

# DataCo Supply Chain Company Data Analysis

Mu Cheng, Shimin Zhang, Yaokai Yu

December 15, 2022

## I. Abstract

In this project, we worked on the DataCo Supply Chain dataset to explore patterns in its transaction records in between 2015 and 2018. This dataset has enough features to extract useful business operation, regional market, and product information by using data mining techniques. Moreover, the features are also factors that could affect the final delivery results and can be used to solve classification problems. Thus, it is a great dataset to practice data science skills. We found this supply chain company's operational efficiency through exploring the relationships between Shipping Modes and Benefits, Predict Risk of Late Delivery for each shipping mode, and etc. We looked into the regional business information by making charts and graphs showing the most popular commodities, top markets, and a world map that provides information showing the routes that specific shipments traveled. Lastly, we helped our client DataCo Supply Chain Company to make predictions for the delivery status based on their locations, the products category, the shipping mode and where the products are shipped from. The decision Tree Model was picked as the best machine learning model to predict risk of delivery based on shipping address, product category, shipping status, and destination address, and a streamlit-based user interface was then built for the DataCo Supply Chain company to manage their business operations.

## II. Introduction of the DataCo Supply Chain Dataset

It's a huge dataset that contains 53 features which provide enough information for us to explore business problems and apply machine learning models to make predictions. There are 6 types of information defined by the features. From the order sales and order benefits, we're able to calculate its business revenues. The customer information provides us location, name, id, and email address that can be used to find VIP customers who made the most orders in this project. In addition, the shipment information is stored in features like product categories and shipping mode (whether it's arranged as first class or standard shipping). Then delivery status and risk delivery provide information for us to find how time of delivery was affected by other factors. Moreover, location data tells us the places where the shipment order is made, and where the shipment will be delivered to.

In order to find important business insights by using data mining techniques, we handpicked features to get more accurate results.

## 1. Dataset and Preprocessing

Before the analysis, we need to discard those features that are identical (data in this feature is all the same), repetitive semantically (other features' data include this features' data's info), repetitive in data (other features' data are totally the same as this one), having too many missing values and invalid data, etc. We did this process both in a coding way and in a hand picked way.

We first tried to find out those invalid features from coding. For example, we deleted the feature 'Customer Password', since all data is identical in xxxxxxxx; we deleted 'Product Status' since it has too many invalid values, mixing up data in string, numbers and time format. In this step, we deleted those features that are identical in data and have too many invalid values.

```
#explore the distribution of each features' data
for i in data_total.columns:
    print(data_total[i].value_counts())
```

```
XXXXXXX    180519
Name: Customer Password, dtype: int64
```

```
0    172934
Standard Class    1859
Second Class     512
First Class      407
Same Day         226
...
3/12/2015 2:05    1
3/16/2015 14:26   1
3/19/2015 9:10    1
3/29/2015 21:04   1
3/8/2016 21:28    1
Name: Product Status, Length: 1572, dtype: int64
```

Then we tried to drop features that had too much data in null, which includes 'Order Zipcode' and 'Product Description'.

```
# Check which features has too many null values
data_total_copy.info()
# drop the following features:
# Product Description
# Order Zipcode
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180519 entries, 0 to 180518
Data columns (total 39 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Type                                  180519 non-null  object
1   Days for shipping (real)              180519 non-null  int64

35  Order Zipcode                        32425 non-null   float64
36  Product Card Id                     177515 non-null   float64
37  Product Description                  7585 non-null    object
38  Shipping Mode                       172934 non-null   object
```

## CS 6220 · DataCo Supply Chain Company Data Analysis

After the above process, we handpicked features that are most valuable. For example, we kept Latitude, Longitude, dropped Customer State, Customer Street, and Customer Zipcode, since they are repetitive semantically. Latitude and Longitude are more valuable in data-mining and machine learning. We kept Benefit per order, Sales per customer, and dropped Order Item Discount and Order Item Discount Rate. Since data, in Order Item Discount and Order Item Discount Rate features, are in much smaller numbers, and from the business perspective, focusing on profit, exploring whether they have a right pricing, operating strategy, is much more important than whether they have a right discount to the orders or not.

In the end, we kept 20 features to do the data-mining and machine learning, which includes: Days for shipping (real), Days for shipment (scheduled), Benefit per order, Sales per customer, Delivery Status, Late\_delivery\_risk, Category Id, Category Name, Customer City, Customer Country, Customer Fname, Customer Id, Customer Lname, Customer Segment, Latitude, Longitude, Market, Order City, Order Country, Shipping Mode.

To figure out whether the orders arrive in advance, on-time or late, we created a new feature called 'Days of Shipping(schedule-real)', which is the result of features 'Days for shipment (scheduled)' minus 'Days for shipping (real)'. In feature 'Days of Shipping(schedule-real)', if the data shows negative, it means the order arrives late, and the number shows how many days it was late for(e.g. '-4' in the data means the order is late for 4 days). On the other hand, if the data shows 0, it means it arrives on time; and if the data shows positive, it means the arrives in advance, and the number shows how many days it was in advanced for(e.g. '2' in the feature means the order is in advance for 2 days).

### **2. Data Analysis**

As part of DataCo Supply Chain Company, or a member of the supply chain industry, we would be interested in the customer satisfaction and profit we make. The feature most related to customer satisfaction shown in the dataset is whether the orders arrive on time or even sooner. So we predicted the risk of being late and profits will be made and filtered the result in different shipping modes, to see if there is any large difference in these areas between different shipping modes. If there exists a large difference in risks and benefits, that means in some specific shipping mode, the operations management needs to be improved, pricing strategy needs to be modified.

When we predict 'Benefit per order' and 'Late\_delivery\_risk', we have taken out all the data in features 'Benefit per order', 'Late\_delivery\_risk' from original data to train those two features, and set the test size = 0.25.

## CS 6220 · DataCo Supply Chain Company Data Analysis

To predict 'Benefit per order', after modifying original data and setting the test size, we used `XGBRegressor()` to do the prediction, since this is a good regression model. We connected `X_test` – the training data's feature 'Shipping Mode', and `y_pred` – the result of predicting feature 'benefit per order' to form a new table below ('Same Day' : 0, 'First Class' : 1, 'Second Class' : 2, 'Standard Class' : 3):

Shipping Mode		benefit
0	1	12.362829
1	0	24.697218
2	1	4.615931
3	3	42.198467
4	3	14.614036
...	...	...
43229	3	17.137531
43230	3	3.462457
43231	1	13.493263
43232	1	10.646018
43233	2	-25.004829

43234 rows × 2 columns

Then we filtered the prediction with different shipping mode:

Shipping Mode	benefit							
	count	mean	std	min	25%	50%	75%	max
0	2287.0	19.873358	25.891947	-261.106689	10.128365	17.278872	27.663382	396.789337
1	6665.0	21.587572	28.575319	-682.516479	10.721028	18.032160	30.197050	543.202271
2	8526.0	21.422237	29.434847	-883.504028	10.595853	17.966870	29.533969	711.853638
3	25756.0	22.010874	34.915741	-1813.129028	10.984971	18.207900	30.275338	918.663513

According to the mean value calculated in different shipping modes, the standard class has the highest benefit, which is 19.87 and the same day class has the lowest benefit, which is 22.01. But the difference of getting the highest and lowest benefit in different shipping modes per order is only \$2.14, which is not large. So there are no lag-offs in operations management and pricing strategy between different shipping modes.

To predict the feature 'Late\_delivery\_risk'(data in this feature is either 1 or 0), we used the same strategy as predicting the feature 'Benefit per order', but used another model called `LogisticRegression()`, since this is a good classification model. We connected `X_test` – the training data's feature 'Shipping Mode', and `y_pred` – the result of predicting feature

## CS 6220 · DataCo Supply Chain Company Data Analysis

'Late\_delivery\_risk' to form a new table, then filtered it in different shipping modes. Here is the result ('Same Day' : 0, 'First Class' : 1, 'Second Class' : 2, 'Standard Class' : 3):

		risk_rate							
		count	mean	std	min	25%	50%	75%	max
Shipping Mode									
0	2272.0	0.642165	0.479469	0.0	0.0	1.0	1.0	1.0	1.0
1	6551.0	0.726912	0.445580	0.0	0.0	1.0	1.0	1.0	1.0
2	8383.0	0.804843	0.396345	0.0	1.0	1.0	1.0	1.0	1.0
3	26028.0	0.504995	0.499985	0.0	0.0	1.0	1.0	1.0	1.0

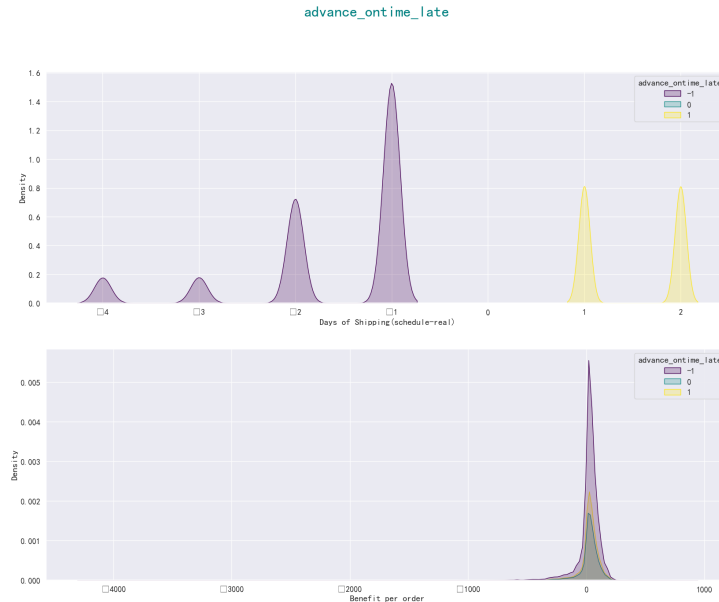
The second class has the highest risk of being late, which is 0.80, and the standard class has the lowest risk of being late, which is 0.50. From the original data, the feature of 'Late\_delivery\_risk' is either 1 or 0, so actually all shipping modes tend to be late since they are all closed to 1.

To further explore the relationships between shipping status and profits, we created a new feature called 'advance\_ontime\_late', which transferred data from different numbers in features 'Days of Shipping(schedule-real)' into 1, 0 and -1, which enable numbers indicated as in advance, on time and late respectively (positive numbers transferred into 1, negative numbers transferred into -1, and 0 remains the same). The standard deviation of 'advance\_ontime\_late' became smaller than 'Days of Shipping(schedule-real)' as shown below, which helped with our analysis.

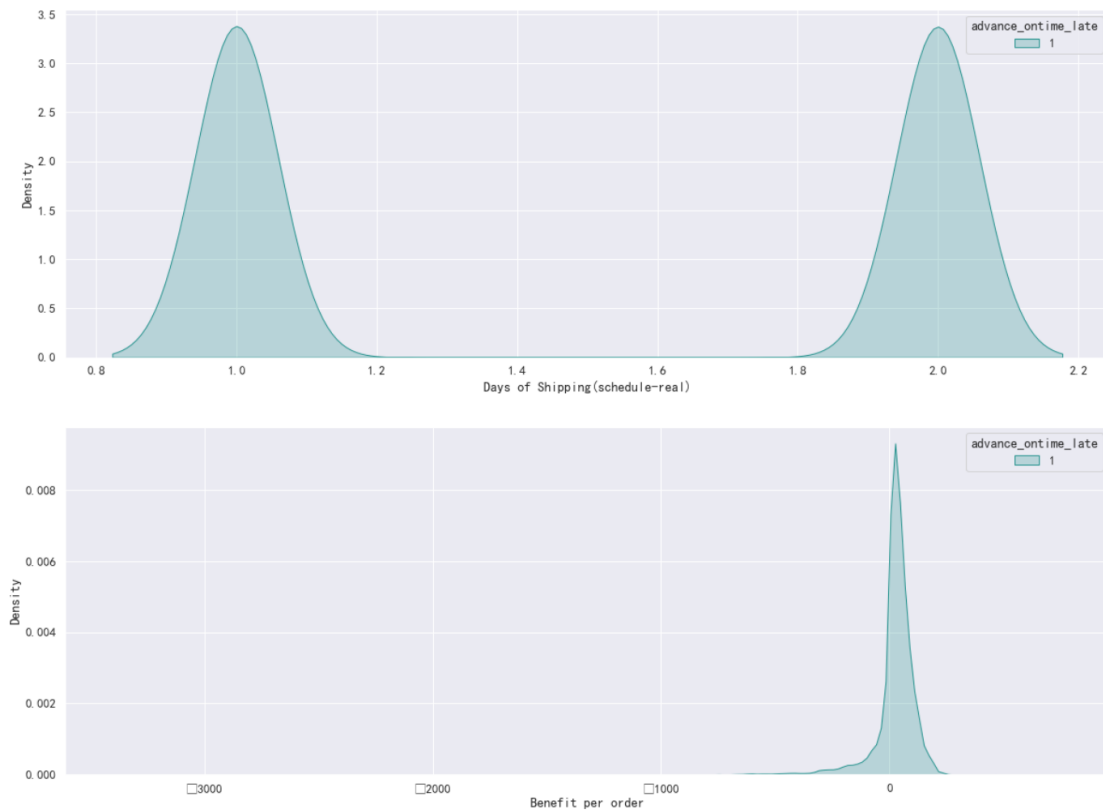
	Days of Shipping(schedule-real)	advance_ontime_late	Benefit per order
count	180519.000000	180519.000000	180519.000000
mean	-0.565807	-0.332563	21.974989
std	1.490966	0.838110	104.433526
min	-4.000000	-1.000000	-4274.979980
25%	-1.000000	-1.000000	7.000000
50%	-1.000000	-1.000000	31.520000
75%	0.000000	0.000000	64.800003
max	2.000000	1.000000	911.799988

To have a general ideas of the trend and relationships between features 'Days of Shipping(schedule-real)', 'advance\_ontime\_late' and 'Benefit per order', we made the density plots below, which shows the distribution of feature 'Days of Shipping(schedule-real)' and 'Benefit per order' when shipping is in advance, on-time and late.

## CS 6220 · DataCo Supply Chain Company Data Analysis

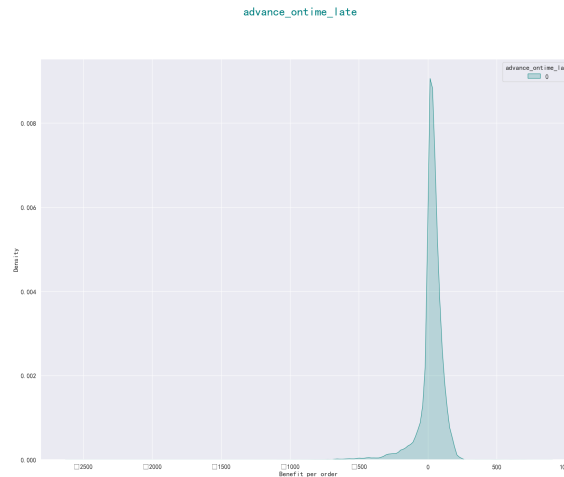


When shipping arrives in advance, most of the shipping status is in advance for 1 or 2 days, and 81.33% of those transactions' benefit is non-negative:



When shipping arrives on time 81.43% of those transactions' benefit is non-negative:

## CS 6220 · DataCo Supply Chain Company Data Analysis



When shipping arrives late, Most of the shipping status is late for 1 to 4 days, and 80.57% of those transactions' benefit is non-negative:



As we can see, no matter what shipping status is, around 81% of the transactions were making profits. Then we explored the relationships between shipping status(advance/on time/late) and making profit(whether benefit per order  $\geq 0$ ) in benefit per order's perspective.

We generated a new feature "benefit\_get" (if benefit  $\geq 0$ , benefit\_get = 1; if benefit  $< 0$ , benefit\_get = 0). First, we generated a density plot showing the distribution of shipping status in advance, on-time and late, and the distribution of benefit\_get is equal to 1 and 0:

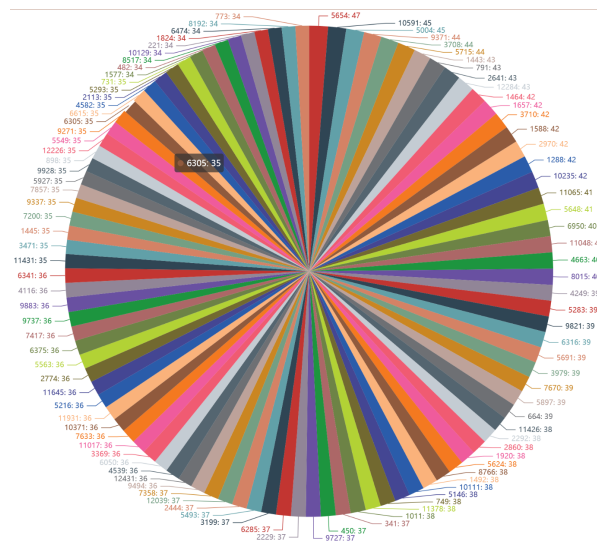
## CS 6220 · DataCo Supply Chain Company Data Analysis



We also explored the relationships between benefit per order and shipping result from calculating percentages. When benefit per order is non-negative, the percentage of shipping is in advance, on-time and late are 24.0%, 8.7% and 57.2% respectively; when benefit per order is negative, the percentage of shipping is in advance, on-time and late are 24.0%, 18.6% and 57.5% respectively (adding up those percentages is not 100% because there is invalid data besides in advance, on time and late). The distribution of shipping status is really similar no matter if the orders are profitable or not. It looks like the company is making a benefit from late orders from the density plot itself. But if we combine what we found out, all shipping statuses are 81% of the orders profitable, there is no relationship between making profit and shipping status.

We tried to find those VIPs from the orders. There are not any customers that really stand out from the group(chart left below), so we did a pie chart for the top 100 customers(pie chart right below). We could see that the difference of number of orders between most and least frequent buyers in the top 100 customers is only 13.

	ID	times	rate
0	5654	47	0.0260%
1	10591	45	0.0249%
2	5004	45	0.0249%
3	9371	44	0.0243%
4	3708	44	0.0243%
5	5715	44	0.0243%
6	1443	43	0.0238%
7	791	43	0.0238%
8	2641	43	0.0238%
9	12284	43	0.0238%



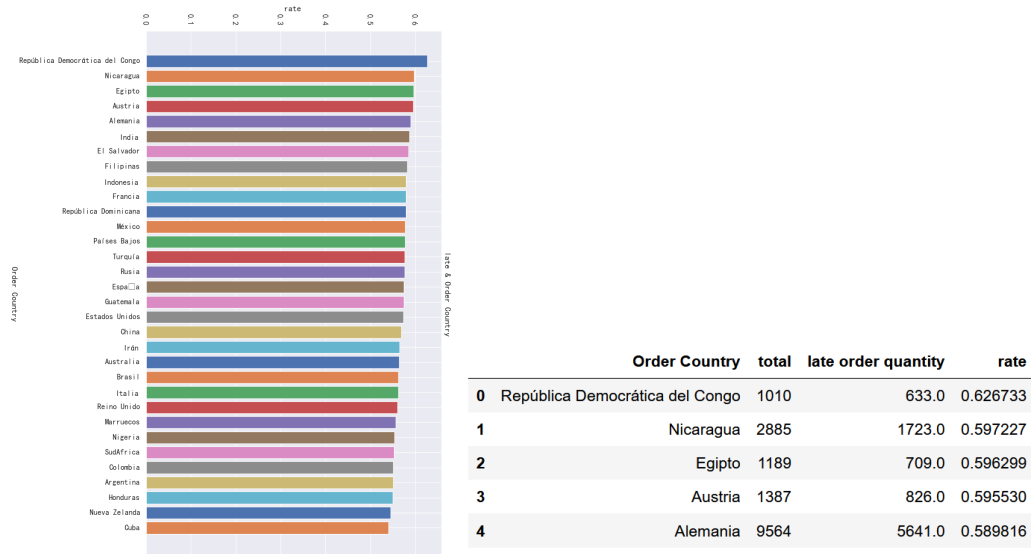
Since a large percentage of orders are late, As a consumer, we would like to learn more about



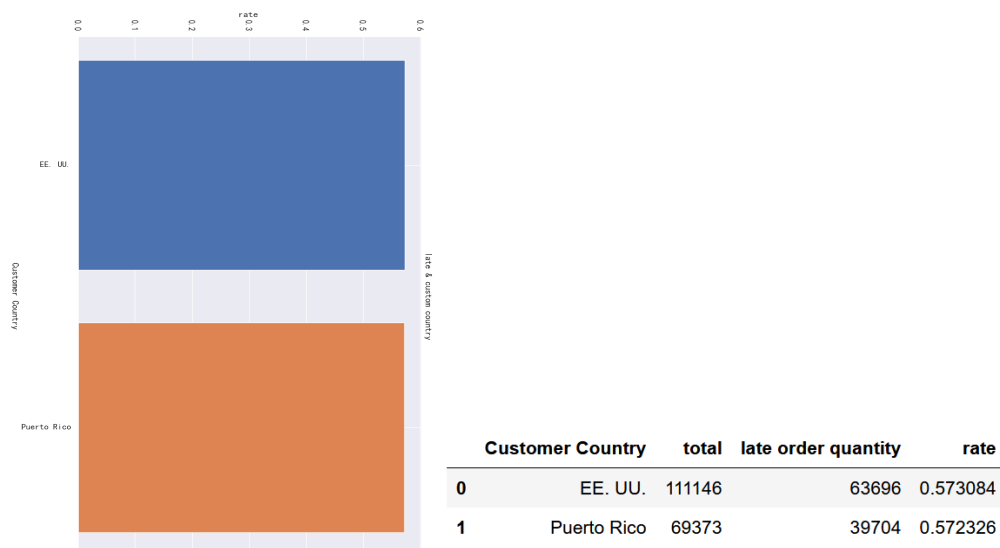
## CS 6220 · DataCo Supply Chain Company Data Analysis

what type of orders are more likely to be late. So we picked out those orders whose shipping status was late and did some analysis, which involves features 'Order Country', 'Customer Country' and 'Goods Category'.

Order Country is the country where goods ship from. We have selected those countries that have more than 1000 orders with the company, and then did the analysis. Most of the countries had more than 50% possibility to be late in their orders. And Congo has the highest rate to be late, which is 62.67%.



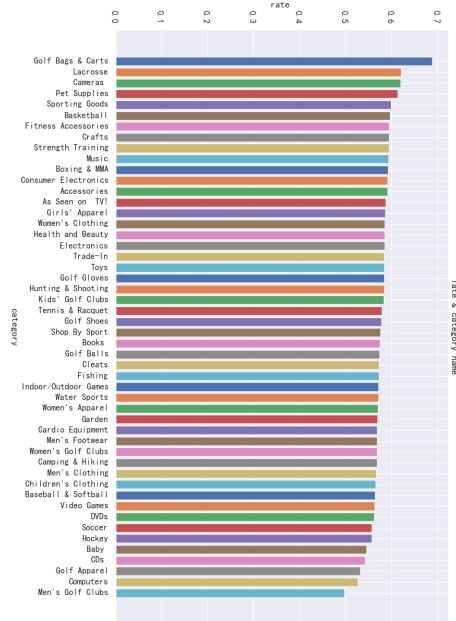
For Customer Country (which is the destination), there are only 2 countries. But both of them have a 57% late order rate, so more than half of the orders are late.



Goods category is the type of goods that the company carried. Most of the goods had more than 50% possibility to be late in their orders. No matter what type of goods it is, And Golf Bags &

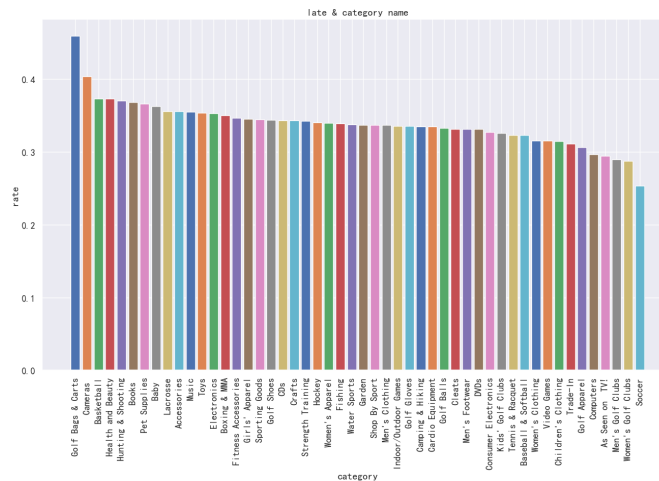
# CS 6220 · DataCo Supply Chain Company Data Analysis

Carts had the highest rate to be late, which is 68.85%.

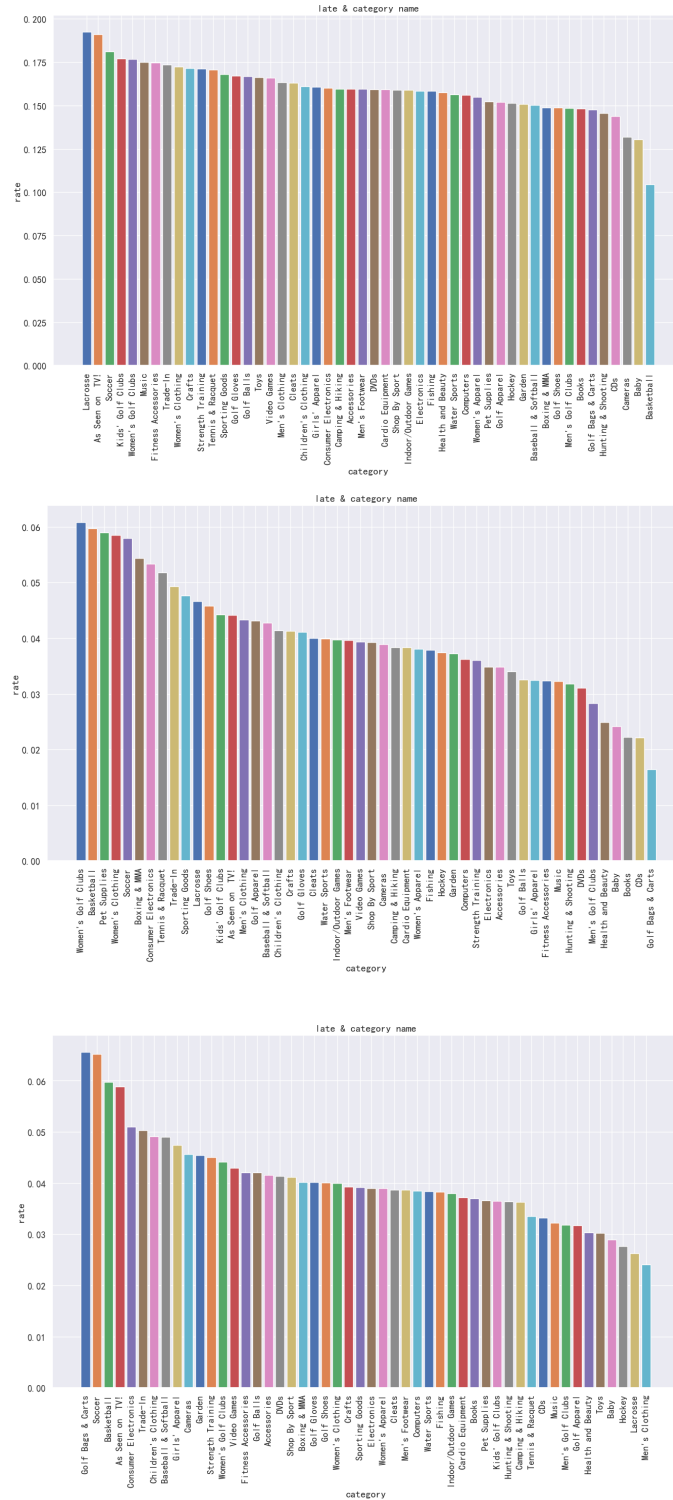


	category name	total	late order quantity	rate
0	Golf Bags & Carts	61	42	0.688525
1	Lacrosse	343	213	0.620991
2	Cameras	592	367	0.619932
3	Pet Supplies	492	302	0.613821
4	Sporting Goods	357	214	0.599440

Since we have learned that late orders were late for 1 - 4 days, we explored the late rate(the percentage of orders being late), in features of category name, customer country and order country, in 1 - 4 late days respectively. After comparing their bar charts of those 4 late days, we found out that, although their rate has changed each year, their sequence of rate does not change after sorting each data segment from high to low late rate. The bar charts below show the 'Category Name' feature's late rate distribution of 1 - 4 late days:

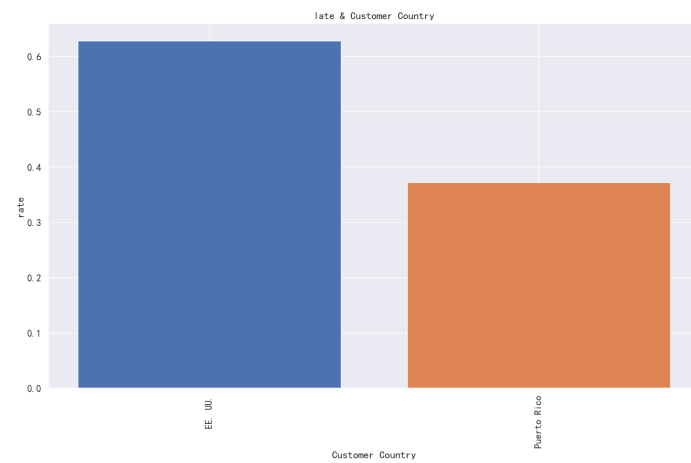
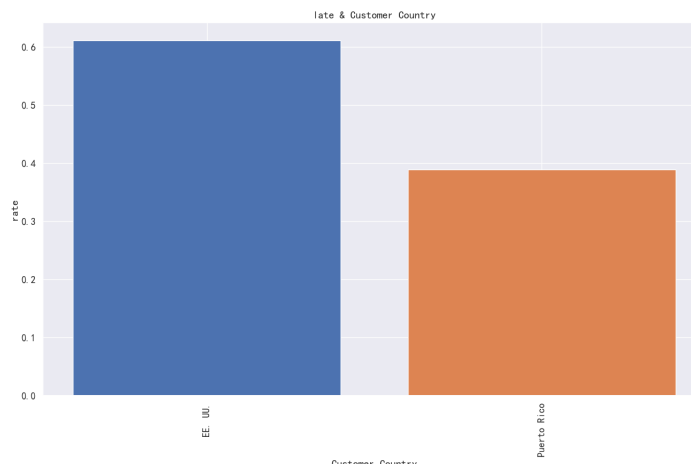
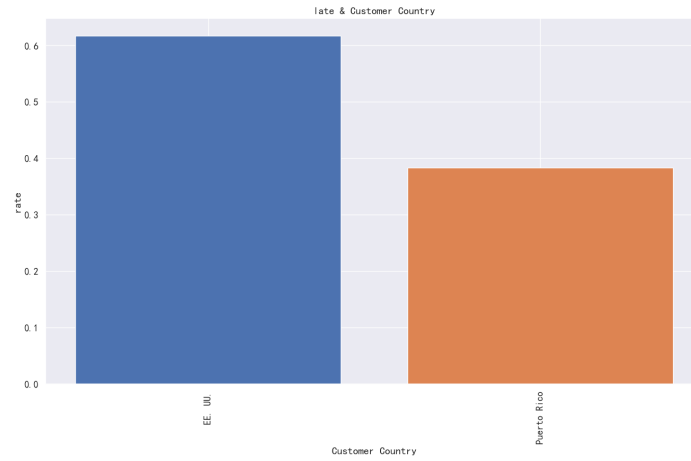


# CS 6220 · DataCo Supply Chain Company Data Analysis

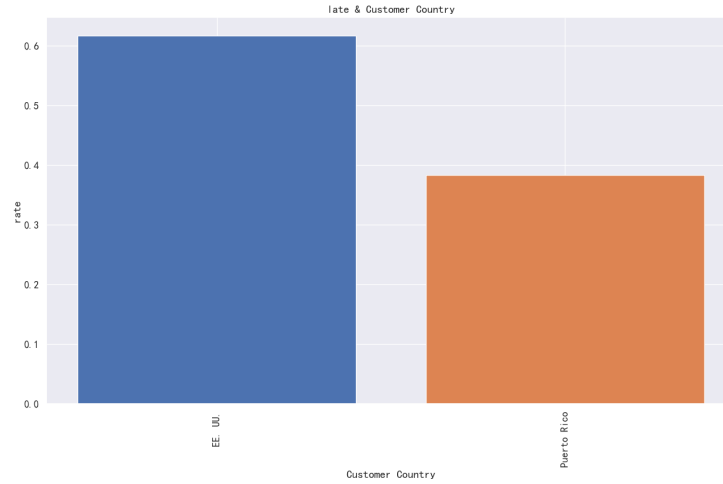


The bar charts below show the 'Customer Country' feature's late rate distribution of 1 - 4 late days (since there is only 2 countries, unlike all the other analysis, we regard the total as all late orders – instead of all orders – so the bar charts below are showing how distribution of order changes in late shipping status from day 1 to day 4):

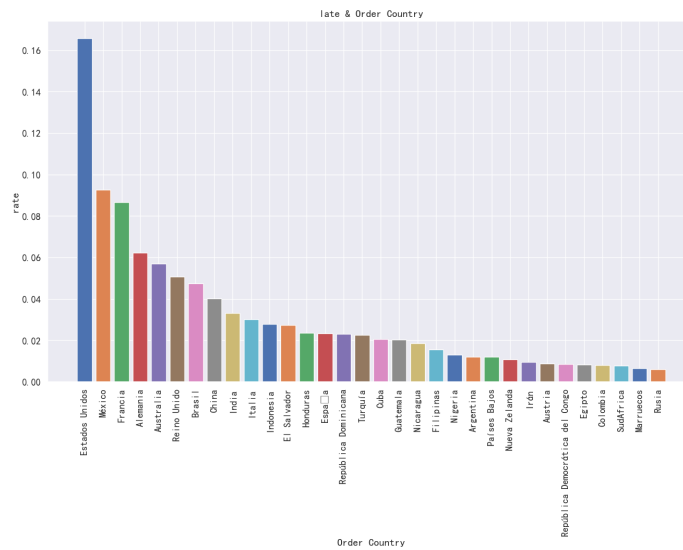
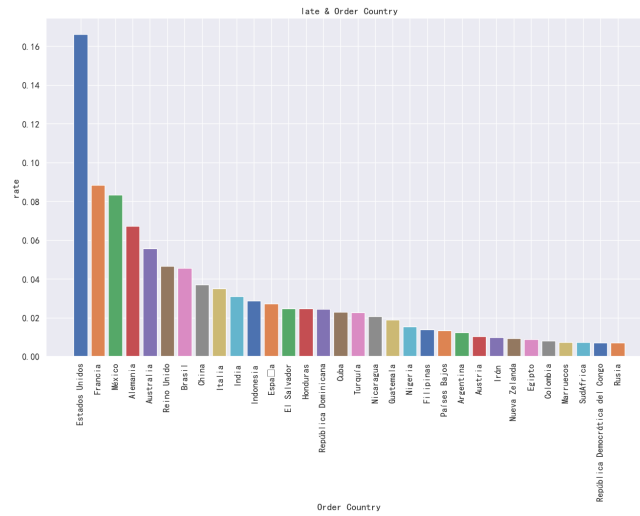
# CS 6220 · DataCo Supply Chain Company Data Analysis



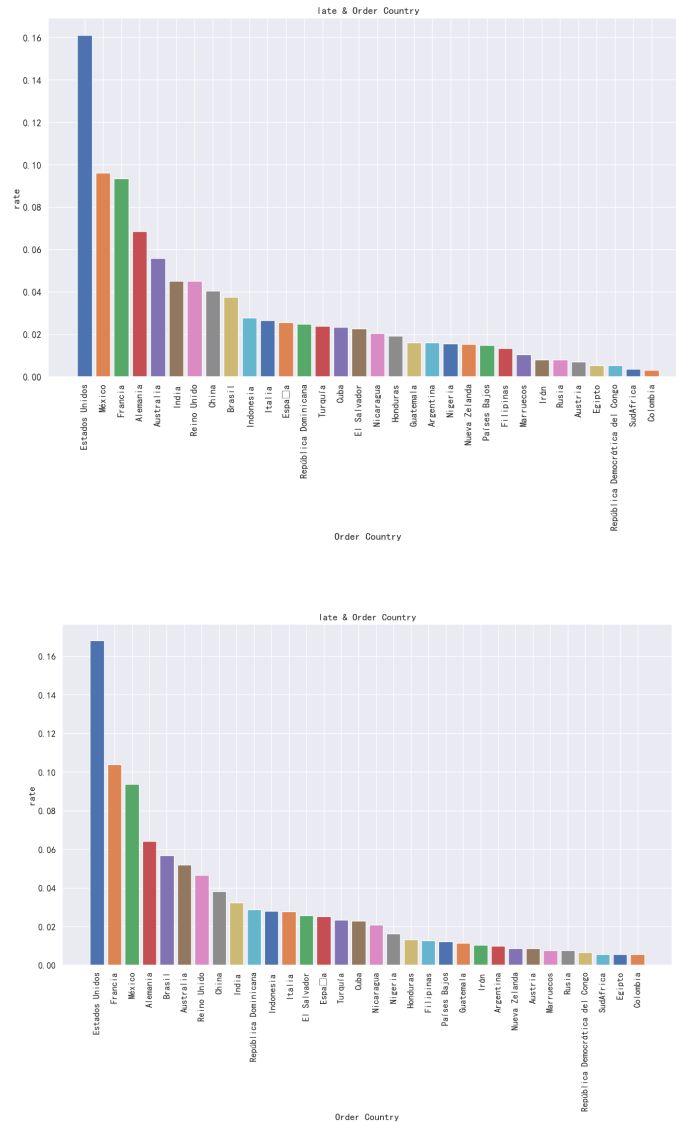
# CS 6220 · DataCo Supply Chain Company Data Analysis



The bar charts below show the 'Category Name' feature's late rate distribution of 1 - 4 late days:



## CS 6220 · DataCo Supply Chain Company Data Analysis



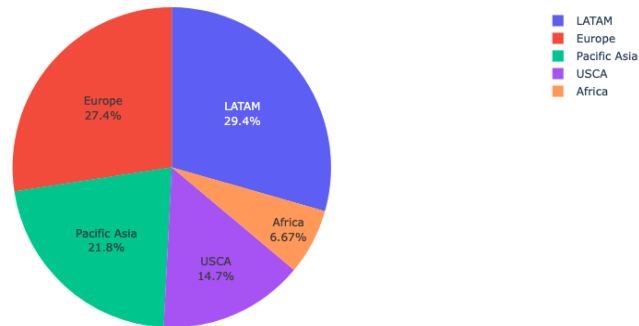
### Dataset and Preprocessing for Regional Information

Location data such as country, state, city names are given to explore departure and destination of the shipment which are useful to further explore the top markets, most popular product category in each market, most popular states in the US that put up a shipment order. More specific geolocational data like longitude and latitude are particularly helpful for us to create a world map. In order to achieve the results, all the relevant location features are selected along with the order quantity and product names. To further narrow down the specific region where the customer is located, we shifted from the grand scope of observing all the countries to focus on the United States. The data analysis will output the market information that's also relevant to the US. While creating a world map to demonstrate the routes of the shipments coming out of the US, two challenges have to be tackled. As the country names were written in Spanish, we had to

## CS 6220 · DataCo Supply Chain Company Data Analysis

preprocess the characters and convert them into English. The second challenge was to find an external dataset that contains the longitude and latitude information of all the countries. We then

Sum of Order Quantity by Market



combined the original dataset with an external location dataset into one dataset that's named as `complete_location_df` so both longitude and latitude data are attached to all the departures and destinations.

### Data Analysis of Regional Information

From the hand-picked location dataset, which stores all departures and destination countries, we could find that there are five global markets where the shipments were delivered to. These markets are ranked based on the number of shipment quantities that's calculated by summing up the order quantities for each market. The below graph on the left shows the ranked order and quantity with unit numbers. We conclude the number one largest market is Latin America and it's followed by Europe, Pacific Asia, US and Canada, and lastly Africa.

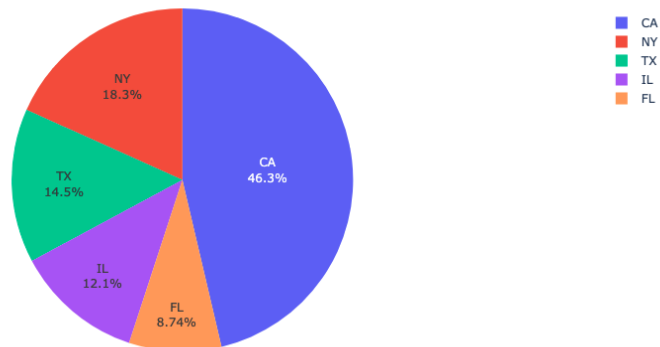


The pie chart below demonstrates the proportion of the order quantity by market in percentages. After that, we continued to clean the dataset to contain the United States as the only customer country and to explore the top states that carry the most shipments. All the numbers are labeled with order quantity represented in percentage. From the pie chart below, we then conclude that

## CS 6220 · DataCo Supply Chain Company Data Analysis

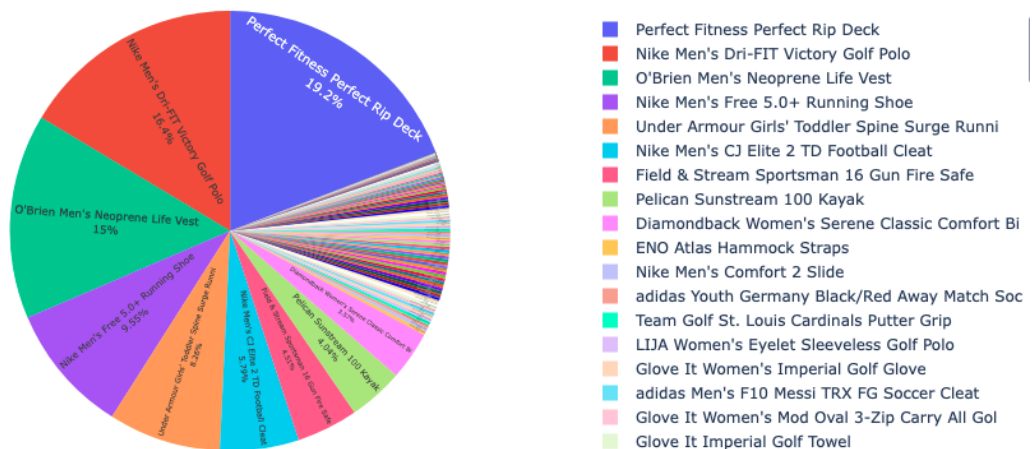
the top five states where most shipment orders were made are California with 46.3% of the total shipment orders made in the US, and it's followed by New York, Texas, Illinois, and Florida. The assumption behind the result is that four out of the five states are located on the borderline of the country where large ports and storage locations are running and Illinois sits in the middle of the country that transfers logistics from both domestic and international orders.

Sum of Order Quantity by US State



We then further explored the most popular products that are ordered from the United States. The reason to do this is to help our client DataCo Supply Chain company to find their potential partners for future business collaborations. This pie chart displays almost all the popular commodities that were ordered from the US and are listed in a descending order. The products shown on the list to the right of the pie chart are mostly sporting goods like sports apparels, running shoes, fitness products. The most popular product is perfect fitness perfect rip deck, followed by Nike Men's Golf Polo, and O'Brien Life Vest, etc.

Percentage of Shipment by Product Name





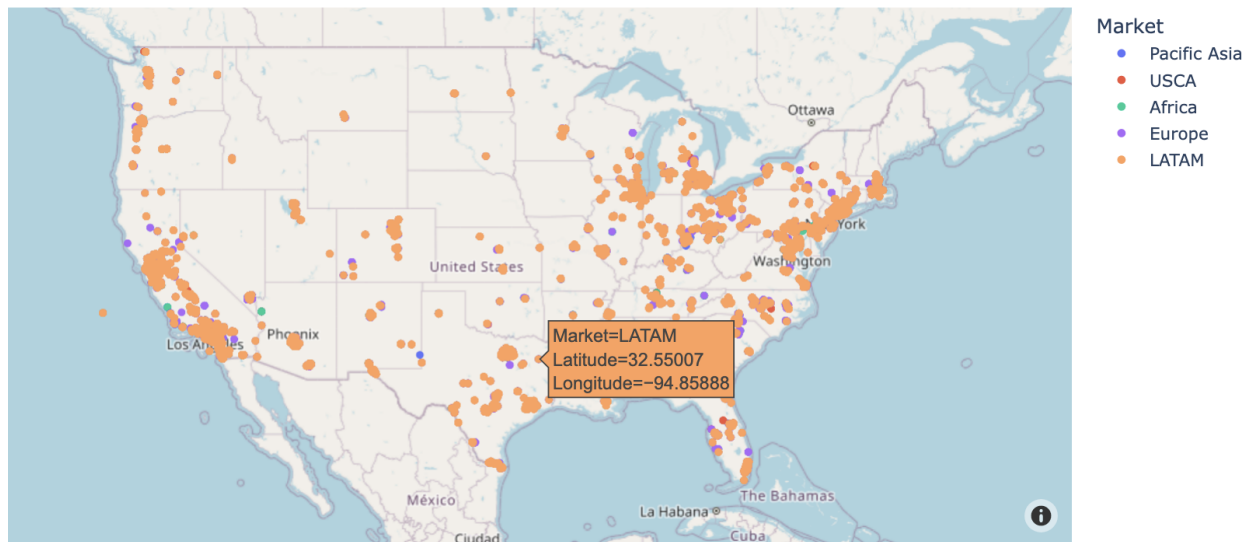
## CS 6220 · DataCo Supply Chain Company Data Analysis

The top 10 countries that ordered the products in the pie chart are shown as follows.

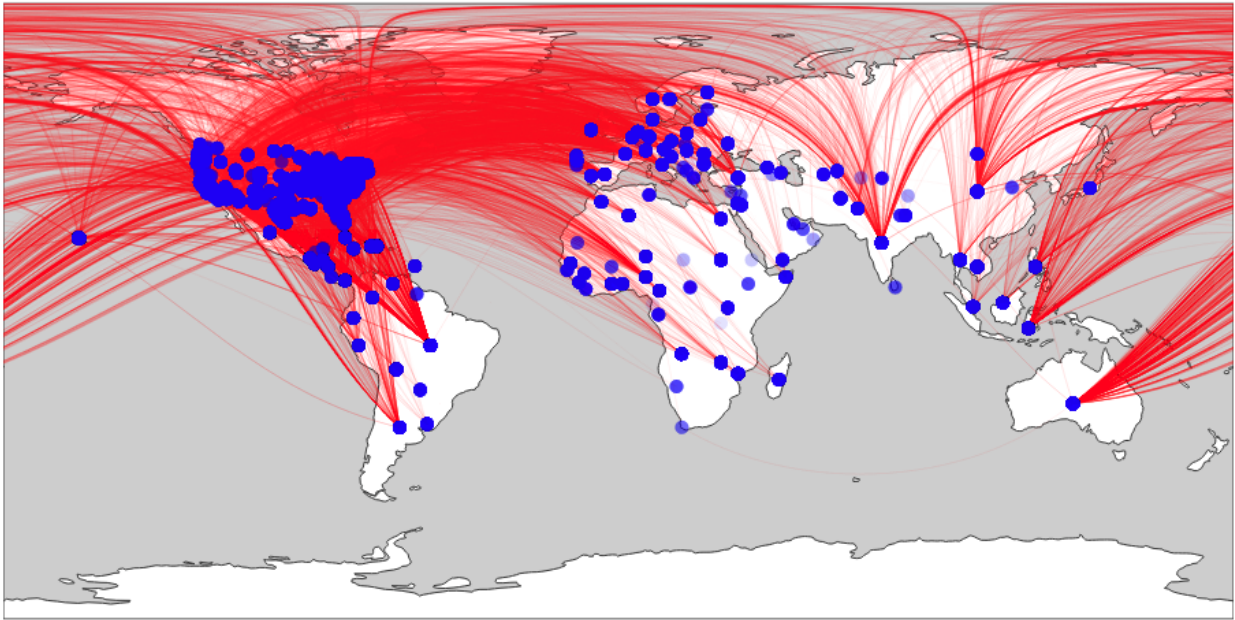
Order Country	
Estados Unidos	10818.0
México	5626.0
Francia	5383.0
Alemania	3778.0
Brasil	3225.0
Australia	2926.0
Reino Unido	2925.0
China	1918.0
Italia	1910.0
India	1621.0

dtype: float64

To visually represent the regional information, we used folium to make a scatter plot on a world map. As shown the orange is the dominant color that represents the largest market, Latin America. Although, the second largest market is Europe that's shown in purple dots, that are covered by orange.

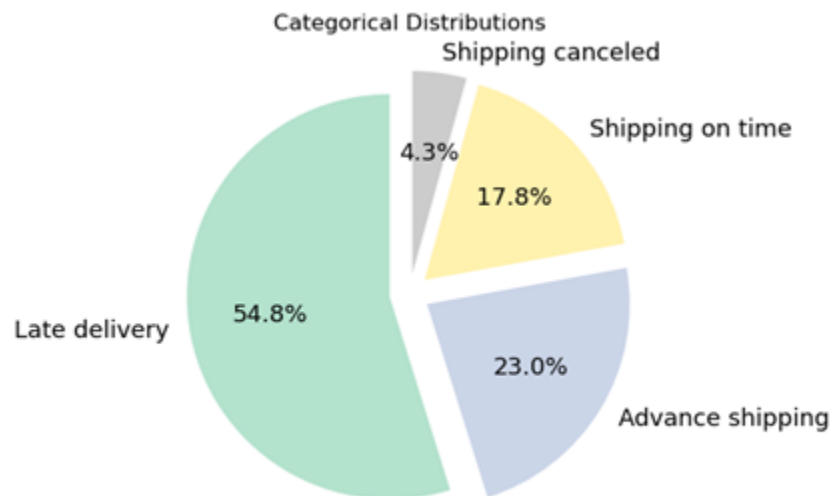


Lastly, the world map is created to show the paths of all the shipments delivered to the destination countries from the United States.



### III. Methods (Explain ML models and evaluations)

#### 1. Techniques to handle imbalanced data



Before handling the imbalanced data, we used StandardScaler to process the data. The dataset used is imbalanced, with the largest class occupying 54.8% and the smallest class only occupying 4.3%. In order to address this issue, we attempted to use SMOTE (Synthetic Minority

Oversampling Technique) and undersampling. Despite the use of SMOTE and undersampling, the accuracy of the minor class did not improve.

```
over = SMOTE(sampling_strategy='minority')
under = RandomUnderSampler(sampling_strategy='majority')
```

### 2. Model selected:

We tried several classifiers including DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, BaggingClassifier and KNeighborsClassifier. The table below shows the metrics of various classifiers using their default parameters.

	Precision	Recall	f1-score
DecisionTree	0.74	0.74	0.74
RandomForest	0.58	0.59	0.59
GradientBoosting	0.54	0.57	0.52
Bagging	0.68	0.62	0.64
KNeighbors	0.53	0.56	0.53

After evaluating the performance and resource cost of various classifiers, we chose DecisionTreeClassifier as the best classifier for this classification problem.

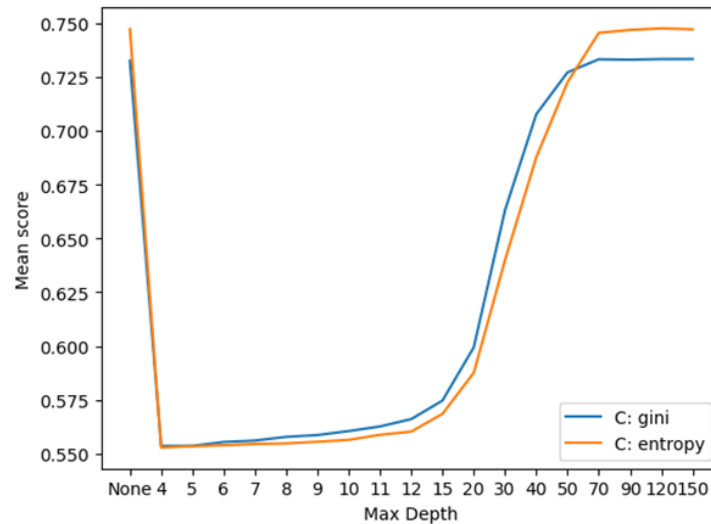
### 3. Hyperparameter tuning

The splitting strategy used was StratifiedKFold cross-validation. Grid search was used to determine the optimal value for the K parameter, and it was found that K=10 provided the best performance. However, other values of K performed only slightly worse.

For the DecisionTreeClassifier, several combinations of criterion and max depth were tried for hyperparameter tuning. However, it was found that the default settings provided sufficient

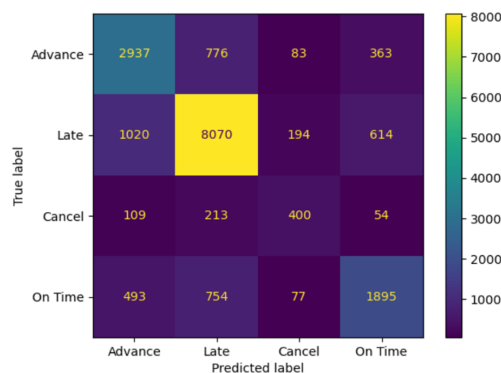
## CS 6220 · DataCo Supply Chain Company Data Analysis

performance and did not need to be changed. Overall, these results indicate that the chosen classifier and splitting strategy are effective for this classification problem.



### 4. The final metrics

	precision	recall	f1-score	support
Advance shipping	0.64	0.71	0.67	4159
Late delivery	0.82	0.82	0.82	9898
Shipping canceled	0.53	0.52	0.53	776
Shipping on time	0.65	0.59	0.62	3219
accuracy			0.74	18052
macro avg	0.66	0.66	0.66	18052
weighted avg	0.74	0.74	0.74	18052

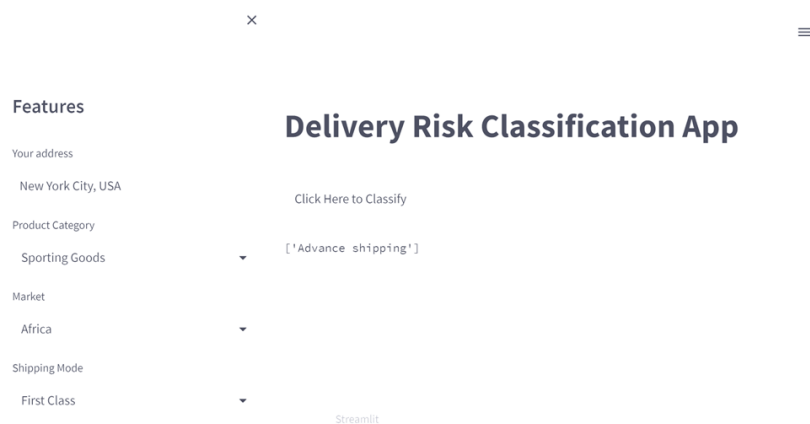


## IV. Graphic User Interface

### 1. Web application

## CS 6220 · DataCo Supply Chain Company Data Analysis

A web application was implemented using streamlit to provide a convenient and user-friendly interface for interacting with the trained model and obtaining delivery predictions. The page includes several parameter options that can be used to help the model make predictions, such as the address, product category, market option, and shipping mode. After filling in these features, users can click the "Click here to Classify" button on the right side of the screen to receive a prediction from the model. The possible predictions are "Shipping on time", "Late delivery", "Advance shipping", and "Shipping canceled". This information can help users determine the delivery risk for a given set of input parameters.



The screenshot shows a web application titled "Delivery Risk Classification App". On the left, there is a sidebar labeled "Features" with a close button (X) at the top. The sidebar contains four input fields: "Your address" with the value "New York City, USA", "Product Category" with a dropdown menu showing "Sporting Goods", "Market" with a dropdown menu showing "Africa", and "Shipping Mode" with a dropdown menu showing "First Class". On the right, there is a "Click Here to Classify" button. Below the button, there is a text input field containing the prediction "[ 'Advance shipping' ]". At the bottom right, there is a "Streamlit" logo.

## 2. Docker

Using a docker image allows the web application to be easily deployed and run on any platform that supports docker. This provides greater flexibility and accessibility for users who want to use the application. Our docker image was created using miniconda3 as the base image from the docker hub. The Dockerfile for this image is shown below.

```
FROM continuumio/miniconda3

# Install python packages
RUN mkdir /opt/api
COPY requirements.txt /opt/api/
RUN pip install -r /opt/api/requirements.txt

# Copy files into container
COPY model /opt/api/model
COPY streamlit_demo.py /opt/api/
COPY data /opt/api/data

# Set work directory and open the required port
WORKDIR /opt/api
EXPOSE 8501

# Run our service script
CMD ["streamlit", "run", "streamlit_demo.py"]
```

### V. Conclusion and future enhancement

The decision tree classifier was chosen as the model for the problem. The model's performance was found to be good for most classes, but not for the minor class of shipping cancellations. In order to improve the model's precision for this class, it may be necessary to explore additional delivery datasets that include information on shipping cancellations. Additionally, incorporating time labels into the dataset may allow for further exploration of events that influence the delivery status, providing valuable insights for clients. Overall, it is important to continue refining the model in order to improve its performance for all classes.

### Reference

1. [SMOTE for Imbalanced Classification with Python - MachineLearningMastery.com](#)
2. [Hyperparameter tuning - GeeksforGeeks](#)
3. [A Docker Tutorial for Beginners \(docker-curriculum.com\)](#)
4. [A Gentle Introduction to k-fold Cross-Validation - MachineLearningMastery.com](#)
5. [DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS](#)