Supply Chain Optimization

-Jason Liu Doyle
2024200

Supply Chain Optimization

Structure

- Business description.
- Technologies used.
- What has been accomplished so far?
- Challenges encountered.
- Results and analysis… next steps.
- Conclusion.

Business Description

1. Implement machine learning models on data given to us by the business to predict whether orders will arrive late or on time.

2. Identify and analyze distinct customer segments to better understand their characteristics.

Hypothesis + Methodology

Objective 1,

- Multiple classification models,
- Narrow to the top three,
- Find the best one.

Objective 2,

- Elbow method
- silhouette scores
- k-means clustering
- EDA

Methodology

- Agile project management methodology.

Success Criteria

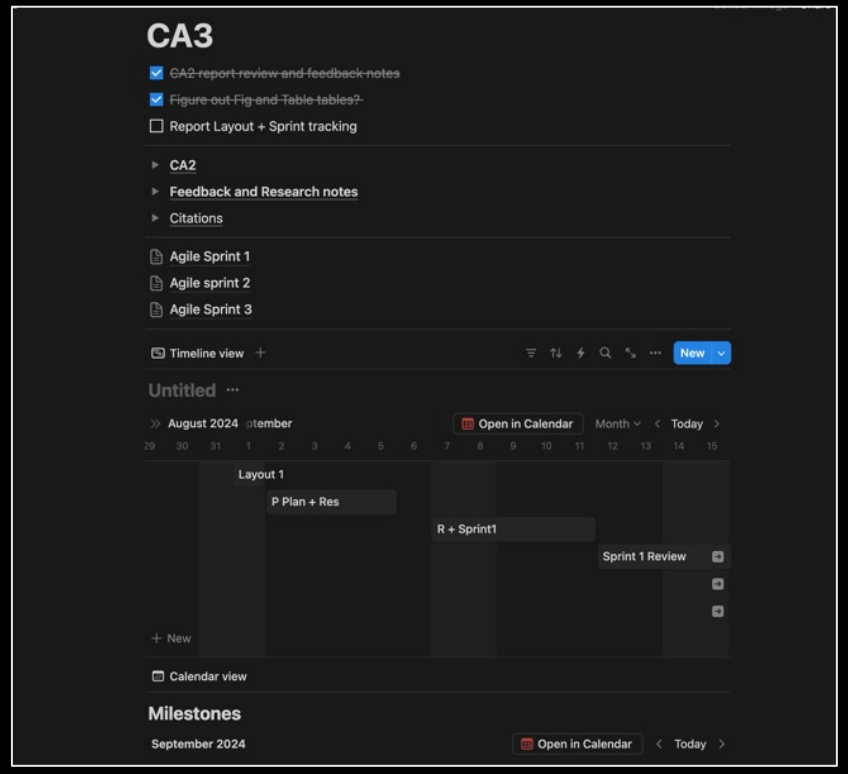
Objective 1

- Best in CA2 + CA3
- Recall (70+).
- High weights (0.1+) and gain (50+).

Objective 2

- well-defined customer segments
- Moderate to high silhouette score.

Agile Methodology



Week 1-2: Review of CA2 + Feedback, Brainstorm ideas, Project scope & Research
Week 3-4: Develop new models and try new EDA visualisations
Week 5-6: Test out model efficiency and compare with previous CA
Week 7-8: Finalise models, and prepare report and presentation

Technologies Used

Libraries

Data Manipulation and Analysis

- Pandas
- Numpy
- Datetime

Model Evaluation and Metrics

- sklearn.metrics
- ConfusionMatrixDisplay
- sklearn.model_selection
- xgboost
- lime.lime_tabular
- shap

Data Visualization

- seaborn (sns)
- matplotlib.pyplot (plt)
- matplotlib.cm & get_cmap
- plotly.graph_objs & plotly.io
- yellowbrick

Statistical Modeling and Tests

- scipy & scipy.stats
- statsmodels.api
- statsmodels.formula.api (ols)

Preprocessing and Pipelines

- sklearn.preprocessing (StandardScaler, LabelEncoder)
- sklearn.pipeline (make_pipeline)

Machine Learning

- sklearn.cluster
- sklearn.decomposition (PCA)
- sklearn.ensemble
- sklearn.naive_bayes (GaussianNB)
- sklearn.svm (SVC, LinearSVC, NuSVC)
- sklearn.tree (DecisionTreeClassifier)
- sklearn.linear_model (LogisticRegression)
- sklearn.neighbors (KNeighborsClassifier)
- sklearn.neural_network (MLPClassifier)
- tensorflow.keras.models & layers
- scikeras.wrappers (KerasClassifier)

Utilities

- warnings
- sys
- matplotlib.lines (Line2D)
- matplotlib.colors & ListedColormap
- mpl_toolkits.mplot3d (Axes3D)

Models Used

NeighborsClassifier(2)
SVC(probability=True)
DecisionTreeClassifier
RandomForestClassifier
AdaBoostClassifier
GradientBoostingClassifier
GaussianNB
QuadraticDiscriminantAnalysis
MLPClassifier

Final Three Models

Random Forest

n_estimators	100, 300
max_depth	10, 20, 30
min_samples_split	2, 10
min_samples_leaf	1, 5
criterion	Gini, Entropy

n_estimators	300
max_depth	30
min_samples_split	2
min_samples_leaf	1
criterion	Gini

n_estimators	200
max_depth	15
min_samples_split	10
min_samples_leaf	4
criterion	Gini

Metric	Value
Cross-Validation Scores	[0.6974 0.7063 0.7009 0.7056 0.7003]
Mean Cross-Validation Accuracy	0.7020962537043967
Training Accuracy	0.7355708862068456
Validation Accuracy	0.7015732328827831

KNN

N_neighbours	5, 10, 20
Weights	uniform, distance
Algorithm	Auto, ball tree
Leaf size	30, 40
p	1, 2

N_neighbours	5
Weights	distance
Algorithm	Auto
Leaf size	30
p	1

N_neighbours	8
Weights	uniform
Algorithm	Auto
Leaf size	30
p	2

Metric	Value
Cross-Validation Scores	[0.5909 0.5905 0.5949 0.588 0.586]
Mean Cross-Validation Accuracy	0.5900609207276766
Training Accuracy	0.7162188113985198
Validation Accuracy	0.6031464657655662

XGBoost

n_estimators	100, 300, 500
Max Depth	3, 5, 7, 9
Learning Rate	0.01, 0.05, 0.1
Sub Sample	0.7, 0.8, 1.0
colsample_bytree	0.7, 0.8, 1.0

n_estimators	500
Max Depth	9
Learning Rate	0.1
Sub Sample	1.0
colsample_bytree	1.0

n_estimators	300
Max Depth	8
Learning Rate	0.1
Sub Sample	1.0
colsample_bytree	1.0

Metric	Value
Cross-Validation Scores	[0.8683 0.8779 0.8832 0.8753 0.8744]
Mean Cross-Validation Accuracy	0.8758216964542205
Training Accuracy	0.9624333387499446
Validation Accuracy	0.8193219587857301

Model Interpretation

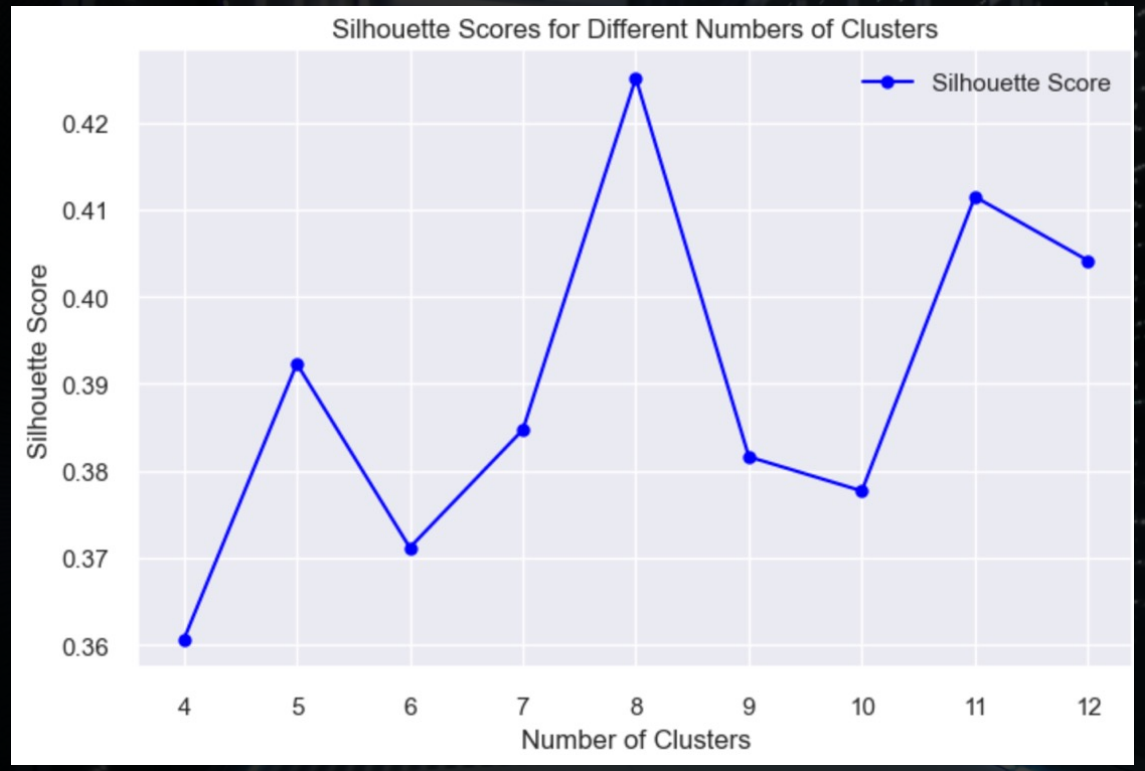
- LIME (Used on a random feature)
- XGBoosts Built in Feature Importance for Weight and Gain

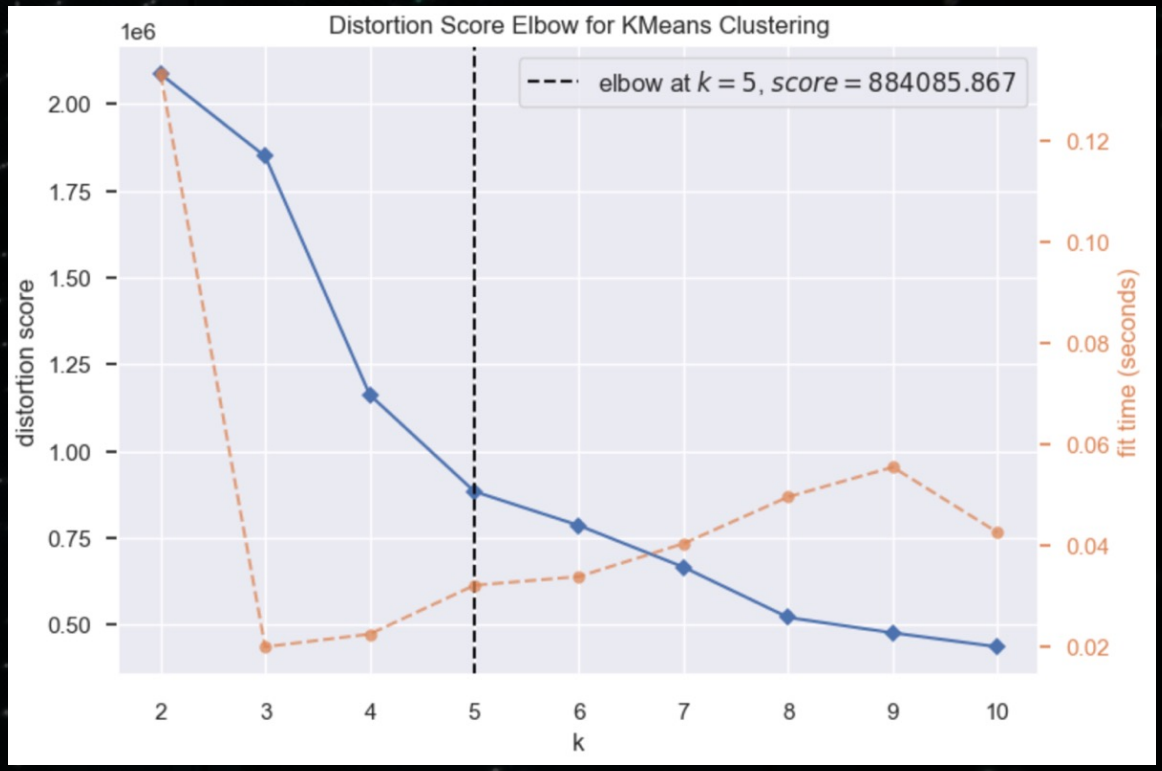
Objective 2

Dimensionality Reduction

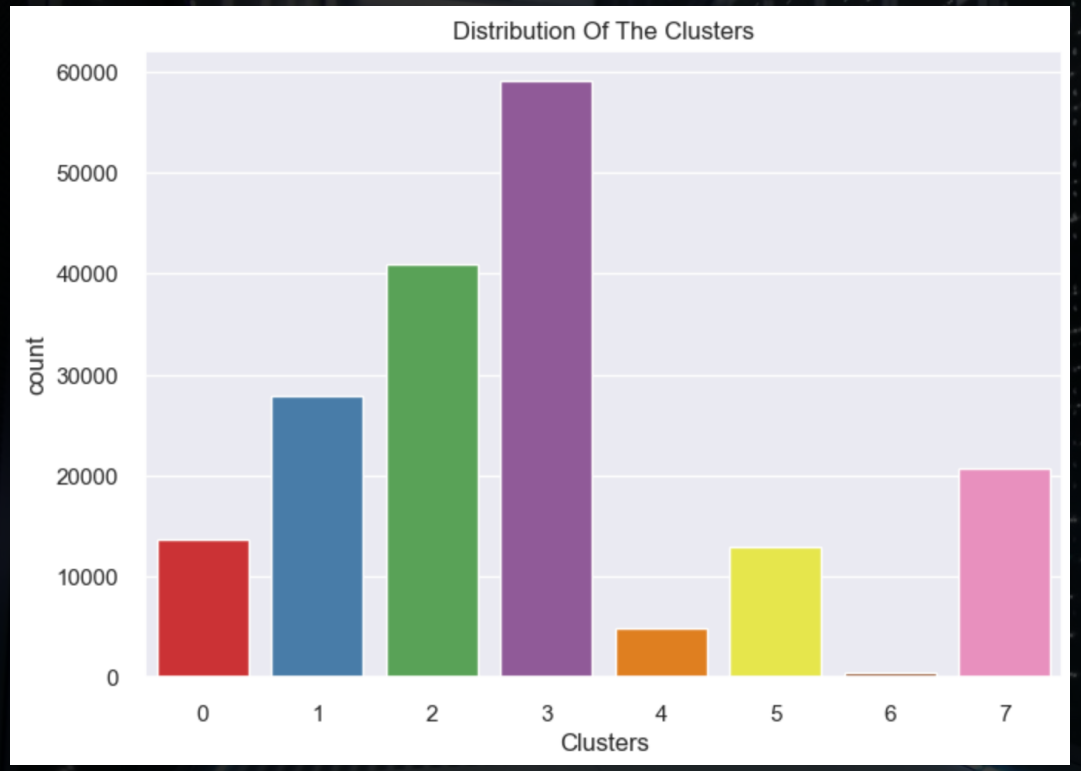
Curse of dimensionality and PCA

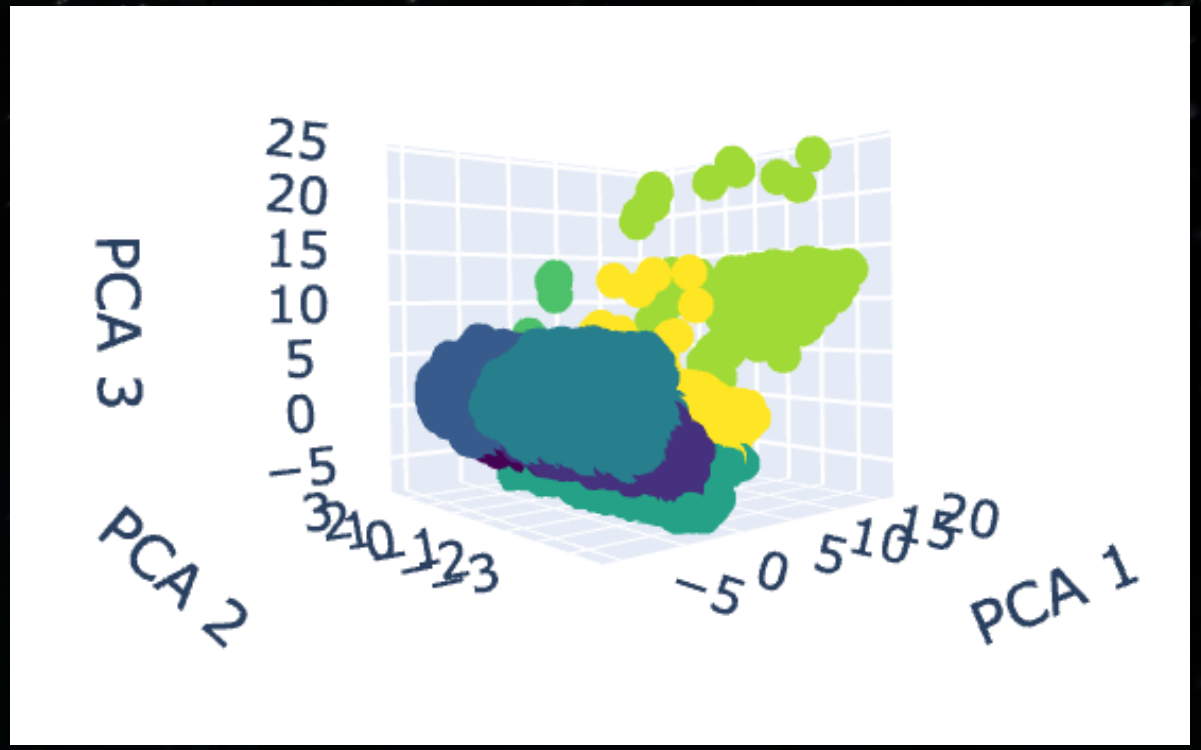
Elbow and Silhouette





K means


Distribution Of The Clusters
count
60000
50000
40000
30000
20000
10000
0
0
1
2
3
4
5
6
7
Clusters


PCA 3
PCA 2
PCA 1
25
20
15
10
5
0
−5

What has been Accomplished so far ?

Dataset

Dataset of Supply Chains used by the company DataCo Global (Constante et al.)

180,519 observations for 44 features

The memory usage is only 60.6+mb which should not be a problem for my laptop

one dataset showing no need to concatenate extra datasets

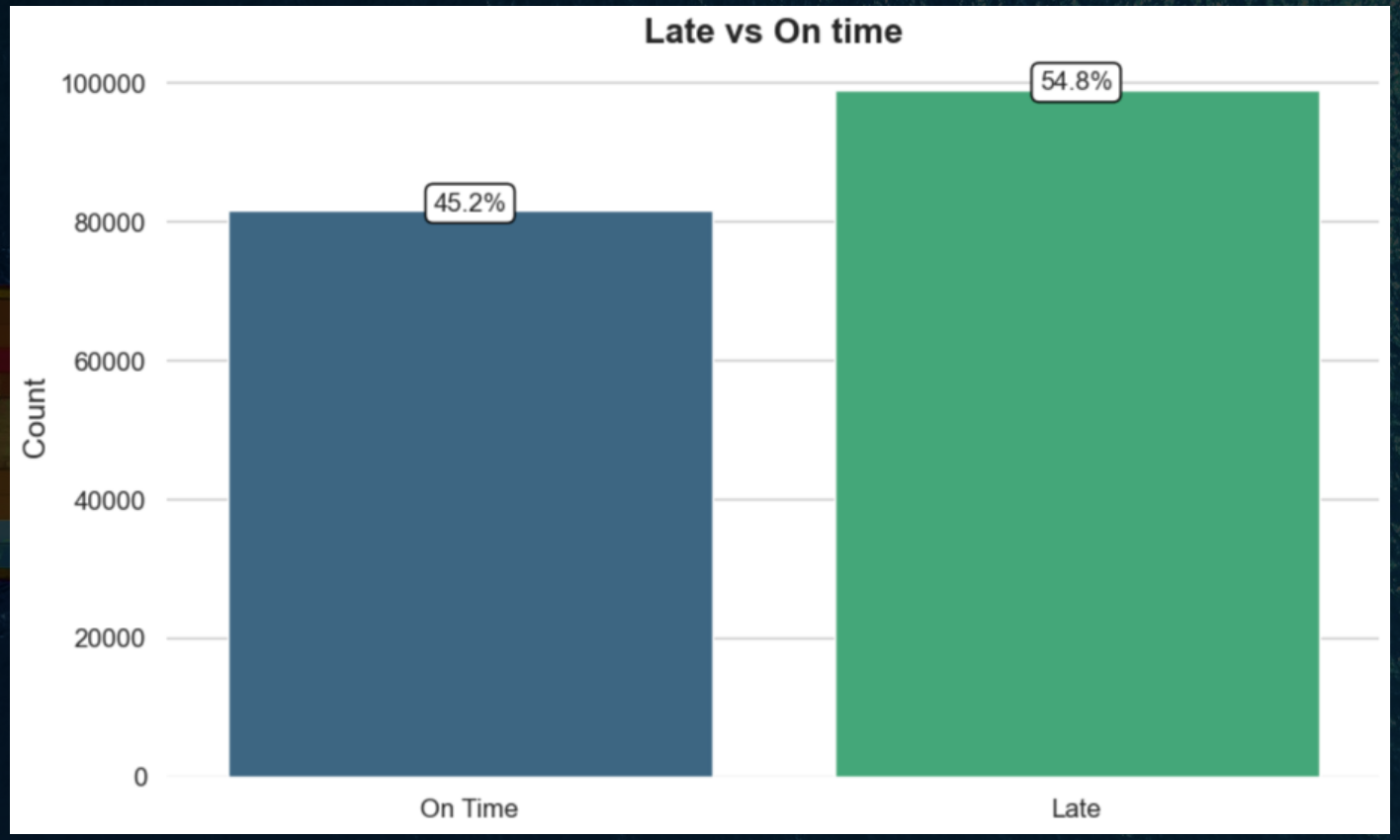
Our target variable for Objective1 will be ‘late_delivery_risk’, which as mentioned before is a binary column; 0 = on time, 1 = is late

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180519 entries, 0 to 180518
Data columns (total 44 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   type                          180519 non-null  object
 1   days_for_shipping_real        180519 non-null  int64
 2   days_for_shipment_scheduled   180519 non-null  int64
 3   benefit_per_order             180519 non-null  float64
 4   sales_per_customer            180519 non-null  float64
 5   delivery_status               180519 non-null  object
 6   late_delivery_risk            180519 non-null  int64
 7   category_id                   180519 non-null  int64
 8   category_name                 180519 non-null  object
 9   customer_city                 180519 non-null  object
 10  customer_country              180519 non-null  object
 11  customer_id                   180519 non-null  int64
 12  customer_segment              180519 non-null  object
 13  customer_state                180519 non-null  object
 14  customer_zipcode              180516 non-null  float64
 15  department_id                 180519 non-null  int64
 16  department_name               180519 non-null  object
 17  latitude                      180519 non-null  float64
 18  longitude                     180519 non-null  float64
 19  market                        180519 non-null  object
 20  order_city                    180519 non-null  object
 21  order_country                 180519 non-null  object
 22  order_customer_id             180519 non-null  int64
 23  order_date                    180519 non-null  object
 24  order_id                      180519 non-null  int64
 25  order_item_cardprod_id        180519 non-null  int64
 26  order_item_discount           180519 non-null  float64
 27  order_item_discount_rate      180519 non-null  float64
 28  order_item_id                 180519 non-null  int64
 29  order_item_product_price      180519 non-null  float64
 30  order_item_profit_ratio       180519 non-null  float64
 31  order_item_quantity           180519 non-null  int64
 32  sales                         180519 non-null  float64
 33  order_item_total              180519 non-null  float64
 34  order_profit_per_order        180519 non-null  float64
 35  order_region                  180519 non-null  object
 36  order_state                   180519 non-null  object
 37  order_status                  180519 non-null  object
 38  product_card_id               180519 non-null  int64
 39  product_category_id           180519 non-null  int64
 40  product_name                  180519 non-null  object
 41  product_price                 180519 non-null  float64
 42  shipping_date                 180519 non-null  object
 43  shipping_mode                 180519 non-null  object
dtypes: float64(13), int64(13), object(18)
memory usage: 60.6+ MB
```

Target

‘late_delivery_risk’

0 = on time, 1 = is late



Cleaning

- Removed useless features
- 'customer_email’,
- 'customer_fname’,
- 'customer_lname’,
- 'customer_password’,
- 'product_image’,
- 'product_status’,
- 'order_zipcode’,
- 'product_description’,
- 'customer_street'

1.1 Renaming the Data + Dropping Columns

Renaming everything with the snake convention

[4]:
```
df.columns = [
    'type',
    '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_email',
    'customer_fname',
    'customer_id',
    'customer_lname',
    'customer_password',
    'customer_segment',
    'customer_state',
    'customer_street',
    'customer_zipcode',
    'department_id',
    'department_name',
    'latitude',
```

Data preparation

- Null sets
- Dulplicates
- Split into
  - numerical
  - categorical


EDA

Pandas Profiling

Overview

Overview | Alerts 5 | Reproduction

Dataset statistics

Number of variables	44
Number of observations	180516
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	62.0 MiB
Average record size in memory	360.0 B

Variable types

Categorical	14
Numeric	23
Text	5
DateTime	2

Variables

Statistical Test Numerical

	count	mean	std	min	25%	50%	75%	max
days_for_shipping_real	180519.0	3.497654	1.623722	0.00000	2.000000	3.000000	5.000000	6.000000
days_for_shipment_scheduled	180519.0	2.931847	1.374449	0.00000	2.000000	4.000000	4.000000	4.000000
benefit_per_order	180519.0	21.974989	104.433526	-4274.97998	7.000000	31.520000	64.800003	911.799988
sales_per_customer	180519.0	183.107609	120.043670	7.49000	104.379997	163.990005	247.399994	1939.989990
late_delivery_risk	180519.0	0.548291	0.497664	0.00000	0.000000	1.000000	1.000000	1.000000
customer_id	180519.0	6691.379495	4162.918106	1.00000	3258.500000	6457.000000	9779.000000	20757.000000
order_item_discount	180519.0	20.664741	21.800901	0.00000	5.400000	14.000000	29.990000	500.000000
order_item_discount_rate	180519.0	0.101668	0.070415	0.00000	0.040000	0.100000	0.160000	0.250000
order_item_product_price	180519.0	141.232550	139.732492	9.99000	50.000000	59.990002	199.990005	1999.989990
order_item_profit_ratio	180519.0	0.120647	0.466796	-2.75000	0.080000	0.270000	0.360000	0.500000
order_item_quantity	180519.0	2.127638	1.453451	1.00000	1.000000	1.000000	3.000000	5.000000
sales	180519.0	203.772096	132.273077	9.99000	119.980003	199.919998	299.950012	1999.989990
order_item_total	180519.0	183.107609	120.043670	7.49000	104.379997	163.990005	247.399994	1939.989990
order_profit_per_order	180519.0	21.974989	104.433526	-4274.97998	7.000000	31.520000	64.800003	911.799988
product_price	180519.0	141.232550	139.732492	9.99000	50.000000	59.990002	199.990005	1999.989990

Graphs

- Distribution of Order Item Profit Ratio by Delivery Status
- Distribution of Order Item Quantity by Delivery Status
- Distribution of Sales by Delivery Status
- Distribution of Order Item Total by Delivery Status
- Distribution of Profit per order by Delivery Status
- Distribution of Order Item Product Price by Delivery Status
- Distribution of Days for shipping by Delivery Status
- Distribution of days for shipment scheduled by Delivery Status
- Distribution of Binned Benefit per Order by Delivery Status
- Order Item Discount by Delivery Status
- Distribution of Order Item Discount Rate by Delivery Status

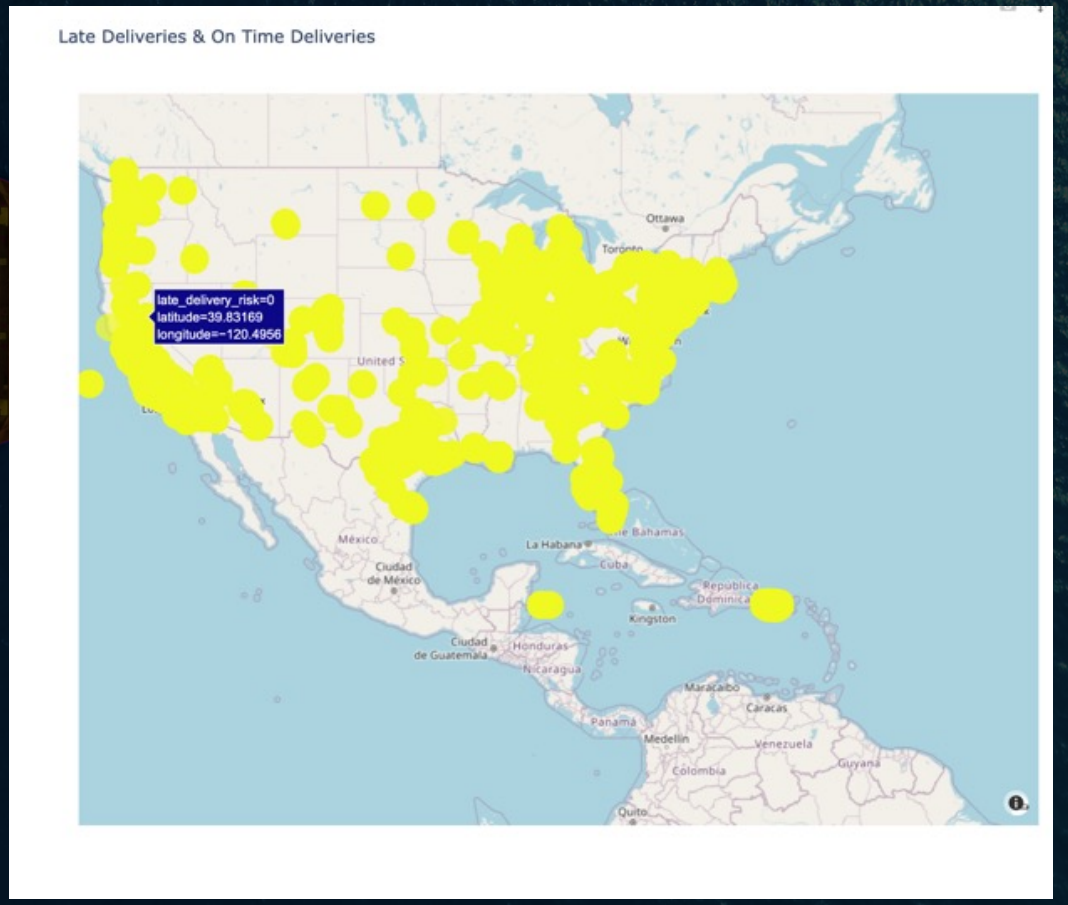
Statistical Test Categorical

	count	unique	top	freq
type	180519	4	DEBIT	69295
delivery_status	180519	4	Late delivery	98977
category_name	180519	50	Cleats	24551
customer_city	180519	563	Caguas	66770
customer_country	180519	2	EE. UU.	111146
customer_segment	180519	3	Consumer	93504
customer_state	180519	46	PR	69373
department_name	180519	11	Fan Shop	66861
market	180519	5	LATAM	51594
order_city	180519	3597	Santo Domingo	2211
order_country	180519	164	Estados Unidos	24840
order_date	180519	65752	12/14/2016 12:29	5
order_region	180519	23	Central America	28341
order_state	180519	1089	Inglaterra	6722
order_status	180519	9	COMPLETE	59491
product_name	180519	118	Perfect Fitness Perfect Rip Deck	24515
shipping_date	180519	63701	1/5/2016 5:58	10
shipping_mode	180519	4	Standard Class	107752

Graphs

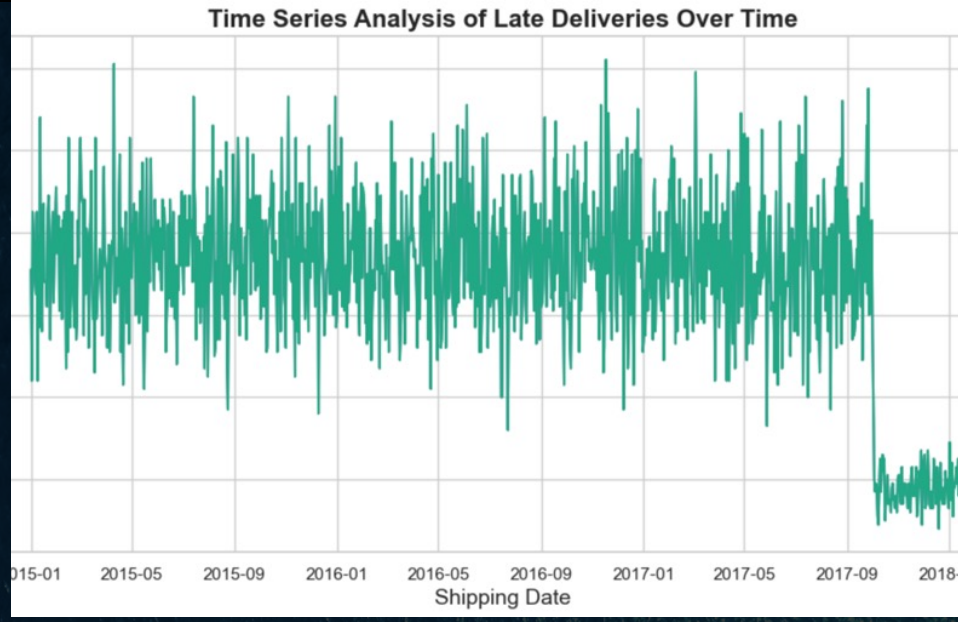
- Distribution of Type by Delivery Status
- Distribution of Mode by Delivery Status
- Distribution of Customer Segment by Delivery Status
- Distribution of Order Status by Delivery Status
- Distribution of Top products by Delivery Status
- Distribution of Department by Delivery Status
- Distribution of Customer Country by Delivery Status
- Distribution of Customer City by Delivery Status
- Distribution of Customer State by Delivery Status
- Distribution of Market by Delivery Status
- Distribution of Top Order Regions by Delivery Status
- Distribution of Top Order Cities by Delivery Status
- Distribution of Top Order Countries by Delivery Status
- Distribution of Top Order States by Delivery Status

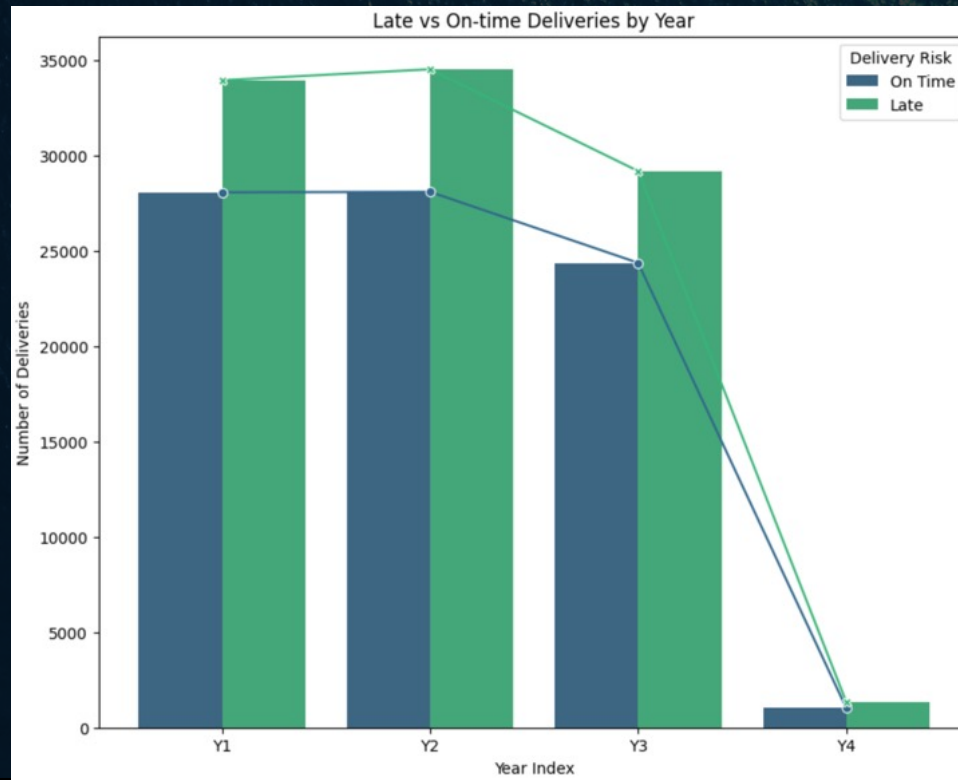
Interactive maps



Graphs

- Time Series Plot of Late Deliveries
- Time Series Plot of 2017
- Time Series Plot of 2016
- Time series plot of 2015
- Late vs On time Delivery by Year
- Late vs On time Delivery by Quarter
- Late vs On time Deliveries by Hour



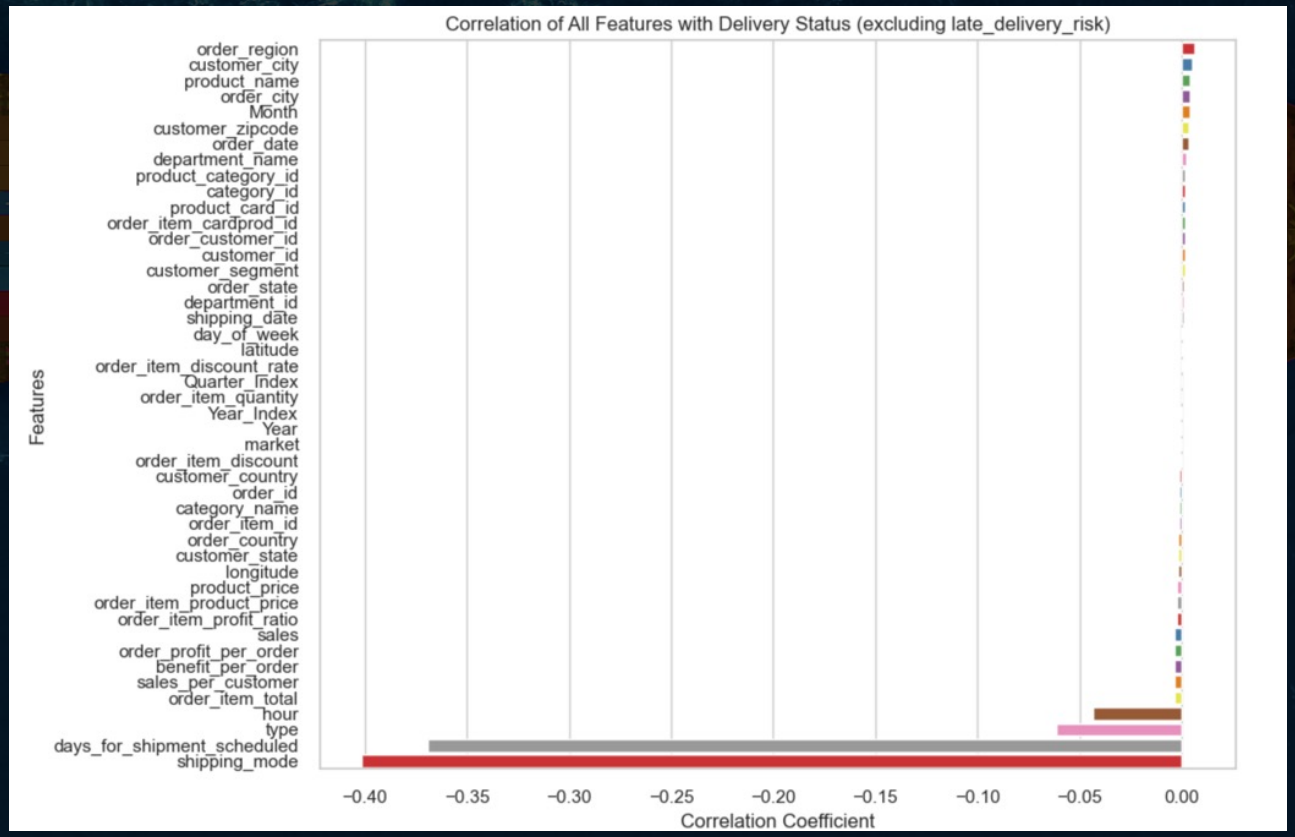


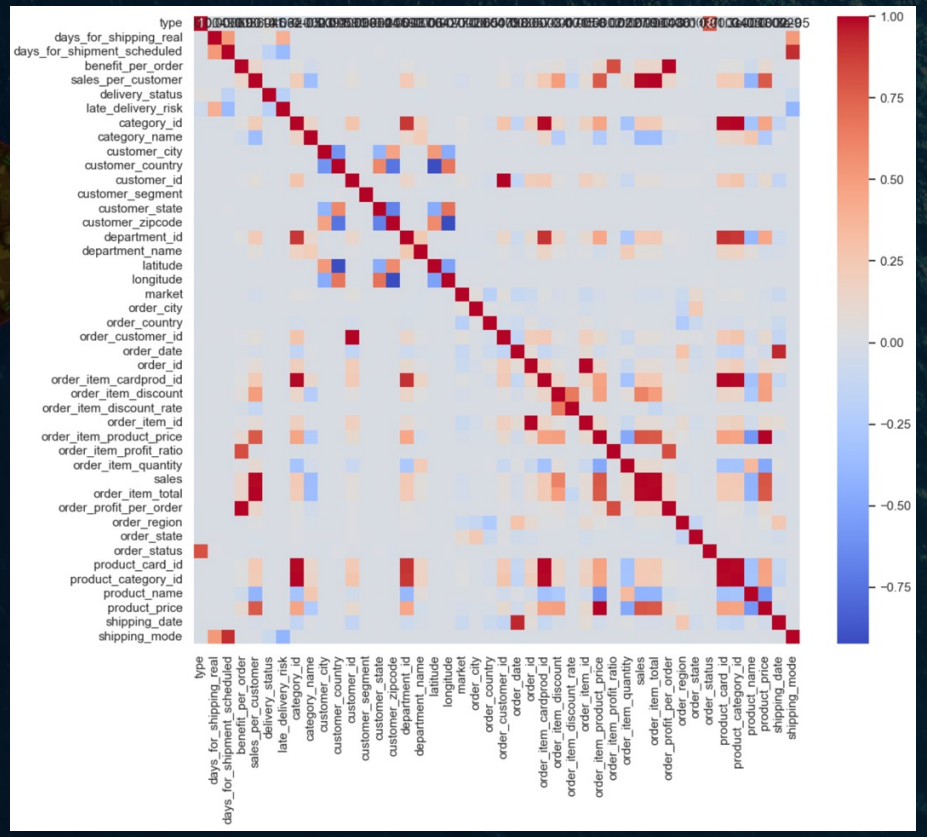
Correlation analysis

Label encoded my data, dropped 'delivery_status','days_for_shipping_real','order_status’ to prevent data leakage

late_delivery_risk	1.000000
days_for_shipping_real	0.401415
order_region	0.006159
customer_city	0.005082
shipping_date	0.004439
product_name	0.003992
order_city	0.003838
order_date	0.003152
customer_zipcode	0.003148
department_name	0.002356
product_category_id	0.001752
category_id	0.001752
order_item_cardprod_id	0.001490
product_card_id	0.001490
customer_id	0.001484
order_customer_id	0.001484
customer_segment	0.001419
order_state	0.001223
department_id	0.001077
latitude	0.000679
order_item_discount_rate	0.000404
order_item_quantity	-0.000139
market	-0.000578
order_item_discount	-0.000750
customer_country	-0.001044
order_id	-0.001293
category_name	-0.001361
order_item_id	-0.001376
order_country	-0.001649
customer_state	-0.001839
longitude	-0.001915
product_price	-0.002175
order_item_product_price	-0.002175
order_item_profit_ratio	-0.002316
sales	-0.003564
benefit_per_order	-0.003727
order_profit_per_order	-0.003727
order_item_total	-0.003791
sales_per_customer	-0.003791
order_status	-0.004130
type	-0.061529
delivery_status	-0.190507
days_for_shipment_scheduled	-0.369352
shipping_mode	-0.401375

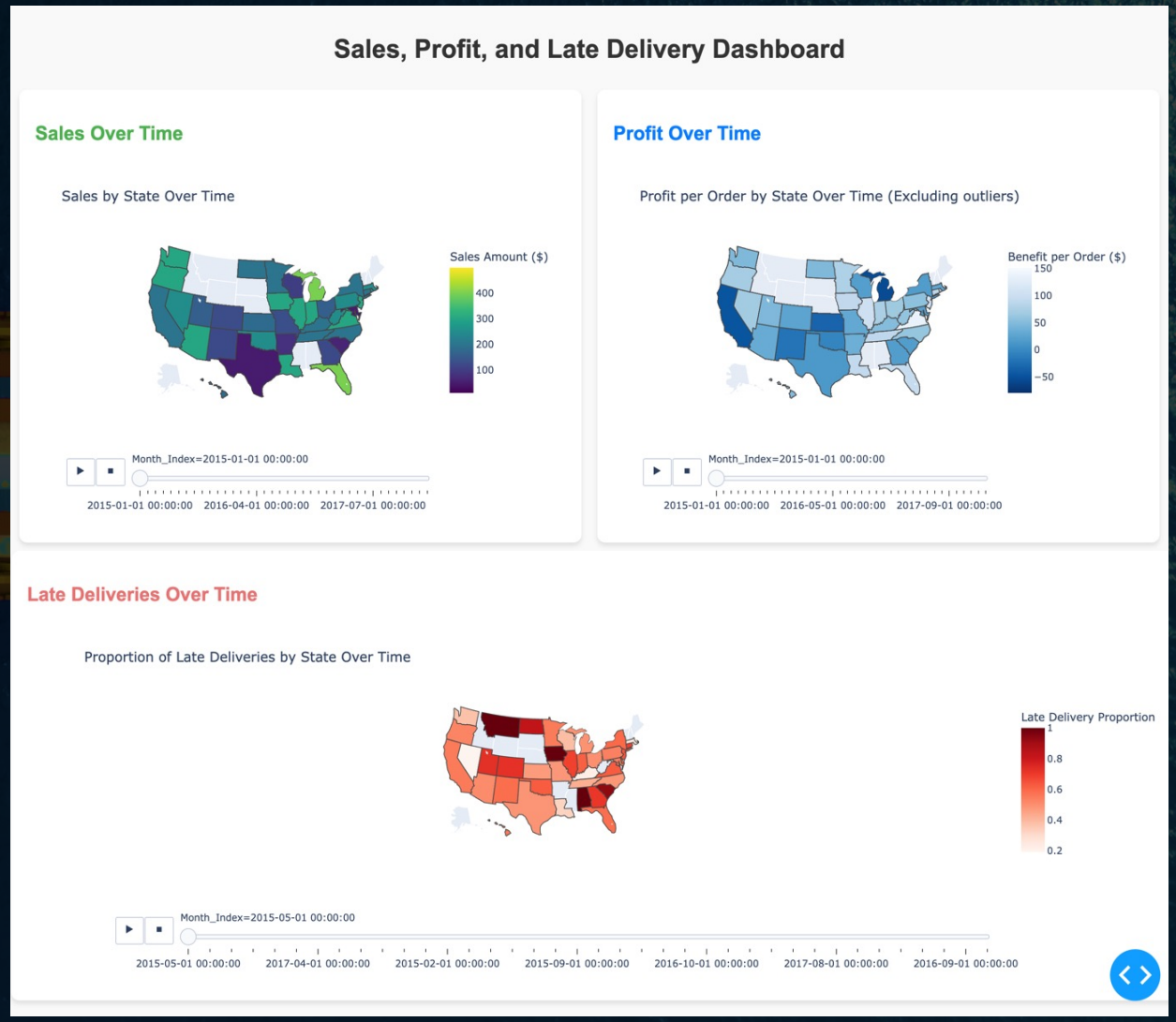
Correlation Visualizations





Dashboard

- Sales and profits over time of monthly increments
- Late delivery over time of monthly increments



Objective 1

Classifier Table

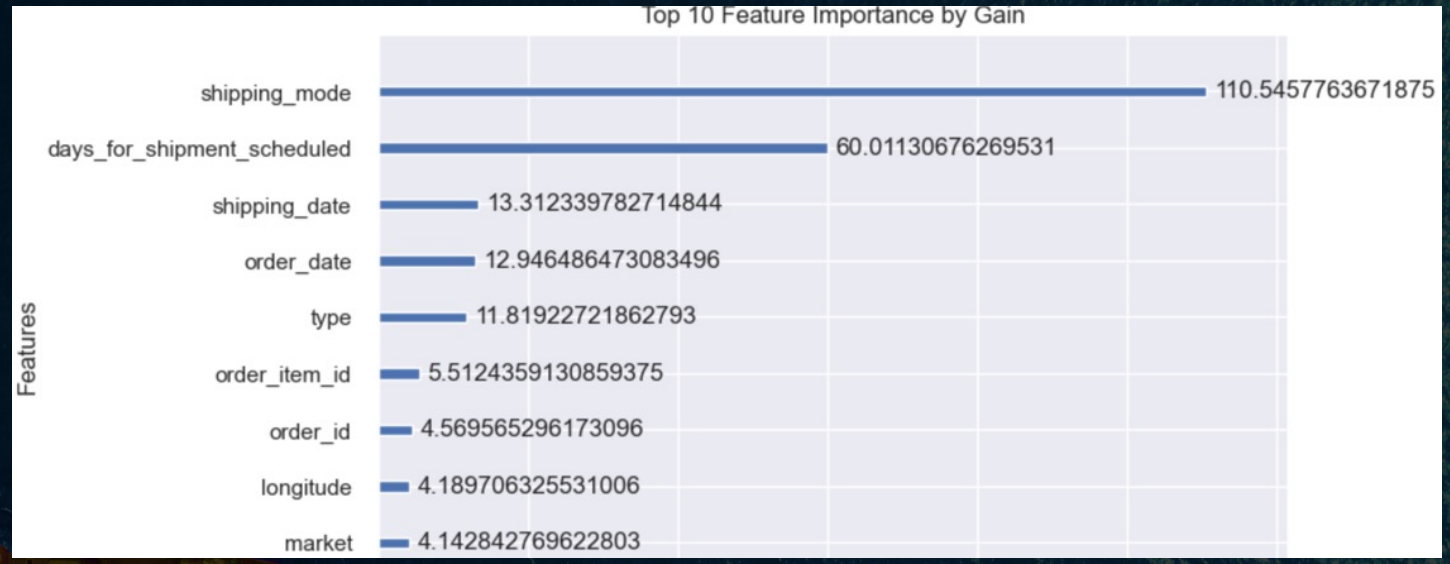
- I label encoded categorical features using only the training data to avoid data leakage.
- Unseen test set categories were assigned -1.
- I set a seed for reproducibility and used 50% of the data for training and validation.
- After splitting, I label encoded the categorical features.
- I trained multiple classifiers, calculated key metrics (accuracy, recall, precision, F1, ROC AUC), and handled errors by logging them.
- The results were visualized in a styled table, highlighting accuracy and other metrics with colors.

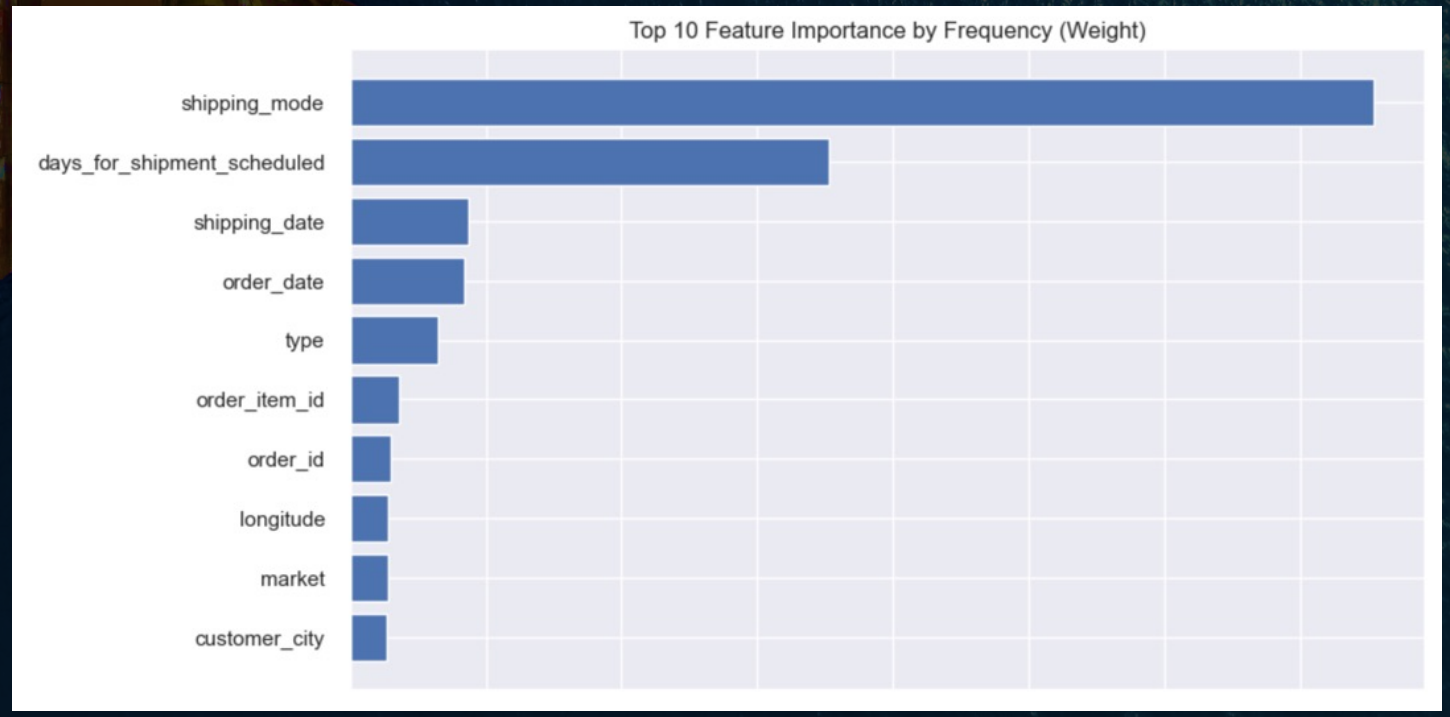
Models 1 – Random Forest

- As mentioned before
- I ran these three models through a grid search
- ran the models again to adjust for overfitting on a custom set of params.
- Used five fold cross validation on each
- 1 – Random Forest – More robust than decision trees
- 2 – KNN
- 3 – Xgboost – Was 4th / next in line

Model 3 – XGBoost Interpretation

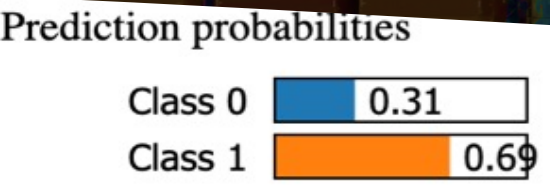
- Importance feature inbuilt in Xgboost
- showed shipping mode and days for shipment scheduled are the most important features, both frequently used and important in improving the model's accuracy

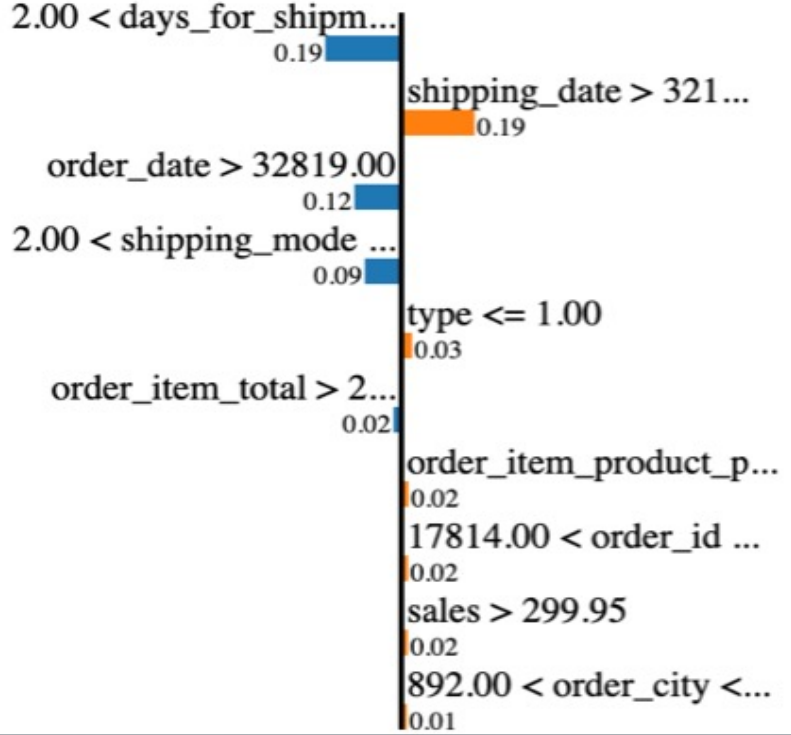




Model 3 - XGBoost Interpretation

- you schedule the delivery for 2 days or longer you reduce the likely hood of being late.
- If the order date was on or before 5/20/2017 8:56 the delivery is more likely to be late.
- If the shipping date was on or after 5/28/2016 11:22 the delivery is more likely to be late
- If it was shipped standard class, it was less likely to be late.
- If the payment type was a transfer, it was more likely to arrive late.





Feature	Value
days_for_shipment_scheduled	4.00
shipping_date	41700.00
order_date	41808.00
shipping_mode	3.00
type	1.00
order_item_total	287.98
order_item_product_price	299.98
order_id	18304.00
sales	299.98

Objective 2

PCA + Scaler

- Conducted dimension reduction and scaled the results

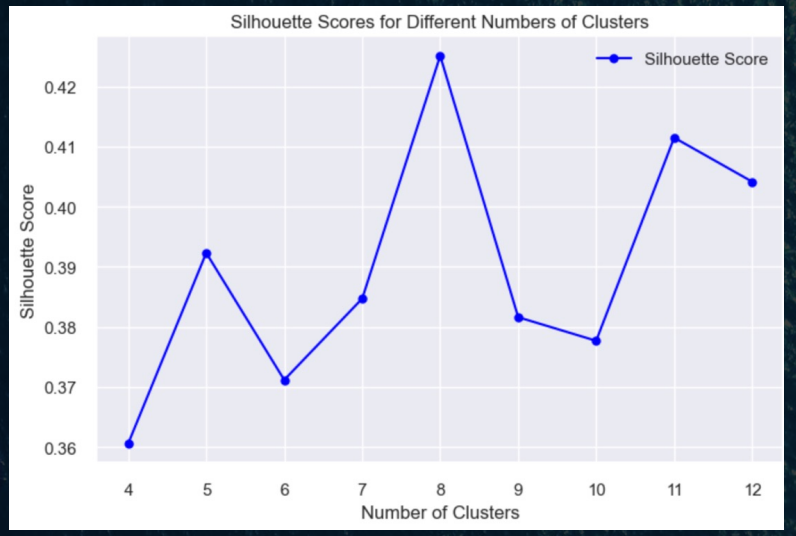
	count	mean	std	min	25%	50%	\
col1	180519.0	-4.408444e-17	2.500059	-5.050225	-1.747766	-0.923238	
col2	180519.0	-2.307151e-16	1.969002	-3.719338	-2.347304	0.636291	
col3	180519.0	-6.549688e-17	1.843902	-8.095556	-1.016276	0.012378	

	75%	max
col1	1.650840	23.351727
col2	1.645526	4.371669
col3	1.086839	21.566523

Elbow + Silhouette

Conducted the elbow method and silhouette scores for multiple steps to find the best cluster amount

Silhouette Score for 4 clusters	**0.36063281891071836**
Silhouette Score for 5 clusters	0.39232081582862216
Silhouette Score for 6 clusters	0.3711968097427292
Silhouette Score for 7 clusters	0.384721207737656
Silhouette Score for 8 clusters	0.42511132144901764
Silhouette Score for 9 clusters	0.38162718184276173
Silhouette Score for 10 clusters	0.3777018003592016
Silhouette Score for 11 clusters	0.4114949402948788
Silhouette Score for 12 clusters	0.4041694138446294



Cluster Analysis

- Interactive Plot of the Clusters
- Distribution of Clusters
- Cluster's Profile Based On Sales per customer And Item Quantity
- Cluster's Profile Based On Sales per customer And Profit per Order
- Cluster's Profile Based On Sales per customer And Product Price
- Cluster's Profile Based On Sales per customer And Shipping Date
- Shipping date reverse label encoding
- Cluster's Profile Based On Latitude And sales per customers
- Cluster's Profile Based On Longitude And sales per customers

Challenges Encountered

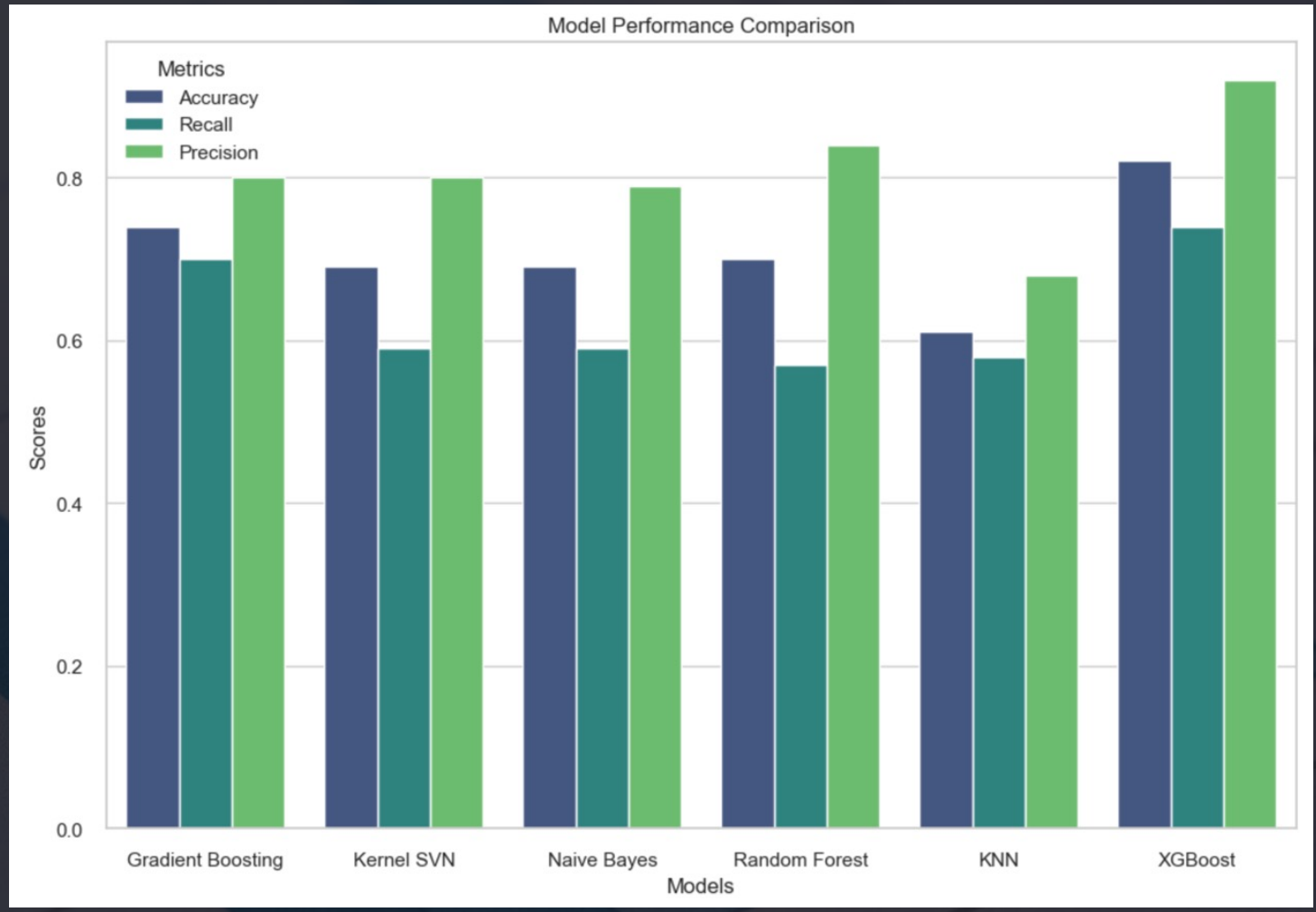
Tech

- 50% of data only
- Lack of domain Experts
- Time limitations
- Inability to collaborate

Results, analysis… next steps.

Results

- XGBoost performed the best out of CA2 and CA3, with the highest recall at 74%, highest accuracy at 92% and highest precision at 82%.
- I found 8 clusters with a moderate silhouette score, leading a insights and a deeper understanding of customers



Insights

- Cluster 6 is the clear high-value group, with high spending and profitability, making it a key segment for business focus.
- Clusters 0, 2, and 3 represent low-value customers who could benefit from strategies to increase profitability or minimize costs.
- Cluster 7 shows promise with moderate sales and profitability, making it a potential growth segment.
- **No.1 Action you can take today to drop late deliveries is increase the estimated delivery time to at least 3 days until supply chain team can improve the logistics**
- **Priority areas for logistics to look into is shipping mode, followed by seasonality of orders (shipping date & order date features)**

Next steps

- Experiment with further parameters
- In depth time series analysis of late deliveries
- In depth analysis of the datasets by location/market/region and supply chain features
- Collaborate with domain experts
- Further feature engineering
- Experimenting with costlier and more complex models
- Using the full dataset

Conclusion