Machine Learning Prediction in Tracking Product Delivery

Capstone Project 4
Michelle Ang

Content

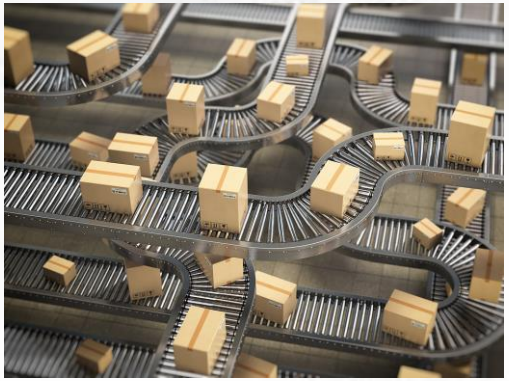
1. Introduction
2. Methodology
3. Process Workflow
   Data Preparation, Visualisation, Data Tranformation & ML Model Training & Evaluation
4. Results
5. Conclusion

Problem Statement

An ecommerce company want to understand key insights from their Product Shipment Tracking data.

Classification Problem:
What factors determine if the product will be delivered on time? (Yes/No)

Goal: Make better improvements on the customer care service & delivery pattern in advance.



Target Audience: Shipping Department of the Ecommerce Company

Methodology

Models

- **Baseline Model**: Logistic Regression
- **Other Models**: Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Support Vector Classification (SVC), Naïve Bayes, Multi-Level Perceptron

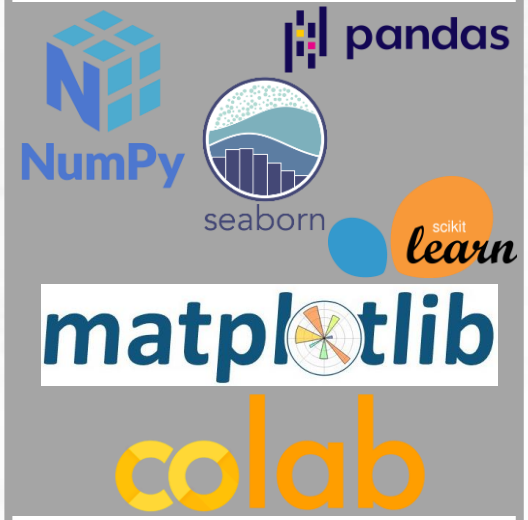
Dataset

kaggle

Metrics

Precision, Recall, F1 Score

Tools

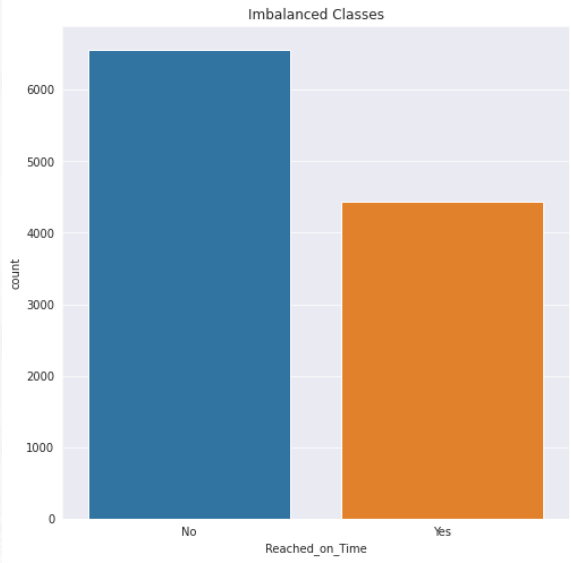


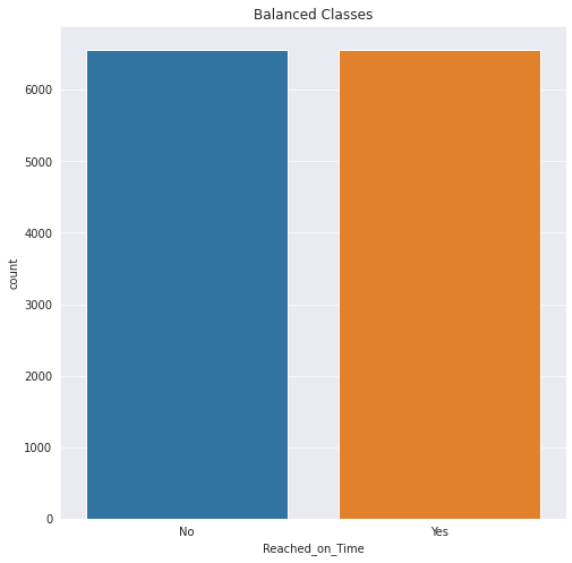
Data Preparation

- Check for non-null value
  Dropping Unnecessary Column
-> Customer Rating /ID

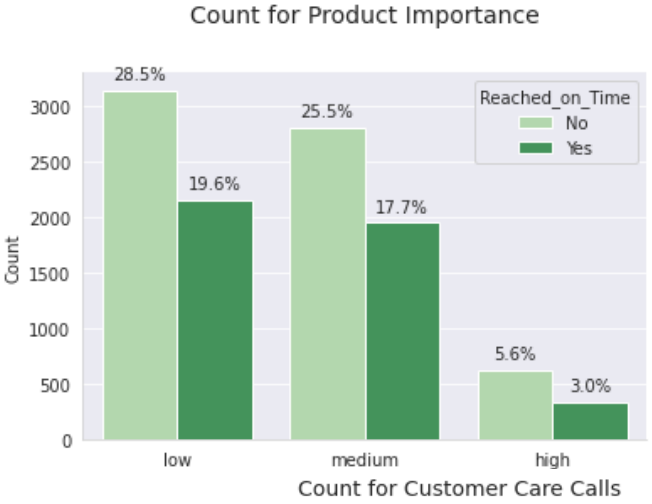
- Slightly Imbalanced Dataset on Target Variable
  -> Reached on Time (Oversampling: SmoteNC)

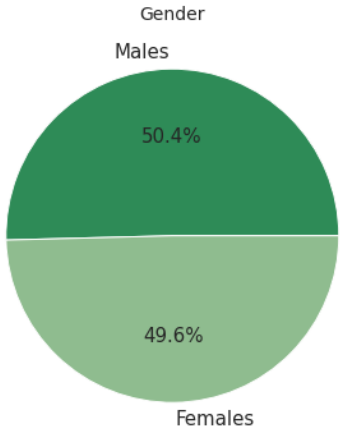
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   10999 non-null  int64
 1   Warehouse_block      10999 non-null  object
 2   Mode_of_Shipment     10999 non-null  object
 3   Customer_care_calls  10999 non-null  int64
 4   Customer_rating      10999 non-null  int64
 5   Cost_of_the_Product  10999 non-null  int64
 6   Prior_purchases      10999 non-null  int64
 7   Product_importance   10999 non-null  object
 8   Gender               10999 non-null  object
 9   Discount_offered     10999 non-null  int64
 10  Weight_in_gms        10999 non-null  int64
 11  Reached_on_Time      10999 non-null  int64
dtypes: int64(8), object(4)
memory usage: 1.0+ MB
```



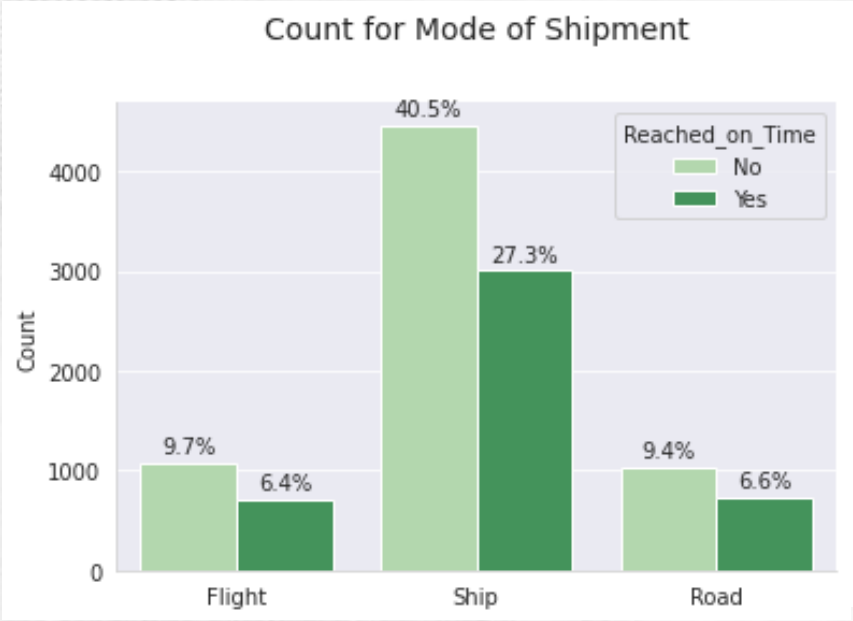


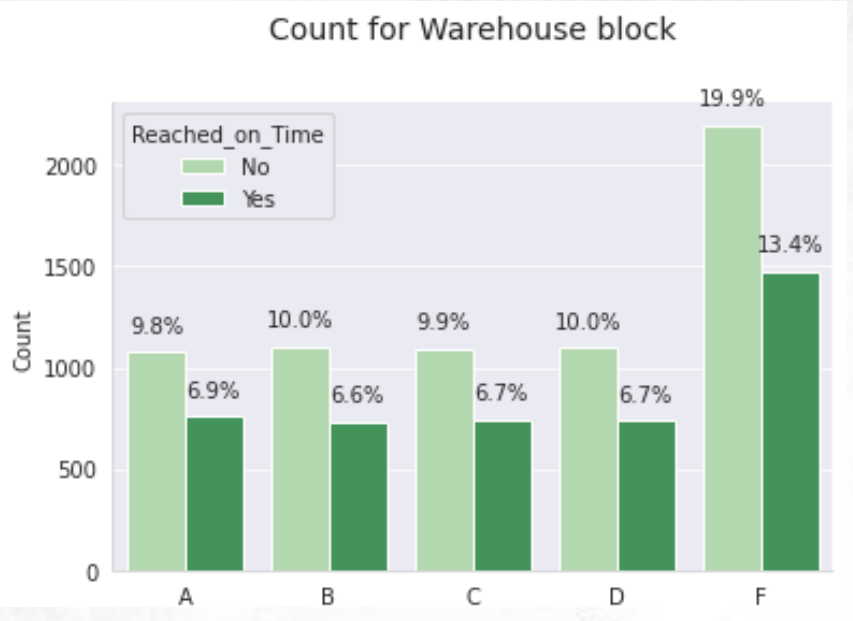
Visualisation





Visualisation





Data Transformation

- Ordinal Categories Feature (OrdinalEncoder)
- Nominal Categories Feature (Dummies)

Warehouse_block	Mode_of_Shipment	Customer_care_calls	Cost_of_the_Product	Prior_purchases	Product_importance	Gender	Discount_offered	Weight_in_gms	Reached_on_Time
D	Flight	4	177	3	low	F	44	1233	No
F	Flight	4	216	2	low	M	59	3088	No
A	Flight	2	183	4	low	M	48	3374	No
B	Flight	3	176	4	medium	M	10	1177	No
C	Flight	2	184	3	medium	F	46	2484	No

- Split the data into train and test datasets (test size = 0.2)
- Data Normalisation using Standard Scaler

Why is a Balancing Dataset Important?

Before Balancing:

Model	Precision	Recall	F1 Score
Decision Tree	0.738881	0.718360	0.680155
Random Forest	0.714576	0.707081	0.678133
Multilevel Perceptron	0.722253	0.703581	0.666045
Naive Bayes- Mixed Naive Bayes	0.694583	0.683713	0.651548
Naive Bayes- GaussianNB()	0.684346	0.678683	0.651380
SVM	0.634397	0.636470	0.635019
Logistic Regression	0.630118	0.631039	0.630506
KNN	0.602046	0.595665	0.596182

After Balancing:

Model	Precision	Recall	F1 Score
Random Forest	0.793453	0.729217	0.717773
Decision Tree	0.815173	0.731281	0.716868
Multilevel Perceptron	0.781763	0.710486	0.695405
Naive Bayes- Mixed Naive Bayes	0.762884	0.697588	0.681897
Logistic Regression	0.716799	0.687823	0.680050
SVM	0.718266	0.686478	0.677791
KNN	0.678958	0.673432	0.672189
Naive Bayes- GaussianNB()	0.792144	0.692314	0.668720

Machine Learning Model Training: Logistics Regression

- Best Estimator after hyperparameter using GridSearchCV

```
# Using best estimator found by GridSearchCV
logreg = gs_logreg.best_estimator_
logreg.fit(X_train_scaled, y_train)
```

```
LogisticRegression(C=0.01, n_jobs=-1, random_state=0, solver='newton-cg')
```

- Result for F1 score is 68%

Model	Training Acc	Testing Acc	Precision	Recall	F1 Score
Logistic Regression	0.692762	0.692308	0.716799	0.687823	0.680050

Machine Learning Model Training: Random Forest

- Best Estimator after hyperparameter using GridSearchCV

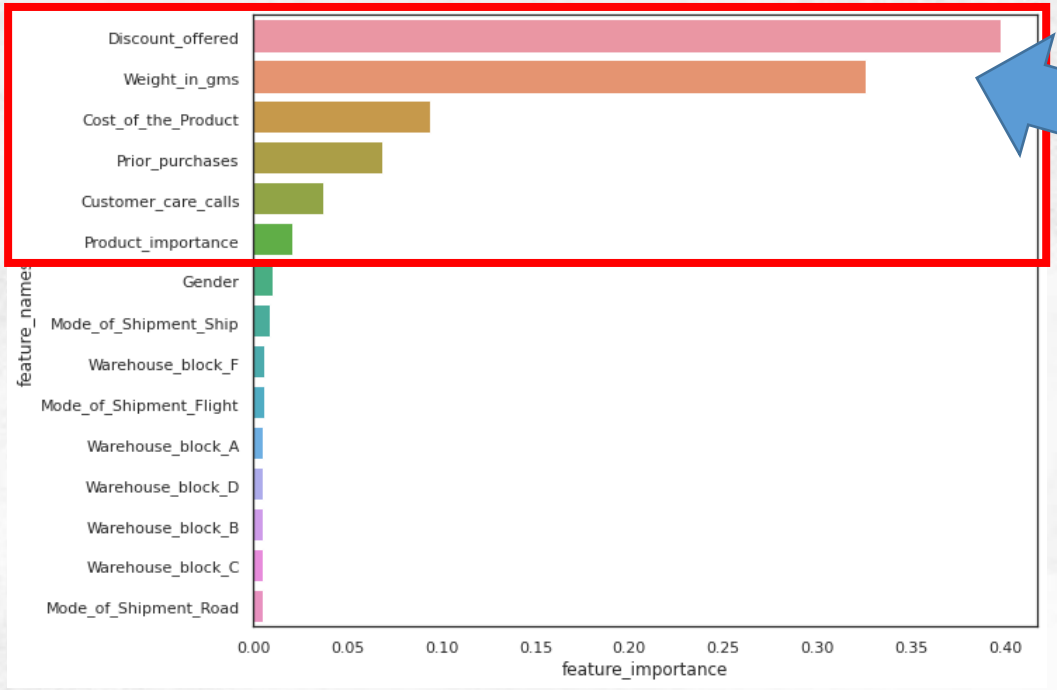
```
# Using best estimator found by GridSearchCV
rf = gs_rf.best_estimator_
rf.fit(X_train_scaled, y_train)
```

```
RandomForestClassifier(max_depth=10, n_estimators=200, n_jobs=-1,
                       random_state=0)
```

- Improved result for F1 score to 72%

Model	Training Acc	Testing Acc	Precision	Recall	F1 Score
Random Forest	0.785429	0.734958	0.793453	0.729217	0.717773

1. Feature Importance by Model Coefficient



75%

Model	Training Acc	Testing Acc	Precision	Recall	F1 Score
Random Forest After Feature Impt	0.736095	0.737624	0.810123	0.731434	0.717861
Random Forest	0.785429	0.734958	0.793453	0.729217	0.717773

2. Feature Selection by using SelectKBest

```
[45] new_features

['Warehouse_block_F',
 'Mode_of_Shipment_Flight',
 'Mode_of_Shipment_Road',
 'Mode_of_Shipment_Ship',
 'Product_importance',
 'Cost_of_the_Product',
 'Discount_offered',
 'Weight_in_gms']
```

```
[46] #original features
     feature_names

['Warehouse_block_A',
 'Warehouse_block_B',
 'Warehouse_block_C',
 'Warehouse_block_D',
 'Warehouse_block_F',
 'Mode_of_Shipment_Flight',
 'Mode_of_Shipment_Road',
 'Mode_of_Shipment_Ship',
 'Product_importance',
 'Customer_care_calls',
 'Cost_of_the_Product',
 'Prior_purchases',
 'Gender',
 'Discount_offered',
 'Weight_in_gms']
```

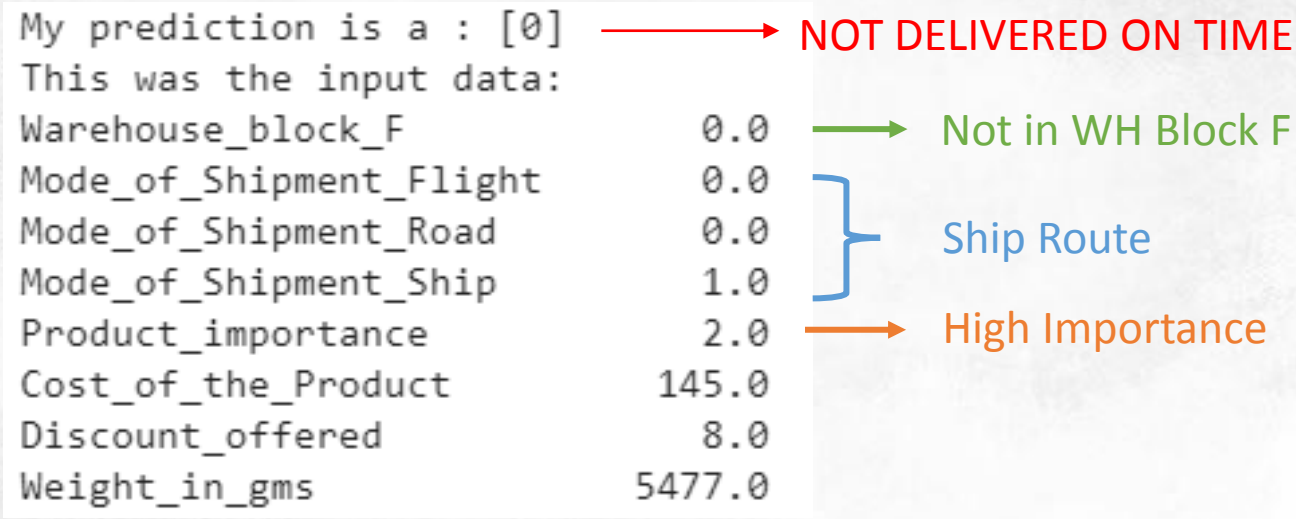
Before selecting the best features:

Model	Precision	Recall	F1 Score
Random Forest	0.793453	0.729217	0.717773
Decision Tree	0.815173	0.731281	0.716868
Multilevel Perceptron	0.781763	0.710486	0.695405
Naive Bayes- Mixed Naive Bayes	0.762884	0.697588	0.681897
Logistic Regression	0.716799	0.687823	0.680050

After selecting the best features:

Model	Precision	Recall	F1 Score
Random Forest	0.796095	0.731913	0.720778
Decision Tree	0.757352	0.721650	0.714891
Multilevel Perceptron	0.779334	0.707380	0.691728
Naive Bayes- Mixed Naive Bayes	0.753355	0.697170	0.683457
SVM	0.726547	0.691679	0.682625

Random Forest Model Prediction



Conclusion

- Random Forest Model has the best F1 Score after SelectKBest

Model	Training Acc	Testing Acc	Precision	Recall	F1 Score
Random Forest	0.784286	0.737624	0.796095	0.731913	0.720778

- Average Precision: 80%, Average Recall: 73% & F1-Score: 72%
- Top 2 Feature Importance: Discount Offered and Weights of the Product
- Prediction model in Tracking Product Delivery: Improved customer relation, improved delivery toward the area and improved manpower in the warehouse

Future Recommendation

- More fine tuning of the model & advanced algorithms
- Have more features
    - Distance from the delivery area
    - Manpower in the warehouse
    - Volume of shipment per transport
    - Proximity to sales period
    - Delivery hours
    - Weather condition
    - Extreme Traffic condition (Suez Canal)

Thank You!

Link:
https://www.kaggle.com/prachi13/customer-analytics