcome.

4. Product Quality Control: Iron Ore Production

This particular dataset contains manufacturing process data from a real world iron mining floatation plant. It contains 24 columns describing different aspects of the flotation process in iron ore mining. This process is a standard procedure to further concentrate the iron ore. The attribute descriptions can be found in Appendix 5.

4.1. Suggested Dataset Improvements

The current dataset contains some iron ore production values in hour intervals and some in 20 second intervals within the same table. The hourly values are simply repeated 180 times, which is also true for the hourly timestamps. Since there are no precise timestamps or disclaimers for the values with a frequency of 20 seconds, it was up to us to figure out which ones were in which frequency and to assume that the intervals were in the correct order for every hour. In order to avoid these assumptions, it would be good if similar datasets in the future provided precise timestamps and appropriately named variables for each frequency or just split the data with different frequencies into separate tables with the ability to join them on the time indexes.

4.2. Supply Chain Context

The Iron Mining Process dataset is limited in its applicability to different supply chain scenarios. The stated main goal on the Kaggle website is production control, which is the only scenario it is properly suited for. Production planning is the only other slightly relevant context but since the dataset contains percentage contents of the final product instead of numeric quantities, it does not allow for this application at the most basic level. Thus, it is only suited for production control, and specifically only for quality control and monitoring.

In manufacturing defect prediction, the main variables usually include values measuring the quality of the input materials as well as other relevant measurements throughout the process[13]. In case of the Iron Ore Mining dataset, the most important variables would thus be the starting purity measures such as % Iron and Silica Feed as well as other direct process measurements such as the Flow and Ore Pulp variables. Indirect process measurements such as the air flow and froth level could also have an impact, which will ultimately be determined throughout the modelling process.

[13] (Santos, et al., n.d.)

4.3. Scenario Development

Currently, the engineers at the plant do not have a convenient and reliable way to measure the iron ore impurity, i.e. the quality of their product. If an engineer wants to assess the quality and contents of the iron ore at the end of the floatation process, the contents of the ore have to be measured in a lab which takes about an hour. This means that engineers can only take actions to ensure proper product quality with at least an hour delay and only in case a sample was even chosen for testing to begin with. Thus, the plant's engineers lack a proper way to diagnose and continuously monitor product quality, which could lead to poor product quality and even cases where the product cannot be sold for its intended purpose.

The goal for analyzing this dataset is hence to provide the engineers a data-driven solution for monitoring the product quality during the iron ore concentrate production process. A successful impurity model would allow the engineers to respond to potential cases of poor product quality in a more timely and organized manner, ultimately helping the plant by improving product quality on average and preventing long periods of poor production quality.

We attempted to model the impurity with two different approaches. Initially, we tried different regression methods in order to directly forecast the percentage of Silica Concentrate as part of the final Iron Ore Concentrate. Moreover, after looking at the distribution of the Silica Concentrate values, we decided to try classification methods in order to identify "impure" or "pure" batches. The classification output could then be used to trigger a warning to engineers instead of showing the potentially imprecise or misleading regression forecast.

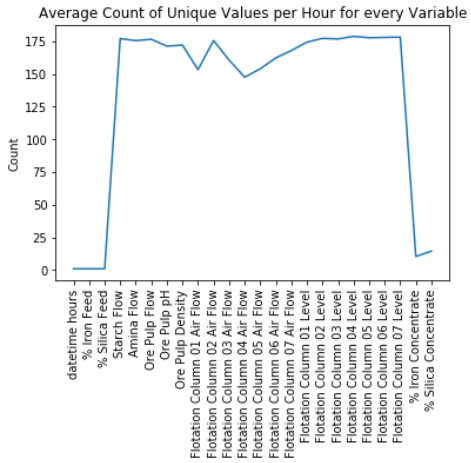
4.4. Data Exploration and Preprocessing

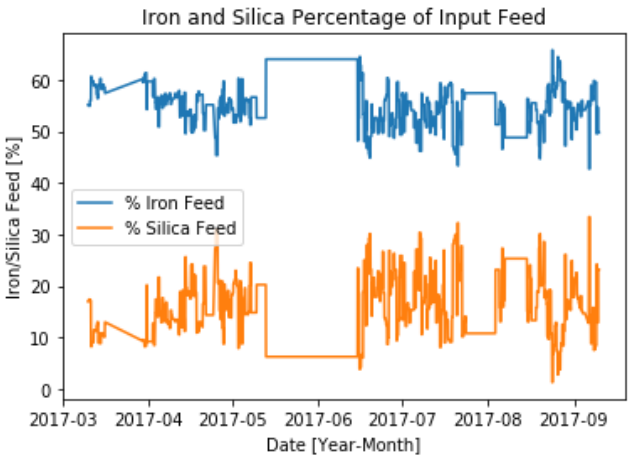
The goals for this step are to gain a deeper understanding of the dataset at hand and prepare it for the modelling step. Upon initial inspection, the dataset contains no explicit missing values and a little more than 737.000 rows. Since the dataset consists of only one table, we did not need to transform its structure initially.

As previously mentioned, some of the variables are provided in hourly frequency and some in 20 second frequency, so the first step was to determine said measurement frequency. Figure 1 shows how many unique values per hour each variable averages. It seems that only % Iron Feed, % Silica Feed, % Iron Concentrate and % Silica Concentrate are provided in hourly frequency, as the rest of the process measurements only contain very few repeating measurements on average. It is important to note, however, that both Concentrate variables have an average unique count of above 1, which indicates some inconsistencies in the data.

As the majority of the variables are not in hourly frequency, we decided to create a column detailing the exact measurement moment, assuming the observations within each hour were in the correct order. Before proceeding, we also tested which hours had less than 180 records, which is the amount of 20 second intervals in an hour. Throughout this process, we noticed that two hours contained less than 180 records and that data for some hours was missing from the dataset entirely. Additionally, some hours for % Silica Concentrate, our intended target variable, contained exactly 180 unique values, seemingly due to an interpolation procedure between the value of the previous and upcoming hour. We decided to remove those hours from the forecasting dataset. This exploration can be seen in Figure 4.1.

When individually graphing all variables as a lineplot, both Iron and Silica Feed stood out for similar reasons. Both graphs show multiple plateaus, where the values are constant for an extended period of time as shown in Figure 4.2. After investigating further, we decided to remove the corresponding rows from the dataset as well, as the input components of the iron ore seem like important factors in this case. In other cases, we could have also considered either excluding the variables completely or keeping them as is, but the perceived value of those two variables in this context shaped our decision.

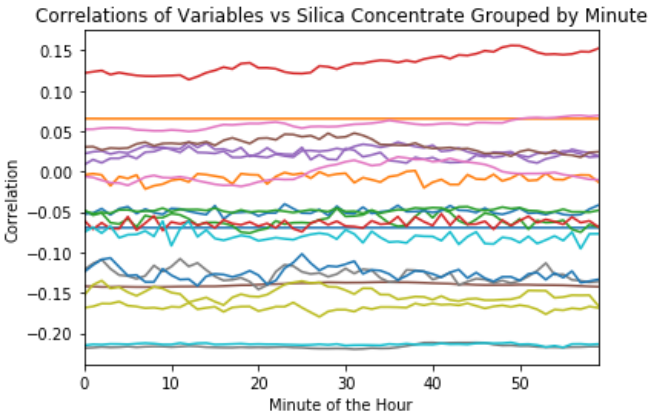




18 Figure 4.1: Lineplot of average unique values per hours & Figure 4.2: Time Series Plot of % Iron Feed and % Silica Feed for the entire dataset.

To further delve into the relationships between the variables, we wanted to examine the correlation between the features in order to inform our modelling decisions. We found that there were no helpful, significant correlations. Appendix 6 shows a pairplot of select variables, which includes scatterplots between each variable and a histogram to show the distribution of each variable along the diagonal. The only apparent patterns indicate a relationship between % Iron Feed and Silica Feed as well as Iron Concentrate and Silica Concentrate, which are to be expected as both are percentage contents of the same material.

Our last hypothesis was that - assuming the data points were ordered correctly and measurements were usually carried out around the same time - each variable should exhibit a higher correlation with the % Silica/Iron Concentrate around the time the measurements were usually taken. Thus, we decided to group the dataset for minutes within the hour, and examine the correlation within those subgroups. The correlations for all variables are fairly low and do not follow a significant hourly pattern overall, as shown in Figure 4.3. As a result, the measurements seem to be either taken at random throughout the hour or the moment of measurement has no impact on the correlation with the process variables.



19 Figure 4.3: Lineplots depicting correlation between all individual variables and % Silica Concentrate grouped by minutes of the hour

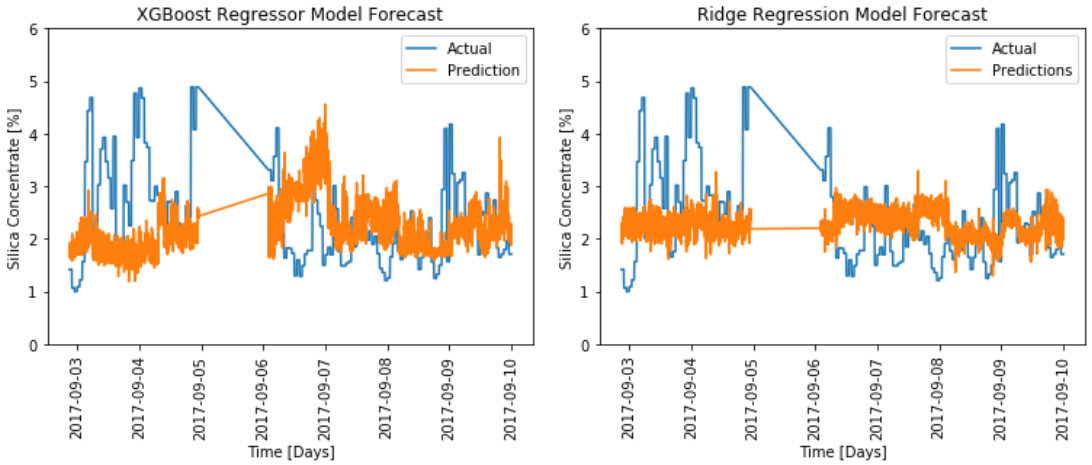
4.5. Data Analysis

After data exploration and cleaning, the next step was to attempt to model the Silica Concentrate based on the input and process variables. Overall, the dataset seemed fairly uncorrelated, meaning it could prove to be difficult to produce accurate models. In addition, the inherent nature of the values and their measurement frequencies presented a cause of concern, as we had to decide which frequency to use for forecasting. We ultimately decided to stick with the lower 20 second frequency, as this allows us to use all of the remaining data. Initially, our goal was to numerically predict the % Silica Concentration using tree-based and regular regression algorithms. Tree-based algorithms seemed especially promising in this case considering the low linear correlation between the variables.

We started the modelling attempts using the XGBoost tree-based Regressor, which uses gradient boosting in order to find the optimal tree structures[14]. After a little bit of experimental parameter tuning, we decided to fit the model on the first 130 days worth of data and predict the rest, as shown in Figure 4.4. While the predictions seem generally close, they do not follow the actual patterns and do a poor job of correctly predicting the spikes in % Silica Concentrate. Comparing the accuracy measures, a value for RMSE of 1.14 and MAE of 0.87 for values ranging between 1 and 5 is very high. The RMSE is the square root of the average squared error, the MAE is the mean absolute error value.

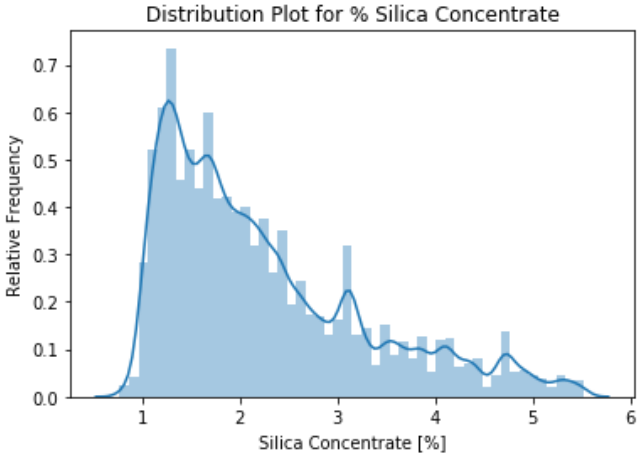
Considering such disappointing results, we decided to fit a Ridge Regression model for comparison, Ridge Regression is similar to Linear Regression, except that it includes a regularization term in its loss function to prevent overfitting[15]. As compared to the XGBoost modelling attempt, the Ridge Regression shows an even lower capacity to correctly forecast the important outliers as shown in Figure 4.4. Even though the accuracy measures for Ridge Regression are slightly better than for XGBoost with an RMSE of 0.97 and a MAE of 0.75, they are hardly inspiring.

[14] (Chen, n.d.)
[15] (Hoerl & Kennard, 2000)



20 Figure 4.4: Time Series Plots depicting the actual values for % Silica Concentrate and the predicted values from the XGBoost Regressor model Ridge Regression model respectively

As numeric prediction did not yield the necessary results, we decided to try a different approach. After looking at the distribution of the % Silica Concentrate target variable shown in Figure 4.5, we noticed a left-skewed distribution with a long tail on the right side, indicating a fairly substantial amount of high impurity cases as the percentage of Silica in the Concentrate increases. As predicting the higher values in Silica percentage is of utmost importance, we chose to implement a classifier which would label a sample as impure if it contained more than 3% Silica based on the distribution plot. Our goal was not only to produce an accurate classifier, but most importantly to produce a classifier that was accurate in predicting impurity.

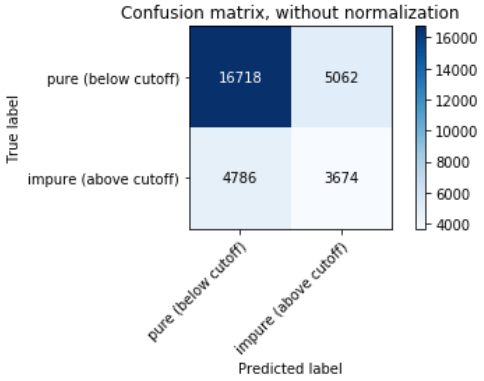


21 Figure 4.5: Histogram and Distribution Plot of the % Silica Concentrate Variable

Again, we decided to try one tree-based and one standard regression method. Due to its inherent nature as defect detection, the classes were fairly unbalanced. Initially, there were close to 4 times as many pure observations as there were impure observations. We had trouble adjusting the models to account for this imbalance, and ultimately decided to randomly pick an equal amount of observations as the impure class from the sample of pure observations. Even though this balancing meant we were losing a large number of observations, the modelling results ultimately improved.

We decided to evaluate the classification methods using precision and recall[16] for the 'impure' observations as well as general accuracy measures. Recall measures the proportion of correctly classified cases of a specific against all actually observed cases of that class, whereas precision measures the proportion of correctly classified cases over all cases classified as that particular class by the model.

The XGBoost classifier especially struggled with the class imbalance, and ultimately did not perform very well. In particular, the algorithm performs fairly poorly when attempting to classify impure observations. The low recall and precision values of 0.11 and 0.34 for the impure label are thus not surprising. In contrast, Logistic Regression proved to be much more stable overall. Figure 4.6 shows the resulting confusion matrix after testing set prediction. Although it certainly presents an improvement over XGBoost, it is still fairly far away from a model that can be used in production, with values of 0.43 and 0.42 for recall and precision. The overall accuracy score of 0.69 is deceiving, as it is heavily skewed by the imbalance of the testing set which was not adjusted for class size.



22 Figure 4.6: Confusion Matrix for the Logistic Regression model predictions

4.6. Results and Possible Improvements

Our final plan was to combine the high frequency interval predictions on an hourly basis, and use the combined prediction to ultimately trigger the alerts. Yet, our models were ultimately too inaccurate to achieve any meaningful results even when combined in that way. Overall, the different measurement frequencies, seeming lack of relevant features and generally poor data quality and documentation severely limited us in our attempts. In order to build an accurate and helpful model, the dataset has to be severely improved in terms of quality, documentation and extensiveness in terms of features and observations. It is also important to note that while the dataset contained more than 700000 observations, the actual amount of observations for Silica Concentrate was only about 4000, with only 290 distinct values after removing interpolated hours. Thus, the size of the dataset might seem large, but seems to contain very little relevant information. In terms of modelling approaches, we also discussed other approaches such as more time series related methods or using hourly patterns as features, yet the quality of existing data and lack of

[16] (Powers, 2007)

continuity in the dataset ultimately deterred us from any attempts. While we have not produced a model that is ready to be used in production, we think our approach is still replicable with a better dataset and our feedback on the dataset is valuable in order to provide higher quality datasets in the future.

The link to the Kaggle code for this section can be found in Appendix 7.

5. Conclusion

This report has demonstrated several use cases and supply chain scenarios in which analytics can be used to improve general business strategy as well as parts of the supply chain. For the Olist dataset, our team leveraged association mining to build a model which can be used in marketing campaigns as a product recommendation system based on sales data. Both the cashew nuts and iron ore production cases present clear supply chain scenarios and analytics solution outlines, but unfortunately both presented shortcomings of the provided datasets in terms of structure, quantity and quality which stopped us from producing an accurate solution. Furthermore, although the dataset used for market basket analysis was well structured, the solution also required a workaround due to row overflow issues. Therefore, we provided specific suggestions for improvement given the supply chain context. Although we could not provide a fully built solution, this process allowed us to better understand the necessary standards for datasets in order to enable analytics.

Bibliography

Agrawal, R. & Srikant, R., 1994. Fast Algorithms for Mining Assocation Rules. *Proceedings of the 20th VLDB Conference Santiago, Chile,* pp. 487-499.

Blattberg, R. C., Kim, B.-D. & Neslin, S. A., 2008. *Market Basket Analysis. In: Database Marketing. International Series in Quantitative Marketing.* 18 ed. New York, NY: Springer.

Carnein, M. & Trautmann, H., 2019. Customer Segmentation Based on Transactional Data Using Stream Clustering. *PAKDD 2019: Advances in Knowledge Discovery and Data Mining,* pp. 280-292.

Chen, T., n.d. *Introduction to Boosted Trees.* [Online]
Available at: https://xgboost.readthedocs.io/en/latest/tutorials/model.html
[Accessed 10 November 2019].

Hahsler, M., 2005. Introduction to arules – A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software.*

Hoerl, A. E. & Kennard, R. W., 2000. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics,* 42(1), pp. 80-86.

Islek, I. & Ögüdücü, S. G., 2015. A retail demand forecasting model based on data mining techniques. *IEEE 24th International Symposium on Industrial Electronics (ISIE),* pp. 55-60.

Johnson, T., 2019. *tinuiti.* [Online]
Available at: https://tinuiti.com/blog/ecommerce/supply-chain-optimization/
[Accessed 9 November 2019].

Nanncy, C., 2017. *Supplier Segmentation – The First Step of an Effective SRM Programme.* [Online]
Available at: https://spendmatters.com/uk/supplier-segmentation-first-step-effective-srm-program/

Omayma A.Nada, H. A. W. H., 2006. Quality prediction in manufacturing system design. *Journal of Manufacturing Systems,* 25(3), pp. 152-171.

ORTEC, 2019. *Demand Forecasting and Order Generation.* [Online]
Available at: https://ortec.com/en/dictionary/demand-forecasting-and-order-generation
[Accessed 9 November 2019].

Powers, D. M. W., 2007. *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation,* Adelaide: Technical Report SIE-07-001.

Rao, B., 1998. *4. INTEGRATED PRODUCTION PRACTICES OF CASHEW IN INDIA.* [Online]
Available at: http://www.fao.org/3/ac451e/ac451e04.htm
[Accessed 22 November 2019].

Santos, I., Nieves, J., Penya, K. Y. & Bringas, G. P., n.d. *Optimising Machine-Learning-Based Fault,* s.l.: Deusto Technology Foundation.

Appendix

Appendix 1: Olist Table Descriptions

The Olist dataset consists of 9 tables in its schema. The link to the dataset is as follows:

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

The table definitions can be found below:

1. Customers: Gives details about the location of each customer.
2. Order_Items: Lists the contents in each order and gives further details on each product in an order, such as the seller, freight value and price.
3. Order_Reviews: Each order receives a review, containing for example a score, a message and a time stamp.
4. Products: Provides details of each product, such as product category, dimensions or the number of photos.
5. Product_Category_Name_Translation: A Portuguese to English translation table.
6. Geolocation: Matches zip codes with the corresponding longitudes and latitudes.
7. Order_Payments: Provides payment information for each order, such as payment type and value.
8. Orders: Details about each order, containing customer, order status and different timestamps.
9. Sellers: Gives details about the location of each seller.

Appendix 2: Kaggle Link to Olist Code

The link to the Kaggle Kernel used to analyse the Olist dataset is listed below. The code present in the link is used for the report.

https://www.kaggle.com/mchindasook/da-for-scm-e-commerce-dataset

Appendix 3: Cashew Truck Delivery Attribute Description

The link to the cashew truck delivery dataset is as follows:

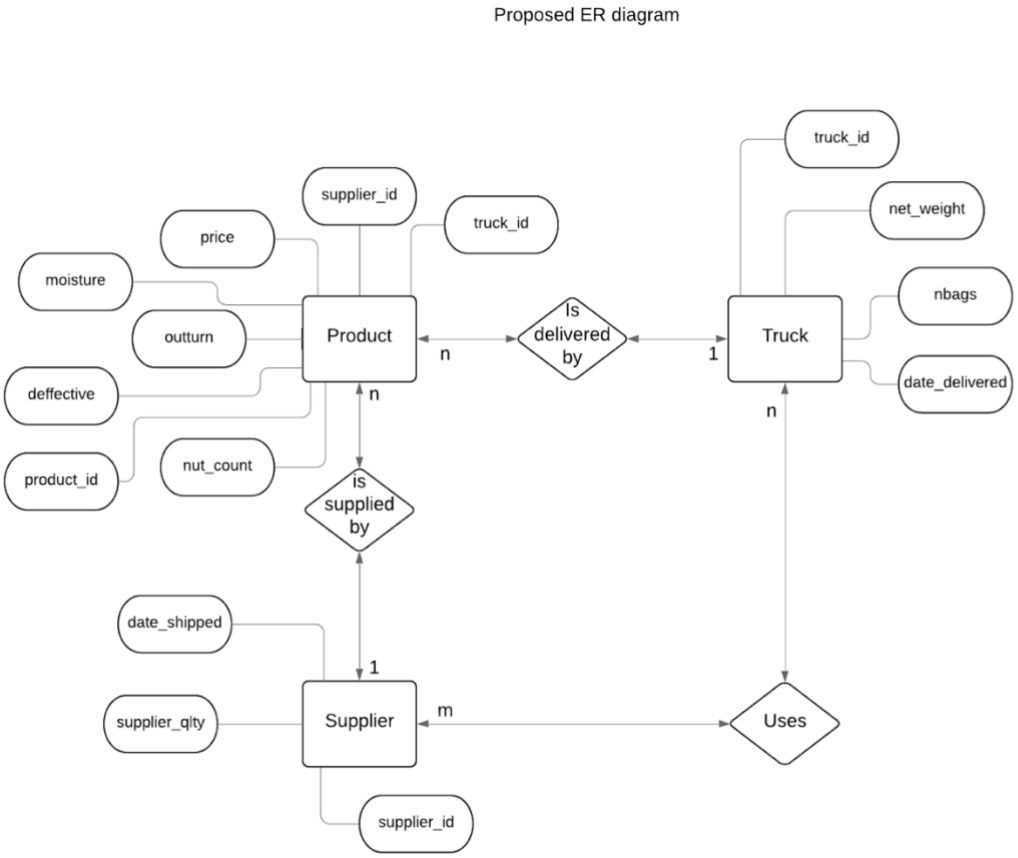
https://www.kaggle.com/extralime/cashew-truck-arrivals

Listed below are the attributes pertaining to the cashew truck delivery dataset:

- Date - Date of Arrival
- truckid - Vehicle Identification Number
- nbags - number of bags found in delivery
- net_weight - Net kg of cashew nuts (tare: truck weight, bag weight)
- origin - Origin ID of the cashew nuts (integer code)
- supplier - Supplier ID for supplier
- moisture - moisture % of cashew nuts
- nut_count - Number of raw cashew nuts per KG
- outturn - Quality metric (lbs of good cashew kernels per 80kg of raw cashew)
- defective - rate of defective kernels
- price - omitted
- year - year
- month - month
- day – day

The most important attributes are nut_count, supplier, moister, outturn, defective and net_weight.

Appendix 4: Proposed ER Diagram for the Cashew Nuts Dataset



Appendix 5: Iron Ore Attribute Description

The link to the dataset can be found below:

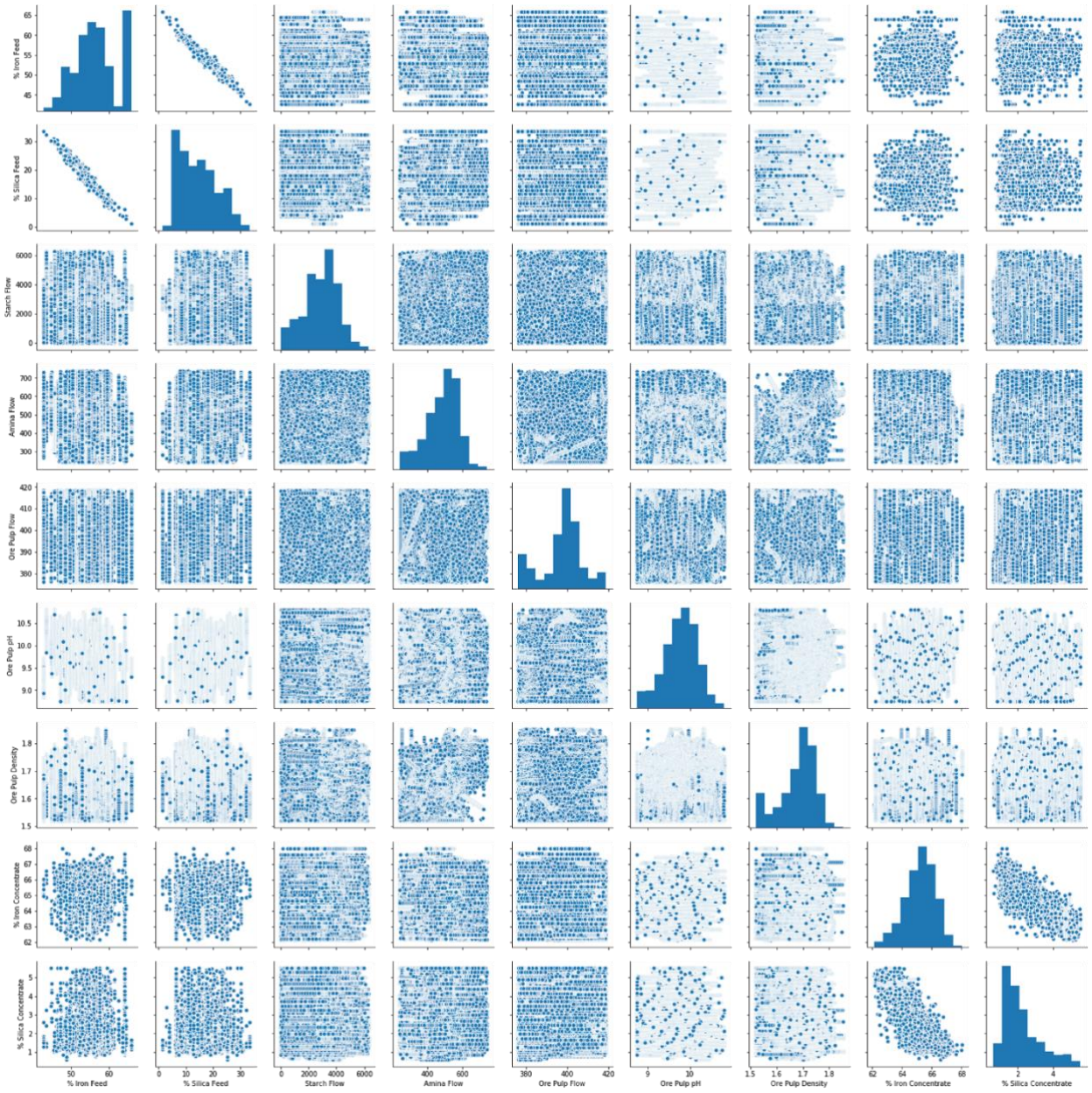
https://www.kaggle.com/edumagalhaes/quality-prediction-in-a-mining-process

The dataset columns are shown below with the column descriptions:

- Date: The date and timestamp for the manufacturing
- % Iron Feed: The percentage of Iron which is inserted into the flotation cells and are normally fetched from the Iron ore.
- % Silica Feed: The percentage of silica which is fed to the flotation cells that comes from the Iron ore and it is the impurity for this procedure.
- Starch Flow: The flow of starch in the flotation cells, measured in m_3/h.
- Amina Flow: The flow of amina in the flotation cells, measured in m_3/h.
- Ore Pulp Flow: The flow of Ore pulp during the iron ore production procedure
- Ore Pulp pH: The pH monitored on a scale from 0 to 14
- Ore Pulp Density: Density of the mixture on a scale from 1 to 3, measured in kg/cm_3
- Flotation Column Air Flow (01-07): This field measures the air flow that the flotation cell is provided during the procedure, measured in Nm_3/h.
- Flotation Column Level (01-07): This field measures the froth level that the flotation cell is provided during the procedure, measured in millimeters(mm).
- % Iron Concentrate: This is the percentage of Iron which represents how much iron is the end result of the flotation process. It is normally a lab measurement and represented as a percentage from 0 to 100%.
- % Silica Concentrate: This is the percentage of silica which points how much silica is there as the end result of the flotation process. It is also a lab measurement and represented as a percentage from 0 to 100%.

Appendix 6: Pairplot of Iron Ore Variable Correlations

Paired Scatterplots of % Iron Feed, % Silica Feed, Starch Flow, Amina Flow, Ore Pulp Flow, Ore Pulp ph, Ore Pulp Density, % Iron Concentrate and % Silica Concentrate. The diagonal features histograms of the respective variable.



Appendix 7: Kaggle Link to Iron Ore Production Code

The link to the Kaggle Kernel used to analyse the iron ore production dataset is listed below. The code present in the link is used for the report.

https://www.kaggle.com/mkoerner1/iron-mining-production-prediction