Final Project - Update 4

Esmeralda, Divya, Steven

CS463

Team

- Divya D'Souza
- Steven Buks
- Esmeralda Hernandez De Loa

<u>Interest in project</u>: being able to understand and predict when a delivery is late will make you very valuable in any company that provides a physical product.

Description

- **Creating a model that can predict whether or not a shipment will arrive late or not. (Supply and Demand industry)**
- Given many variables in the supply chain industry, we will be able to investigate what features in this given dataset are most important in determining whether or not a shipment will be late
- This can give us insight into optimizations that can be made to the supply chain

Background

- Impacts of AI and ML in Supply Chain:
    - Optimization of inventory management
    - Proactive Risk Management
    - Cost Reduction and Increased Profitability

Wyrembek, Mateusz, et al. "Causal Machine Learning for Supply Chain Risk Prediction and Intervention Planning." International Journal of Production Research, Jan. 2025, pp. 1–20. EBSCOhost, https://doi.org/10.1080/00207543.2025.2458121.

- Breaks down the steps of implementing machine learning models to supply chain data
- Uses models to predict 'Delay'
    - Causal ML: aims to estimate causal effects as opposed to focus on prediction (ex CART)
        - ATE (Average Treatment Effect)
        - CATE (Conditional Average Treatment Effect)

Data

- Data Source: Kaggle
- Data had 53 columns and 180,519 entries
- Some entire columns had null values, we decided to drop these columns since they did not convey any meaningful information
- We then discussed which columns we could drop based on relevance
- Created target variable 'Late Arrival' based on 'Delay' which we engineered (unique)

EDA - **Before**

Some examples of the columns we dropped:

'Category Id', 'Category Name', 'Customer Email', 'Customer Fname', 'Customer Lname', 'Customer Id', 'Customer Segment', 'Customer Password', etc...

```
RangeIndex: 180519 entries, 0 to 180518
Data columns (total 53 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 Email                180519 non-null  object
 12  Customer Fname                180519 non-null  object
 13  Customer Id                   180519 non-null  int64
 14  Customer Lname                180511 non-null  object
 15  Customer Password             180519 non-null  object
 16  Customer Segment              180519 non-null  object
 17  Customer State                180519 non-null  object
 18  Customer Street               180519 non-null  object
 19  Customer Zipcode              180516 non-null  float64
 20  Department Id                 180519 non-null  int64
 21  Department Name               180519 non-null  object
 22  Latitude                      180519 non-null  float64
 23  Longitude                     180519 non-null  float64
 24  Market                        180519 non-null  object
 25  Order City                    180519 non-null  object

 26  Order Country                 180519 non-null  object
 27  Order Customer Id             180519 non-null  int64
 28  order date (DateOrders)       180519 non-null  object
 29  Order Id                      180519 non-null  int64
 30  Order Item Cardprod Id        180519 non-null  int64
 31  Order Item Discount           180519 non-null  float64
 32  Order Item Discount Rate      180519 non-null  float64
 33  Order Item Id                 180519 non-null  int64
 34  Order Item Product Price      180519 non-null  float64
 35  Order Item Profit Ratio       180519 non-null  float64
 36  Order Item Quantity           180519 non-null  int64
 37  Sales                         180519 non-null  float64
 38  Order Item Total              180519 non-null  float64
 39  Order Profit Per Order        180519 non-null  float64
 40  Order Region                  180519 non-null  object
 41  Order State                   180519 non-null  object
 42  Order Status                  180519 non-null  object
 43  Order Zipcode                 24840 non-null   float64
 44  Product Card Id               180519 non-null  int64
 45  Product Category Id           180519 non-null  int64
 46  Product Description           0 non-null       float64
 47  Product Image                 180519 non-null  object
 48  Product Name                  180519 non-null  object
 49  Product Price                 180519 non-null  float64
 50  Product Status                180519 non-null  int64
 51  shipping date (DateOrders)    180519 non-null  object
 52  Shipping Mode                 180519 non-null  object
```

EDA - **After**

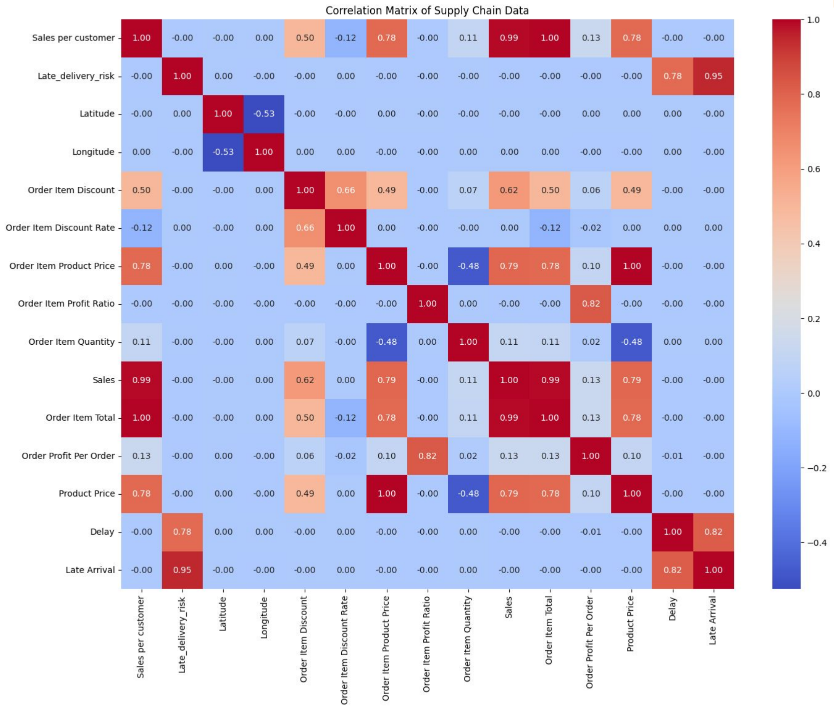
Some examples of the columns we dropped:

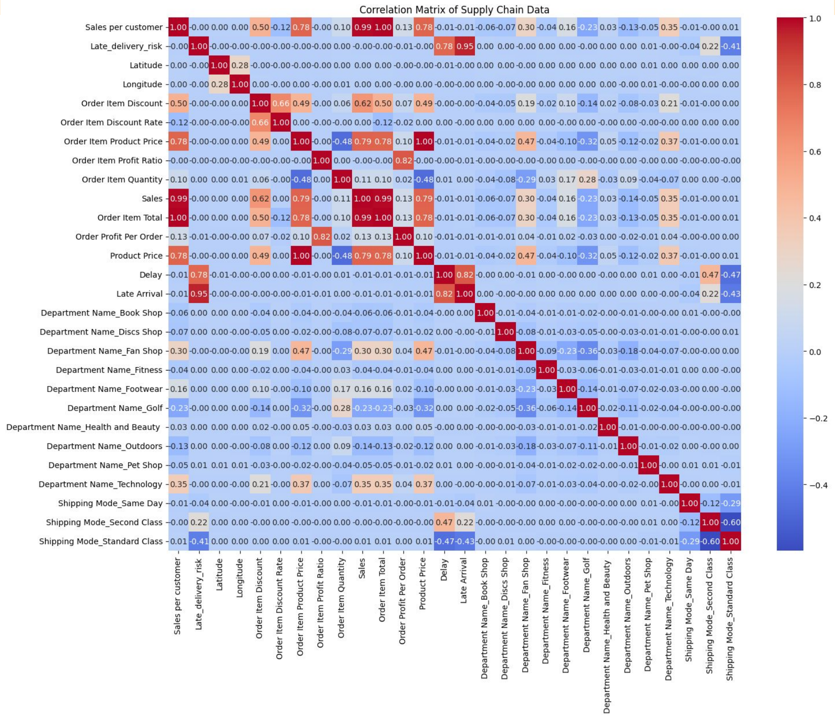
'Category Id', 'Category Name', 'Customer Email', 'Customer Fname', 'Customer Lname', 'Customer Id', 'Customer Segment', 'Customer Password', etc...

```
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   Type                           180519 non-null  object
 1   Sales per customer             180519 non-null  float64
 2   Late_delivery_risk             180519 non-null  int64
 3   Customer City                  180519 non-null  object
 4   Customer Country               180519 non-null  object
 5   Customer State                 180519 non-null  object
 6   Department Id                  180519 non-null  int64
 7   Department Name                180519 non-null  object
 8   Latitude                       180519 non-null  float64
 9   Longitude                      180519 non-null  float64
 10  Order City                     180519 non-null  object
 11  Order Country                  180519 non-null  object
 12  order date (DateOrders)        180519 non-null  object
 13  Order Item Discount            180519 non-null  float64
 14  Order Item Discount Rate       180519 non-null  float64
 15  Order Item Product Price       180519 non-null  float64
 16  Order Item Profit Ratio        180519 non-null  float64
 17  Order Item Quantity            180519 non-null  int64
 18  Sales                          180519 non-null  float64
 19  Order Item Total               180519 non-null  float64
 20  Order Profit Per Order         180519 non-null  float64
 21  Order State                    180519 non-null  object
 22  Order Status                   180519 non-null  object
 23  Order Zipcode                  24840 non-null   float64
 24  Product Price                  180519 non-null  float64
 25  shipping date (DateOrders)     180519 non-null  object
 26  Shipping Mode                  180519 non-null  object
 27  Delay                          180519 non-null  int64
```

EDA

Some heatmaps!

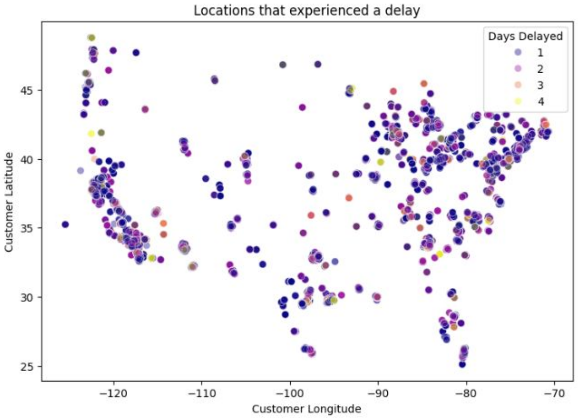


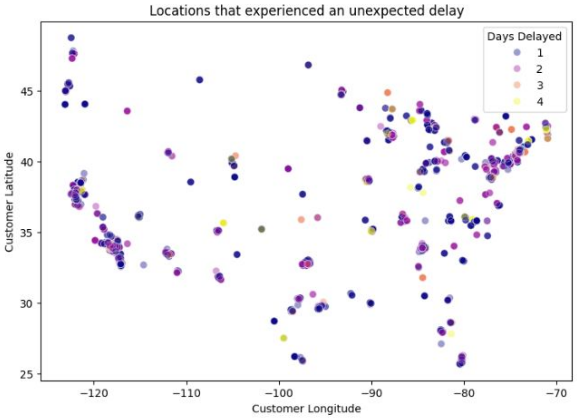


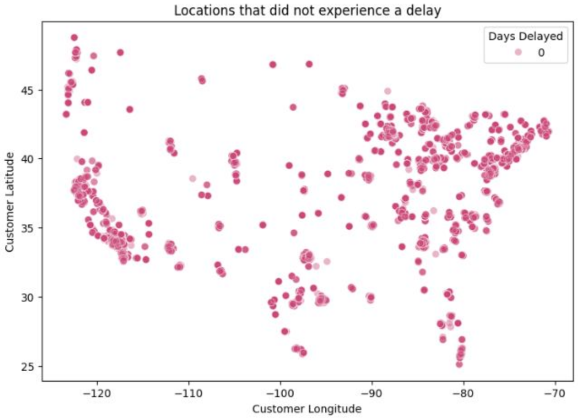
EDA

Do delivery locations impact late arrival?

Spoiler - **NO**







EDA Conclusions:

- No valid correlations between delay and other initial numeric variables.
- After encoding, delays for
  Second Class shipment >>> *Standard* Class shipment
- *Standard* Class shipments have a negative correlation with delay which tells us that they are least likely to be delayed, even over Same Day deliveries
- No trends in delivery location specific delays

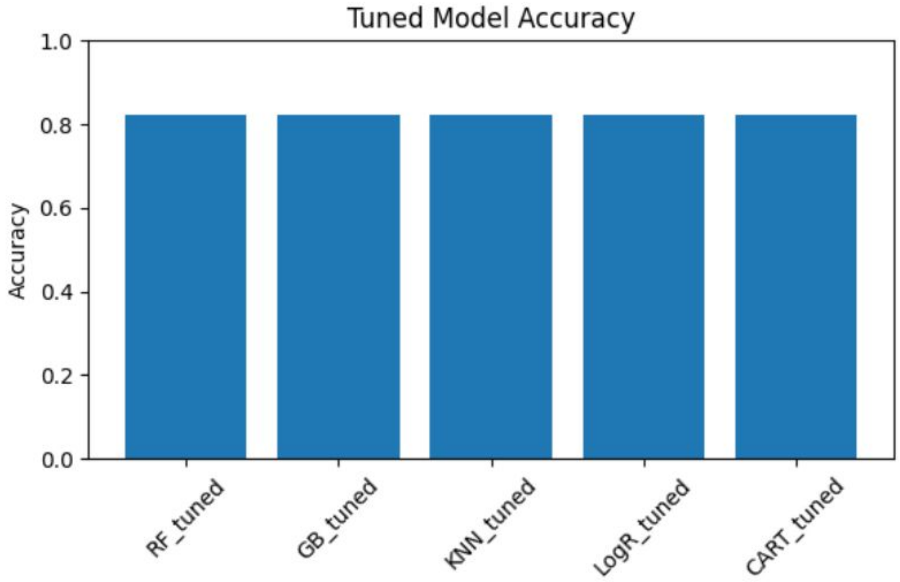
Feature Engineering

To analyze trends, we:

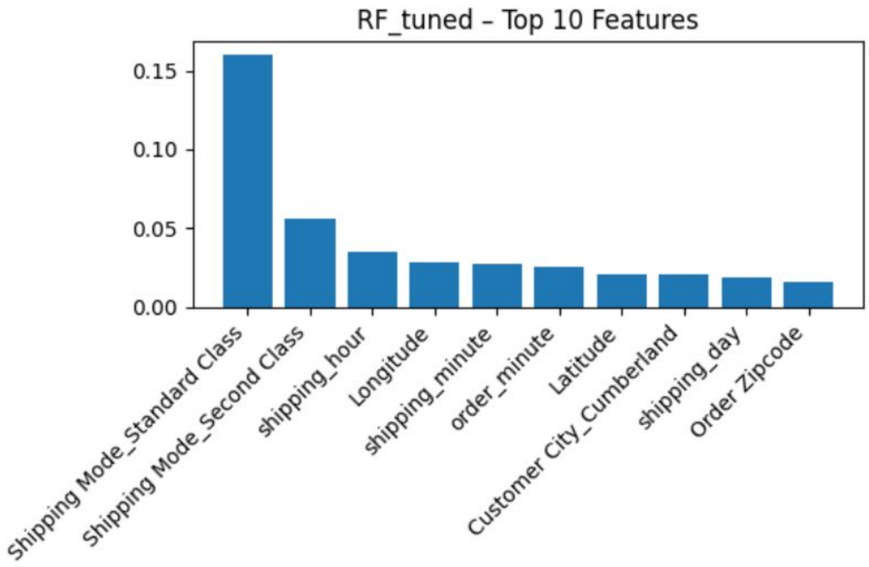
- One-hot-encoded:
  - Department Name, Shipping Mode, Customer City, Customer Country, Order City, Order Country, Customer Country, Order State, Order Country
- Extracted date-time information from '*order date (DateOrders)*' and '*shipping date (DateOrders)*' (unique)
  - Made separate columns for day, month, year, time of day to look for trends and patterns
- We now have **4661 numeric and bool columns** ready for Ensambles
- Dropped 'Delay' and 'Late_delivery_risk' columns for running models

Models

- Random Forest
- Gradient Boosting
- KNN
- Logistic regression
- CART



Models - Random Forest



GridSearchCV

best_estimator_: RandomForestClassifier

```
RandomForestClassifier(bootstrap=False, max_depth=10, min_samples_split=5,
                       n_estimators=25, random_state=42)
```

RandomForestClassifier

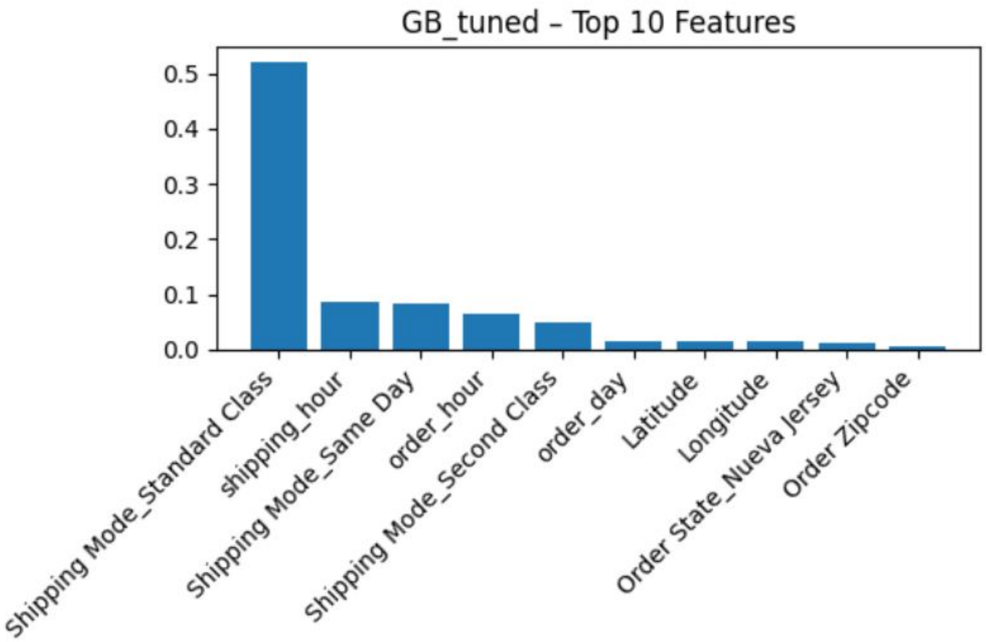
```
RF_tuned Accuracy: 0.8238
              precision    recall  f1-score   support

           0       0.82      0.22      0.34       250
           1       0.63      0.97      0.76       346

    accuracy                           0.65       596
   macro avg       0.72      0.59      0.55       596
weighted avg       0.71      0.65      0.59       596

Confusion Matrix:
 [[ 54 196]
 [ 12 334]]
```

Models - Gradient Boosting




```
GridSearchCV
best_estimator_: GradientBoostingClassifier
GradientBoostingClassifier(n_estimators=35, random_state=42)
GradientBoostingClassifier
```

```
GB_tuned Accuracy: 0.8238
              precision    recall  f1-score   support

           0       0.62      0.89      0.73       250
           1       0.89      0.60      0.72       346

    accuracy                           0.72       596
   macro avg       0.75      0.75      0.72       596
weighted avg       0.77      0.72      0.72       596

Confusion Matrix:
 [[223  27]
 [138 208]]
```

Models - KNN

```
▸ GridSearchCV
▾ best_estimator_: KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3, weights='distance')
▸ KNeighborsClassifier
```

```
KNN_tuned Accuracy: 0.8238
              precision    recall  f1-score   support

           0       0.63      0.63      0.63       250
           1       0.73      0.73      0.73       346

    accuracy                           0.69       596
   macro avg       0.68      0.68      0.68       596
weighted avg       0.69      0.69      0.69       596

Confusion Matrix:
 [[158  92]
 [ 93 253]]
```

Models - Logistic Regression

GridSearchCV

best_estimator_: LogisticRegression

LogisticRegression(C=10, max_iter=1000)

LogisticRegression

```
LogR_tuned Accuracy: 0.8238
              precision    recall  f1-score   support

           0       0.68      0.74      0.71       250
           1       0.80      0.75      0.77       346

    accuracy                           0.74       596
   macro avg       0.74      0.74      0.74       596
weighted avg       0.75      0.74      0.74       596

Confusion Matrix:
 [[184  66]
 [ 87 259]]
```

Models - CART

```
GridSearchCV
best_estimator_: DecisionTreeClassifier
DecisionTreeClassifier(max_depth=10, min_samples_split=5, random_state=42)
DecisionTreeClassifier
```

```
CART_tuned Accuracy: 0.8238
              precision    recall  f1-score   support

           0       0.73      0.76      0.74       250
           1       0.82      0.79      0.81       346

    accuracy                           0.78       596
   macro avg       0.77      0.78      0.77       596
weighted avg       0.78      0.78      0.78       596

Confusion Matrix:
 [[190  60]
 [ 72 274]]
```

Results

- Best Model: **CART**
  - Has highest f1-score
    - Good balance between precision and recall
    - <u>Low false positives</u>: most positive predictions were correct
    - <u>Low false negatives</u>: most actual positive were correctly predicted