# Predicting on time shipping with Machine Learning

**Instructor**

**Xiaodi Zhu**

**Student**

**Anchal Patel**

**Erica Leonida**

**Thi Diem My Nguyen**

**Tiyanna Jenkins**

**FINC. 514 Introduction to Data Science**

**May 12, 2021**

## Table of Contents

**1. Introduction** **1**
**2. Data Summary** **1**
2.1. Data Source 1
2.2. Statistical summary and Correlation matrix 2
**3. Data Cleaning Process** **6**
**4. Methodology and Results** **6**
**5. Model Selection** **11**
**6. Conclusion and future works** **11**
6.1. Results and Analysis 11
6.2. Limitations of our approach 11
6.3. Potential future works 12
**Reference** **13**
**Appendix** **14**

## 1. Introduction

This research project focused on how to apply Machine Learning to make predictions regarding shipping on time, from an international e-commerce company to their customers. Using logistic regression and train/testing method with building three models to predict the shipping, we want to discover key insights from which factors significantly affect 'Reached on time' and predict whether a product will be delivered on time or not.

With the globalization of trade, transit time reliability has become a critical point in the shipping industry as irregularities will lead to more delays further down the supply chain. The company that sells electronic products provides freight forwarding services to its clients, offering them a complete set of supply chain solutions for shipping their goods across the world.

Using Machine Learning computing, we developed a model capable to predict on time shipping. Our model delivers predictions with a 68% accuracy as early as the transport is booked. The model relies on historical data, for example, cost of the product, product's weight, prior purchase, discount… We researched and tested several Machine Learning algorithms in order to select the one that maximizes the prediction accuracy score. This algorithm introduces a prediction component to the output by giving an estimated on time delivery to the customer for a shipment.

## 2. Data summary

### 2.1. Data source

We found the dataset coming from Kaggle. The data is publicly available on Kaggle in February 2021, it is not necessarily needed up to date. We have our data saved in a CSV file covering a total of 10,999 observation shipments, called Train(2).csv. We first read our dataset into a pandas data frame called dat, and then use the head() function to show the first five records from our dataset. To see data please check Appendix A at the end of the document.

Data variable definition

| | |
|---|---|
| ID | ID Number of Customers. |
| Warehouse block | The Company have big Warehouse which is divided in to block such as A,B,C,D,F |

| | |
|---|---|
| Mode of shipment | The Company Ships the products in multiple way such as Ship, Flight and Road |
| Customer care calls | The number of calls made from enquiry for enquiry of the shipment |
| Customer rating | The company has rated from every customer. 1 is the lowest (Worst), 5 is the highest (Best) |
| Cost of the product | Cost of the Product in US Dollars |
| Prior purchases | The Number of Prior Purchase |
| Product importance | The company has categorized the product in the various parameter such as low, medium, high |
| Gender | Male and Female |
| Discount offered | Discount offered on that specific product |
| Weight in gms | It is the weight in grams |
| Reached on time | It is the target variable, where 1 indicates that the product has not reached on time and 0 indicates it has reached on time |

The dataset contains 12 variables, we use 'Reach on time' as a target variable.

These feature variables will help us predict whether a shipment is on time or not: warehouse blocks, mode of shipment, cost of the product, prior purchases, product importance, gender, and weight in grams.

2.2. Statistical summary and Correlation matrix

We explore our data set to get a feel of what it looks like and get some insights into it.

Basic descriptive statistics of the number variables:

| **Variable** | **Cost of the product** | **Discount offered** | **Weight in gms** |
|---|---:|---:|---:|
| Observation | 10,999 | 10,999 | 10,999 |
| Mean | 210 | 13 | 3,634 |
| Median | 214 | 7 | 4,149 |
| Max | 310 | 65 | 7,846 |
| Min | 96 | 1 | 1,001 |

|  |  |  |  |
|---|---|---|---|
| STD | 48 | 16 | 1,635 |

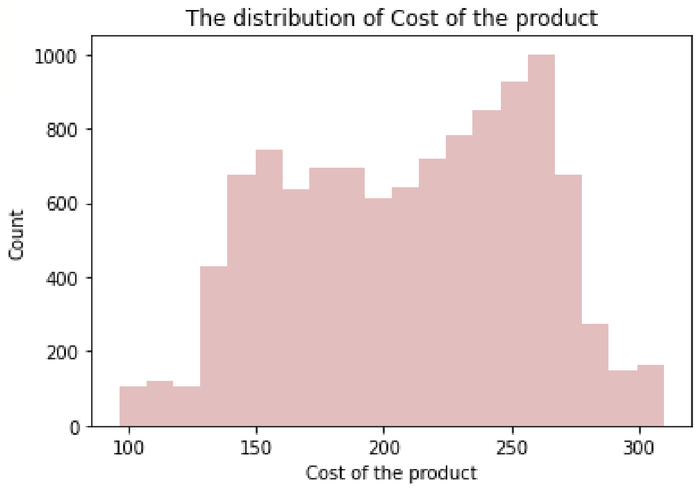


*Figure 1: Distribution of cost of the product*

Formula is being used:

Mean is the average of the total number. It is the most common and well-known method for measuring central tendency and can be used to handle both discrete and continuous data.

Mean= Average (A: A) / 2

(number1: number2) / 2

The standard deviation is a measure of the spread of the distribution. The larger the Std dev. the bigger the spread.

Standard Deviation = Stdev (A: A)

Minimum value is the smallest number of observation and lowest value in the row. As you can see in the table (a) minimum value of the Weight in grams is 1001 and cost of the product is 96.

Minimum Value = min (A: A)

Maximum Value is the largest number of observation and the highest value in the row. Highest value of Weight in grams is 7846 and cost of the product is 310.

Maximum Value = Max (A: A)

The mode is the value that appears most commonly in a data set. A set of data may have one mode, more than one mode, or no mode at all. Other popular measures of central tendency include the mean, or the average of a set, and the median, the middle value in a set. Median is defined as the middle of all numbers which is 4149 and 214.

Median= median (A: A)

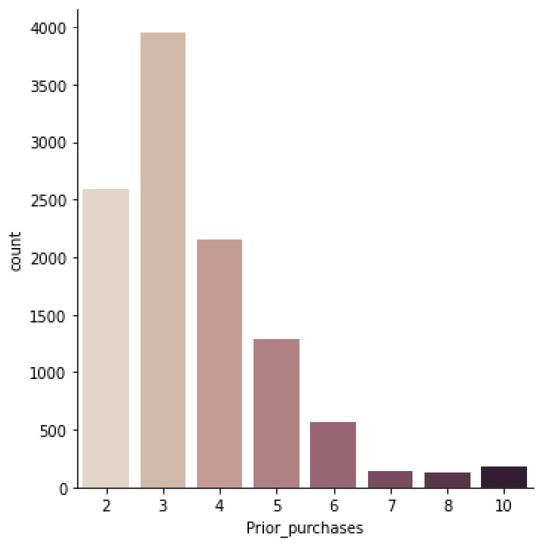


*Figure 2: Distribution of cost of prior purchases*

Let’s also look at how many shipments in the dataset are on time and how many are not. Below is the bar plot of the same:

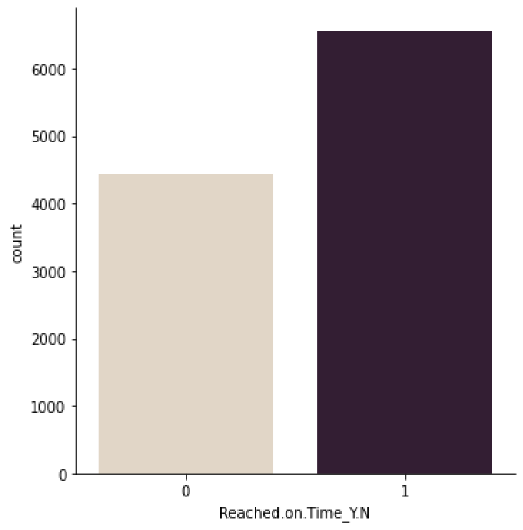
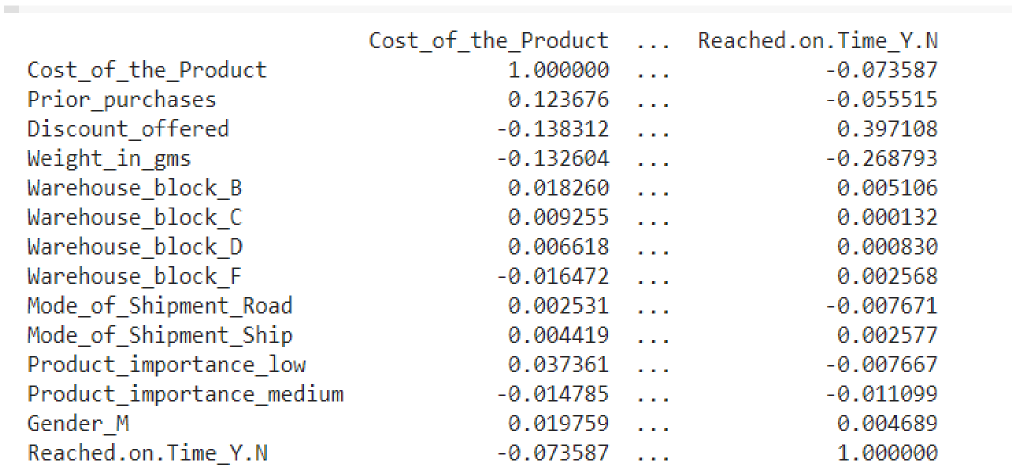


*Figure 3: Number of observations in each category Reached on time.*

Correlation matrix:

By using codes, we have found out the results for correlation matrix. sub_dat.corr() is being used for more information take a look at Appendix D.



```
                          Cost_of_the_Product  ...  Reached.on.Time_Y.N
Cost_of_the_Product                  1.000000  ...            -0.073587
Prior_purchases                      0.123676  ...            -0.055515
Discount_offered                    -0.138312  ...             0.397108
Weight_in_gms                       -0.132604  ...            -0.268793
Warehouse_block_B                    0.018260  ...             0.005106
Warehouse_block_C                    0.009255  ...             0.000132
Warehouse_block_D                    0.006618  ...             0.000830
Warehouse_block_F                   -0.016472  ...             0.002568
Mode_of_Shipment_Road                0.002531  ...            -0.007671
Mode_of_Shipment_Ship                0.004419  ...             0.002577
Product_importance_low               0.037361  ...            -0.007667
Product_importance_medium           -0.014785  ...            -0.011099
Gender_M                             0.019759  ...             0.004689
Reached.on.Time_Y.N                 -0.073587  ...             1.000000
```

By correlation matrix, we can find the correlation of every pair of features (and the outcome variable), and visualize the correlations using a heatmap.

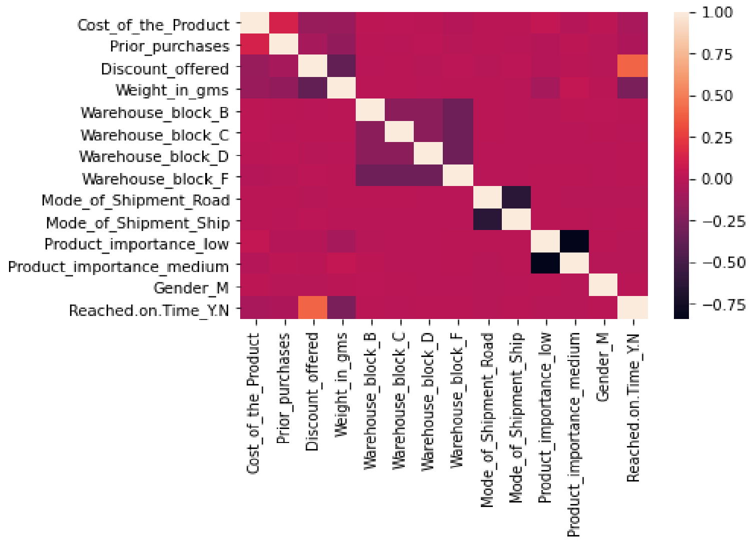


*Figure 4: Heatmap of feature (and outcome) correlations*

In the above heatmap, brighter colors indicate more correlation. As we can see from the table and the heatmap, Cost of the product, Prior purchases, Discount offered, and Weight in grams all have a significant correlation with the outcome variable. Also notice the correlation between pairs of features, like Gender and Prior purchases, or Weight in grams and Warehouse block.

It is also helpful to visualize relations between a single variable and the outcome. Below, we will see the relation between Cost of the product and Reached on time. Below is the relation between Prior purchases and Reached on time.

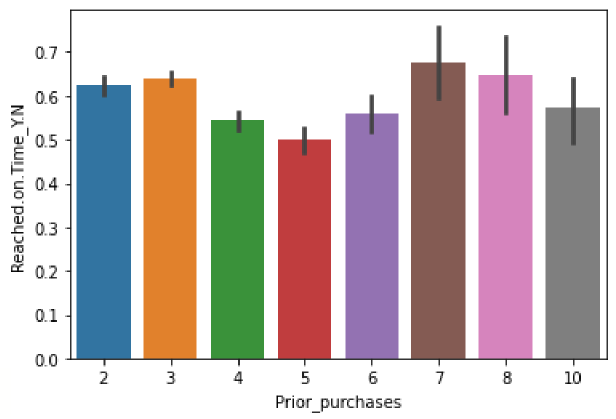


*Figure 5: Relations between a Prior purchases and Reached on time.*

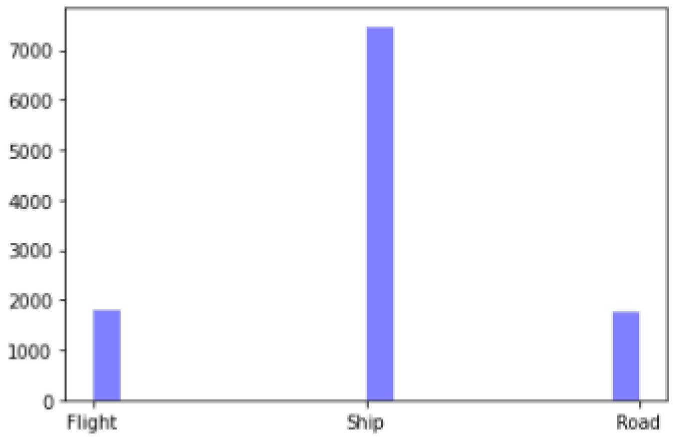


*Figure 6: Mode of shipment*

The above histogram is based on the mode of the shipment. According to the Figure 6 Three modes of shipment are being used, i.e., Flight, Ship, Road. The Ship is the largest in use to transport goods from one place to another.

## 3. Data cleaning process

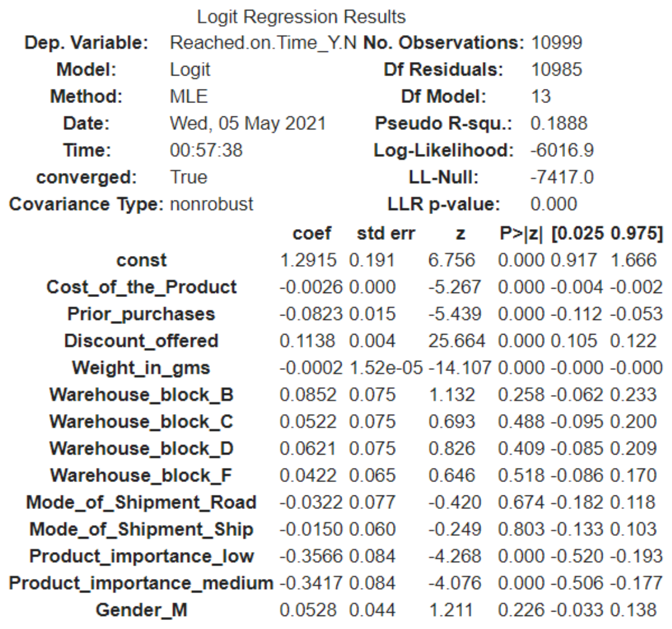
The data is clean (has no null value) so we don't need to do the cleaning process. For more information on what codes we have used take look at Appendix E.

## 4. Methodology & Results

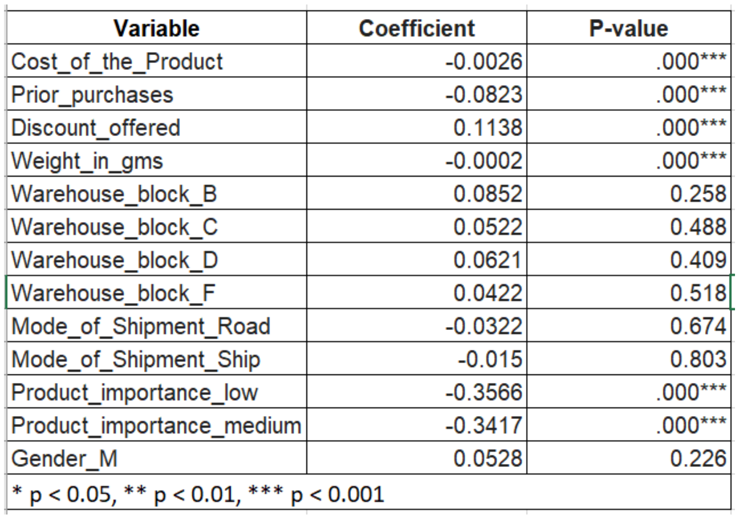
4.1. To answer the research question about which factor(s) significantly affect reach on time, we used Logistic Regression model. Logistic Regression model describes and estimates the relationship between a dependent variable ***Y***, which takes only two possible values, resulting from the occurrence or not of an event, and independent variables affecting that phenomenon. Its popularity is supported by the lack of necessity to meet many requirements for linear regression and general linear models, which include the linearity of the relationship between a dependent and an independent variable, as well as normality and homoscedasticity of the distribution of independent variables or the necessity to use the variables given using the metric system of measures.

In this project, we used Reached on time as a dependent variable Y and warehouse blocks, mode of shipment, cost of the product, prior purchases, product importance, gender, and weight in grams as independent variables codes are included at the bottom of the document check Appendix F.

The outcome allows to estimate the parameters of the logistic regression model, the values of which are presented in the table.



```
                         Logit Regression Results
     Dep. Variable:  Reached.on.Time_Y.N No. Observations: 10999
            Model:   Logit                   Df Residuals:  10985
           Method:   MLE                       Df Model:    13
             Date:   Wed, 05 May 2021      Pseudo R-squ.:   0.1888
             Time:   00:57:38              Log-Likelihood:  -6016.9
        converged:   True                         LL-Null:  -7417.0
 Covariance Type:    nonrobust               LLR p-value:   0.000
                                coef   std err     z     P>|z| [0.025  0.975]
           const               1.2915  0.191     6.756   0.000  0.917   1.666
    Cost_of_the_Product       -0.0026  0.000    -5.267   0.000 -0.004  -0.002
      Prior_purchases         -0.0823  0.015    -5.439   0.000 -0.112  -0.053
      Discount_offered         0.1138  0.004    25.664   0.000  0.105   0.122
       Weight_in_gms          -0.0002  1.52e-05 -14.107  0.000 -0.000  -0.000
     Warehouse_block_B         0.0852  0.075     1.132   0.258 -0.062   0.233
     Warehouse_block_C         0.0522  0.075     0.693   0.488 -0.095   0.200
     Warehouse_block_D         0.0621  0.075     0.826   0.409 -0.085   0.209
     Warehouse_block_F         0.0422  0.065     0.646   0.518 -0.086   0.170
   Mode_of_Shipment_Road      -0.0322  0.077    -0.420   0.674 -0.182   0.118
   Mode_of_Shipment_Ship      -0.0150  0.060    -0.249   0.803 -0.133   0.103
   Product_importance_low     -0.3566  0.084    -4.268   0.000 -0.520  -0.193
Product_importance_medium     -0.3417  0.084    -4.076   0.000 -0.506  -0.177
          Gender_M             0.0528  0.044     1.211   0.226 -0.033   0.138
```



| Variable | Coefficient | P-value |
|---|---|---|
| Cost_of_the_Product | -0.0026 | .000*** |
| Prior_purchases | -0.0823 | .000*** |
| Discount_offered | 0.1138 | .000*** |
| Weight_in_gms | -0.0002 | .000*** |
| Warehouse_block_B | 0.0852 | 0.258 |
| Warehouse_block_C | 0.0522 | 0.488 |
| Warehouse_block_D | 0.0621 | 0.409 |
| Warehouse_block_F | 0.0422 | 0.518 |
| Mode_of_Shipment_Road | -0.0322 | 0.674 |
| Mode_of_Shipment_Ship | -0.015 | 0.803 |
| Product_importance_low | -0.3566 | .000*** |
| Product_importance_medium | -0.3417 | .000*** |
| Gender_M | 0.0528 | 0.226 |
| * p < 0.05, ** p < 0.01, *** p < 0.001 | | |

Parameters of the logistic regression model and their assessment:

+ Parameters turned out to be statistically significant: Cost of the product, Prior purchase, Discount offered, Weight in grams, Product importance.

+ Parameters did not turn out to be statistically significant: Warehouse block, Mode of shipment, Gender.

4.2. To answer the research question about how to use Machine Learning to predict whether a shipment will be delivered on time or not, we apply the training/testing method with three models Logistic Regression, Decision Tree, and SVM. After running the models, we will choose and pick the best model base on the accuracy score.

We selected statistically significant variables to tie our model. To test and validate the performance of the different algorithms, we split the data in a training set and a test or validation set. All algorithms were trained and tested on the same training and test set so that we could compare their performance. Appendix G shows how we used the test and train method using Google Colab.

Then we apply three model's Logistic regression, Decision tree, SVM.

We used Python programming language to write the code for our models. To create, train and use the final models for prediction, there are three relevant scripts in the final program. The first script is used to clean the data, remove missing values and transform all columns to the right format, our dataset has no missing value, so we don't do this part. The second script automatically reads in all the relevant data, trains (on new data or adding data to an existing model), and saves the final model. The third script is used for prediction in a production environment.

To predict on time shipping, we apply on three model's Logistic regression, Decision tree, SVM.

➢ Apply Logistics regression model:

About logistic regression: Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, Logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables. Appendix H shows the code we use to fit the model.

Accuracy Score for LR model: 0.6403636363636364. The model makes 64% accuracy rate on the testing dataset.

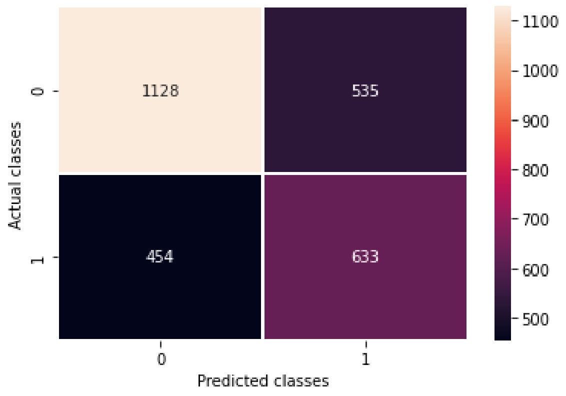


*Figure 7: Confusion matrix for logistic regression model*

➢ Apply Decision Tree model:

About decision tree: A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

In our decision tree model, we have 5 classes with 6 input variables, and we use Reached on time as the output variable. The code we used to fit the model at Appendix I.

Accuracy Score for Decision Tree model: 0.6832727272727273

The model makes 68% accuracy rate on the testing dataset.

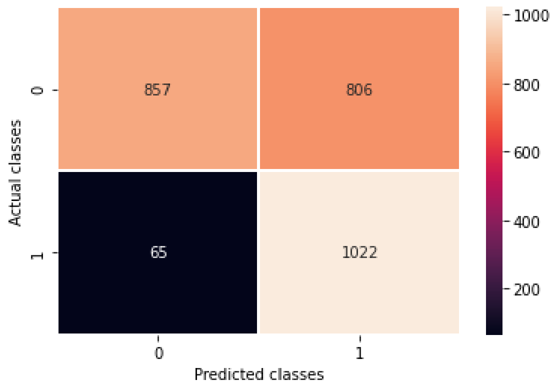


*Figure 8: Confusion matrix for Decision tree model*

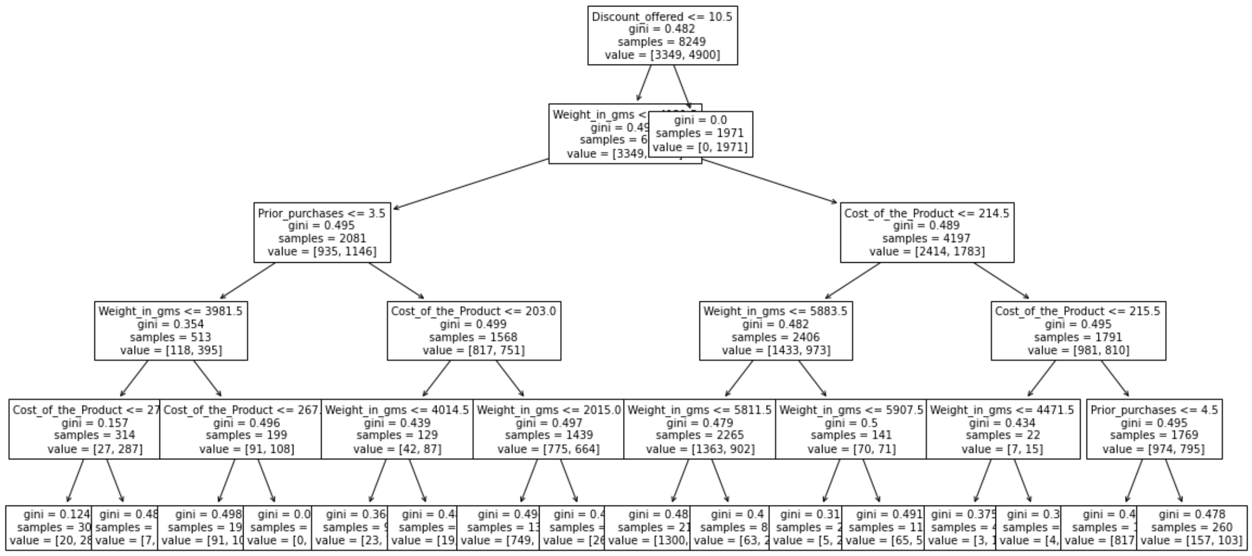


*Figure 9: Plot of Decision Tree*

➢ Apply SVM model

About SVM: A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model set of labeled training data for each category, they can categorize new text. Appendix J describes the code in detail we have used to fit the model.

Accuracy Score for SVM model: 0.6432727272727272. The model makes 64% accuracy rate on the testing dataset.

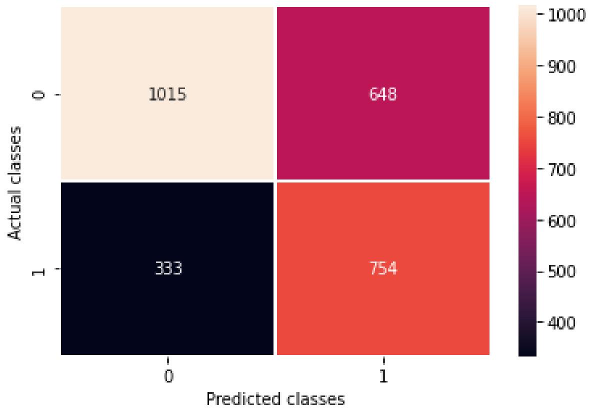


*Figure 10: Confusion matrix for SVM model*

## 5. Model selection

| Model | Accuracy Score | Type I error |
|---|---:|---:|
| Logistic Regression | 64% | 454 |
| Decision Tree | 68% | 65 |
| SVM | 64% | 333 |

Compare three models, we see that Decision Tree model has the highest accuracy score and lowest type I error. We will choose this model to predict the on time shipping on our dataset.

## 6. Conclusion and future works

In this section, we will discuss the results as well as the performance of the models. We will evaluate results and performance from an operation point of view as well as from a modelling standpoint. The latter will explore the limitations of using such models to make predictions as opposed to using “simple” and more traditional approaches.

### 6.1. Results and Analysis

➢ Relationship between feature variables and target variables.

+ Cost of the product, Prior purchases, Discount offered, Weight in gram and Product important function have a significant influence on the model, especially Cost of the product and Prior purchases. It is good to see our machine learning model match what we have been assuming.

+ Cost of the product has a negative influence on the prediction, higher Cost of the product is correlated with a not on time shipment, higher on Cost of the product, less on not on time shipment.

➢ Predict on time shipment:

After comparing the results yielded by three different algorithms (Decision Tree, Logistic Regression, and SVM), we decided to build the final prediction model with a Decision Tree algorithm. We selected the set of Decision Tree model as they result in the highest accuracy score (68%) and the least type I error (65), depending on the model segments.

Not only does the Decision Tree model performs better on predicting the target variable, it is also easier to train and implement in a production system. The time required to train the Decision Tree model is less than for the SVM.

### 6.2. Limitations of our approach

Machine Learning is a method that is rather data-intensive in two ways: It requires a lot of data in order to kick-start the analysis and modeling and it requires that this data be of at least moderate and at best great quality. For great quality to be achieved, this means there should be no missing or wrong data points in the dataset, as well as consistent and useable formatting of the data. If not, the length of the data cleaning part of the project might be considerably extended, which could pose a timing problem, especially if the project is bound in time.

Furthermore, developing such a model demands analytics and coding skills. These two skills, even if required, are not enough: having subject-matter experts providing input on the industry practices and interpreting results and data is crucial to the success of such an endeavor.

Looking further down the road, since the end goal of a project like ours is to supply a functioning tool, this means the company will have to work on the models to make them a part of their computing environment. For example, the models we are delivering draw their data from several Excel files but in the future, we can imagine a direct link to a database instead.

## 6.3. Potential future works

In this study, we limited our scope to electronic products, but the same models developed for this purpose could be used for any products, given that the models are trained and supplied with the appropriate data. Consequently, the relevant factors we have identified as impactful for predicting on time shipping, such as cost of the product or prior purchases, in these models are relevant for any other products.

Furthermore, another insight of our study is that our approach could be applied to any shipping industry. We could envision a similar project being conducted for the other industry for example. As long as the necessary data is available and the impactful factors can be identified, the method can be adapted for any field of industry.

In the future, it would be worth investigating the possibility of including weather patterns in the model to improve its accuracy, if access to reliable data can be guaranteed. Another way to improve the model would be to know the area of customers. This is the area of the customers where products will be delivered to. We are confident that this would increase the accuracy of the model even more.

## Reference

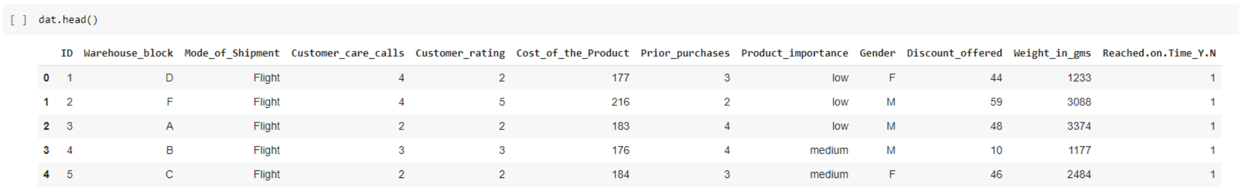
Charles, A., & Jonquais, J., & Krempl, F. (2019). Predicting Shipping Time with Machine Learning. In *The Program in Supply Chain management*.

Borucka, A. (2020). Logistic regression in modeling and assessment of transport services. *Open Engineering*, 10(1), 26-34.

Gopalani, P. (2021). Product Shipment Delivered on time or not? To Meet E-Commerce Customer Demand. In *https://www.kaggle.com/prachi13/customer-analytics*.

Li, S. (2017). Building A Logistic Regression in Python, Step by Step. In *https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8*.

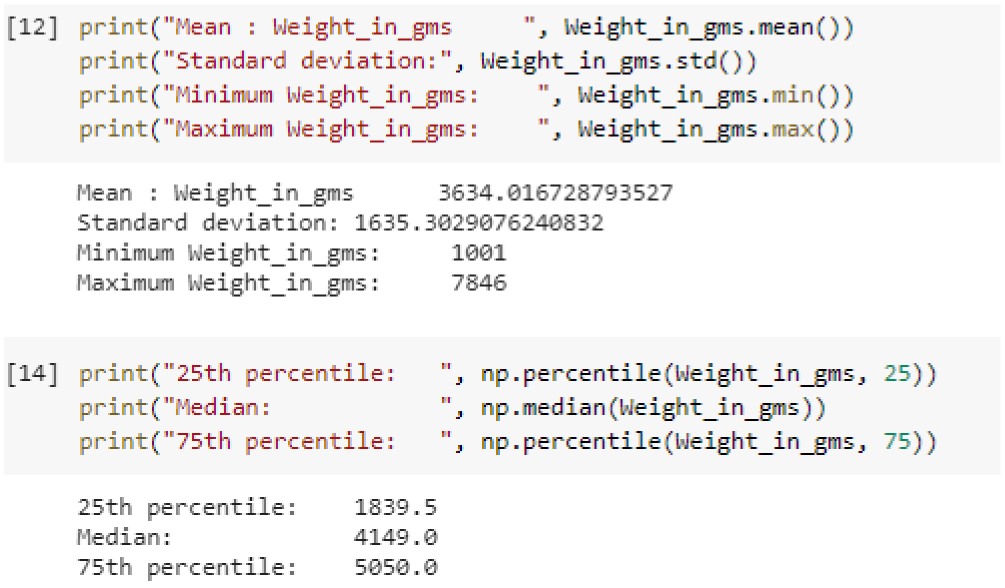
# Appendix



```
[ ] dat.head()
```

| | ID | Warehouse_block | Mode_of_Shipment | Customer_care_calls | Customer_rating | Cost_of_the_Product | Prior_purchases | Product_importance | Gender | Discount_offered | Weight_in_gms | Reached.on.Time_Y.N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | D | Flight | 4 | 2 | 177 | 3 | low | F | 44 | 1233 | 1 |
| **1** | 2 | F | Flight | 4 | 5 | 216 | 2 | low | M | 59 | 3088 | 1 |
| **2** | 3 | A | Flight | 2 | 2 | 183 | 4 | low | M | 48 | 3374 | 1 |
| **3** | 4 | B | Flight | 3 | 3 | 176 | 4 | medium | M | 10 | 1177 | 1 |
| **4** | 5 | C | Flight | 2 | 2 | 184 | 3 | medium | F | 46 | 2484 | 1 |

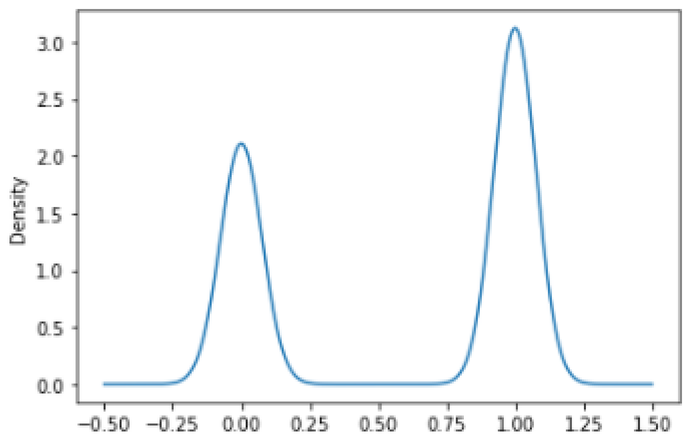
*Appendix A: Data Source Excel Sheet.*



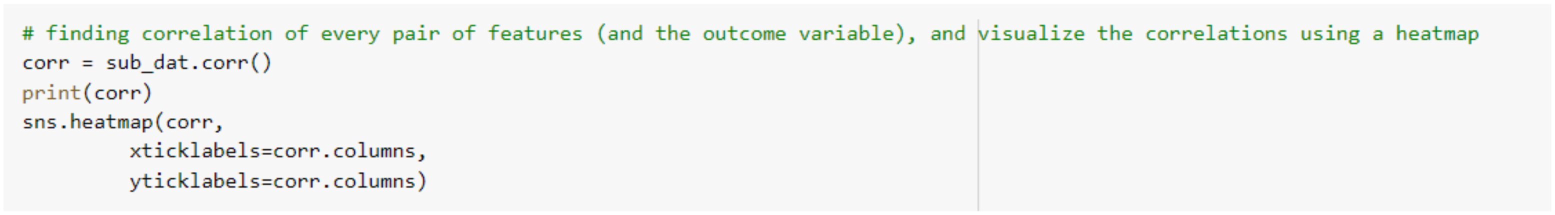
```
[12] print("Mean : Weight_in_gms     ", Weight_in_gms.mean())
     print("Standard deviation:", Weight_in_gms.std())
     print("Minimum Weight_in_gms:    ", Weight_in_gms.min())
     print("Maximum Weight_in_gms:    ", Weight_in_gms.max())

     Mean : Weight_in_gms      3634.016728793527
     Standard deviation: 1635.3029076240832
     Minimum Weight_in_gms:     1001
     Maximum Weight_in_gms:     7846

[14] print("25th percentile:    ", np.percentile(Weight_in_gms, 25))
     print("Median:             ", np.median(Weight_in_gms))
     print("75th percentile:    ", np.percentile(Weight_in_gms, 75))

     25th percentile:    1839.5
     Median:             4149.0
     75th percentile:    5050.0
```

*Appendix B: Shows the calculations by using code. Percentile for weight in grams*

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a6ce79410>
```

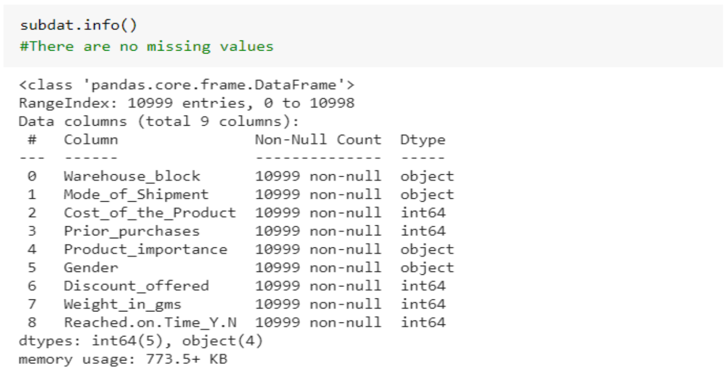


*Appendix C: Shows the density of the product delivery.*



```python
# finding correlation of every pair of features (and the outcome variable), and visualize the correlations using a heatmap
corr = sub_dat.corr()
print(corr)
sns.heatmap(corr,
        xticklabels=corr.columns,
        yticklabels=corr.columns)
```

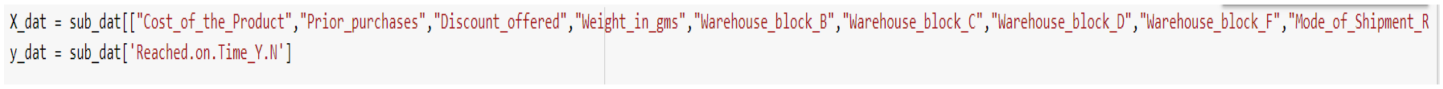
*Appendix D: Correlation Matrix using codes.*



```python
subdat.info()
#There are no missing values
```

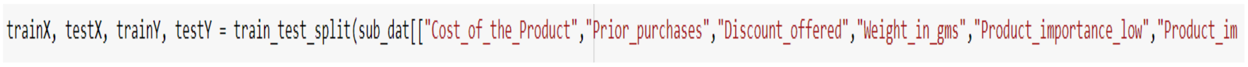
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   Warehouse_block      10999 non-null  object
 1   Mode_of_Shipment     10999 non-null  object
 2   Cost_of_the_Product  10999 non-null  int64 
 3   Prior_purchases      10999 non-null  int64 
 4   Product_importance   10999 non-null  object
 5   Gender               10999 non-null  object
 6   Discount_offered     10999 non-null  int64 
 7   Weight_in_gms        10999 non-null  int64 
 8   Reached.on.Time_Y.N  10999 non-null  int64 
dtypes: int64(5), object(4)
memory usage: 773.5+ KB
```

*Appendix E: Cleaning process*

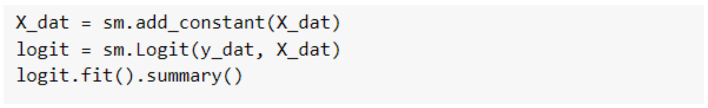


```python
X_dat = sub_dat[["Cost_of_the_Product","Prior_purchases","Discount_offered","Weight_in_gms","Warehouse_block_B","Warehouse_block_C","Warehouse_block_D","Warehouse_block_F","Mode_of_Shipment_R
y_dat = sub_dat['Reached.on.Time_Y.N']
```

*Appendix F: Summary of the data*

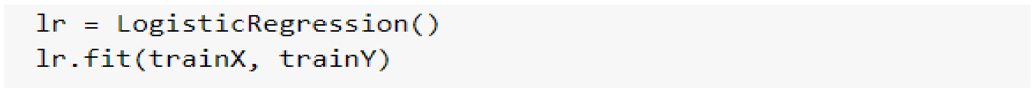


```python
trainX, testX, trainY, testY = train_test_split(sub_dat[["Cost_of_the_Product","Prior_purchases","Discount_offered","Weight_in_gms","Product_importance_low","Product_im
```



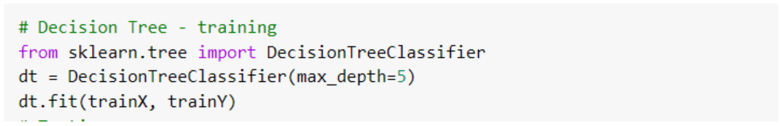
```python
X_dat = sm.add_constant(X_dat)
logit = sm.Logit(y_dat, X_dat)
logit.fit().summary()
```

*Appendix G: Train and Test Method*



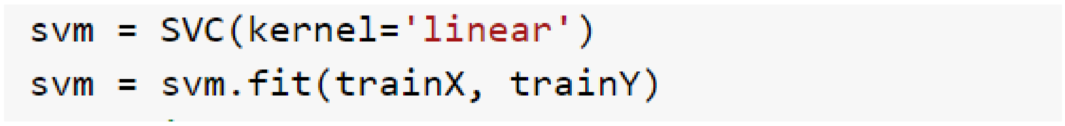
```python
lr = LogisticRegression()
lr.fit(trainX, trainY)
```

*Appendix H: Regression Fit model*



```
# Decision Tree - training
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(max_depth=5)
dt.fit(trainX, trainY)
```

*Appendix I: Decision Tree Model*



```
svm = SVC(kernel='linear')
svm = svm.fit(trainX, trainY)
```

*Appendix J: SVM Model*