Final Project

E_Commerce Shipping Data

_Aulia_Fauzani_Mukti_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_Abdul_Hanif_Akmaluddin_ _ _ _ _ _ _ _ _ _ _ _ _ _
_Bintang_Muhammad_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_Fadhilla_Atansa_Tamardina__ _ _ _ _ _ _ _ _ _ _
_Lucky_Wijaya_Pengestu_ _ _ _ _ _ _ _ _ _ _ _ _ _
_Rifanda_Dwi_Putra_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Table of Contents

underscore

underscore

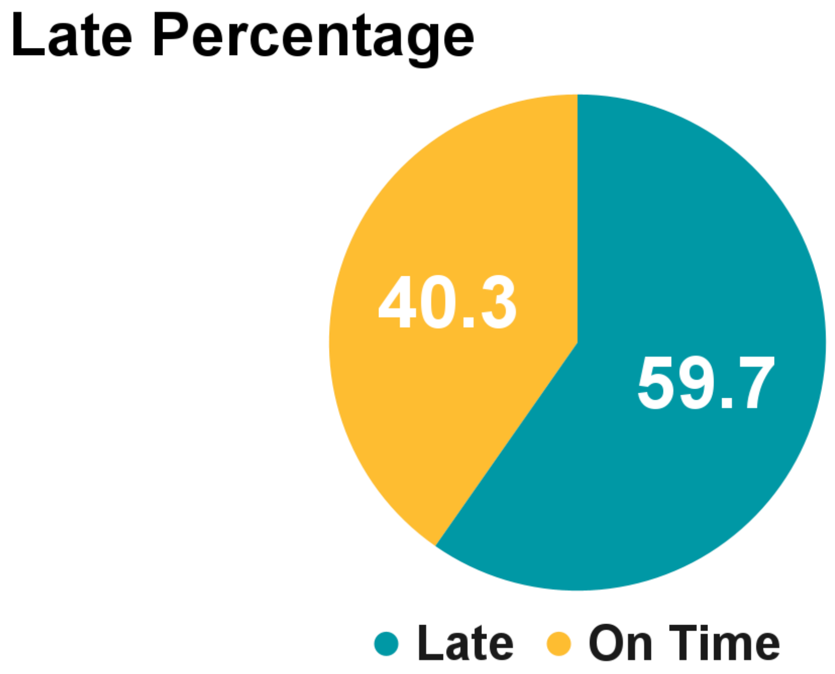
Data Consultant

An international e_commerce company that sells electronic products call _underscore_ to discover key insights & studies from their customer database.

underscore

Background

Problem



59.7% of E_Commerce deliveries are **late**.

6563 of 10999 Customers

Background

Problem

Source :
MHL News & Last Mile
Convey's 2018 Last-Mile Delivery Report

underscore

Late

87% online shoppers identified **shipping speed** as a **key factor** for online shoppers to shop again.

Dissatisfaction

In fact, price is not even as important as speed since **67%** online shoppers **would pay more** to get same day delivery.

Stop Shopping

84% online shoppers are **unlikely to return** after a poor delivery experience.

55% online shoppers will **stop shopping** after receiving late delivery twice.

Revenue Loss

Potential profits **will lose** because the customer left.

52% online shoppers expect a **refund** or discount on shipping cost after receiving late delivery.

underscore

underscore

Background

Problem

underscore as a data consultant will analyze insight & make predictions model about whether the delivery will be received late / on time by the customer to help solve e_commerce shipping problems.

underscore

underscore

Background

Problem

Current Condition

Most of the E_Commerce **Deliveries** are **not reached on time.**

Machine Learning Approach

Insight	Action	Impact
Finding pattern from database feature	**Predictive model**	**Insight & Recommendation**

Business Approach

Action	Business Impact
Recommendation analysis & decision	**On time rate, customer satisfaction & safe potential revenue loss**

underscore

underscore

Data Understanding

Describe

```
1 df.info()
```

```
<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    10999 non-null  object
 2   Shipment     10999 non-null  object
 3   Calls        10999 non-null  int64 
 4   Rating       10999 non-null  int64 
 5   Cost         10999 non-null  int64 
 6   Purchase     10999 non-null  int64 
 7   Importance   10999 non-null  object
 8   Gender       10999 non-null  object
 9   Discount     10999 non-null  int64 
 10  Weight       10999 non-null  int64 
 11  Late         10999 non-null  int64 
dtypes: int64(8), object(4)
memory usage: 1.0+ MB
```

10999

Rows

12

Columns

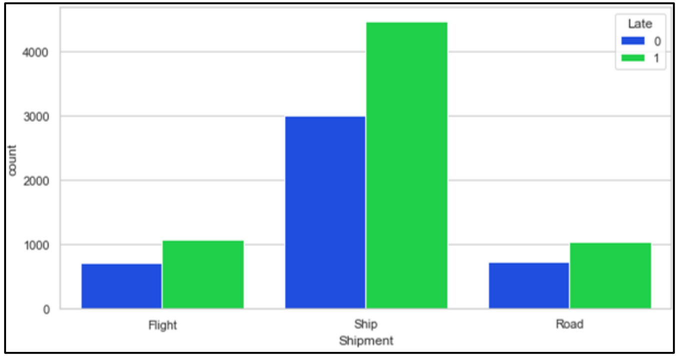
```
cats = ['ID', 'Warehouse', 'Shipment', 'Rating', 'Importance', 'Gender', 'Late']
nums = ['Calls', 'Cost', 'Purchase', 'Discount', 'Weight', ]
```

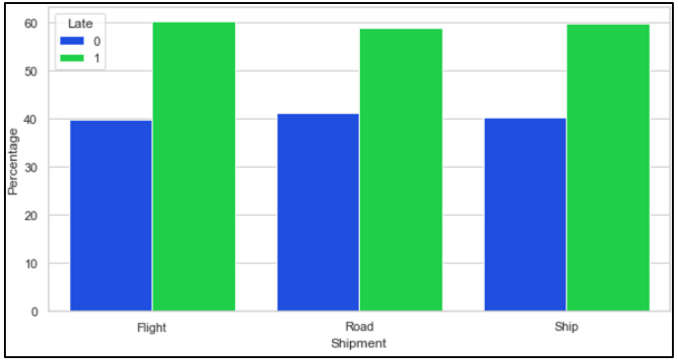
underscore

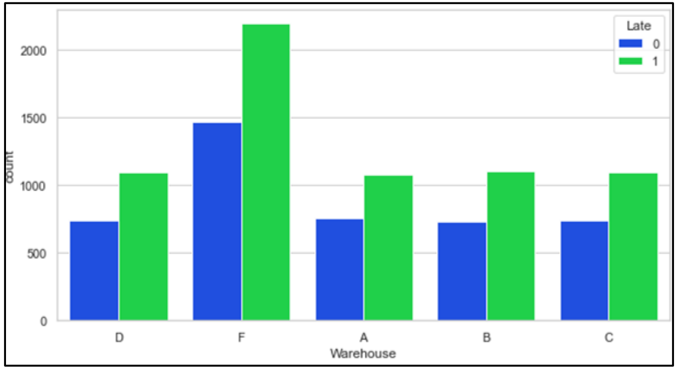
Data Understanding

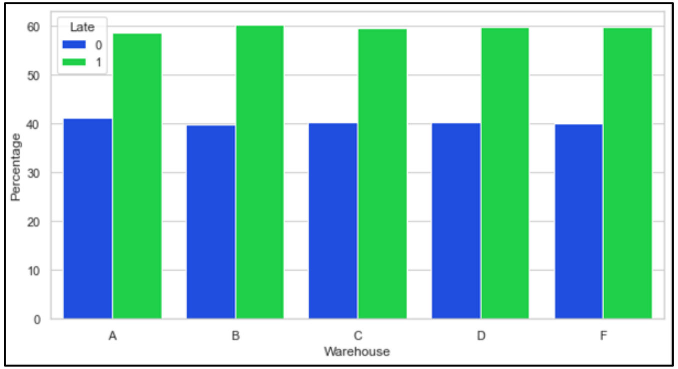
Feature

0 = On Time
1 = Late







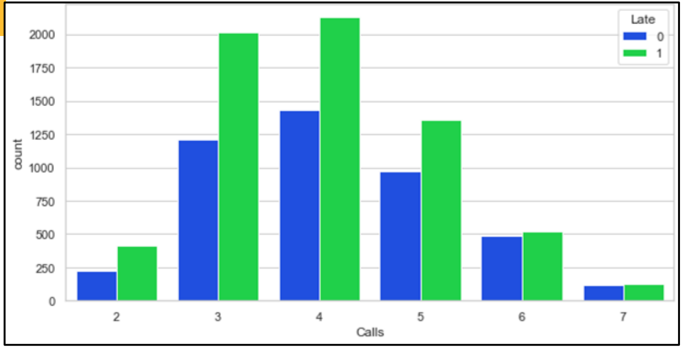


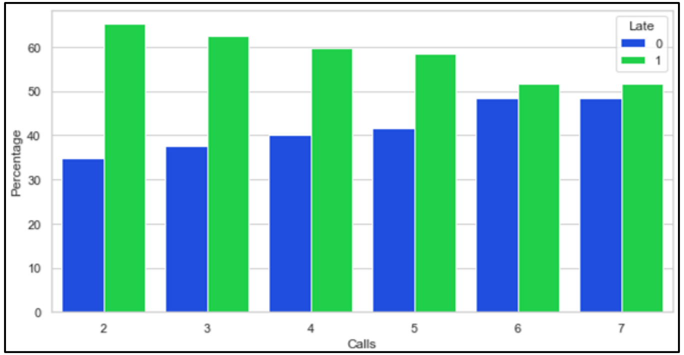
Ship & Warehouse F has the highest frequency of delivery. But it looks almost the same based on the percentage. There's an assumtion that the late is influenced by other factors.

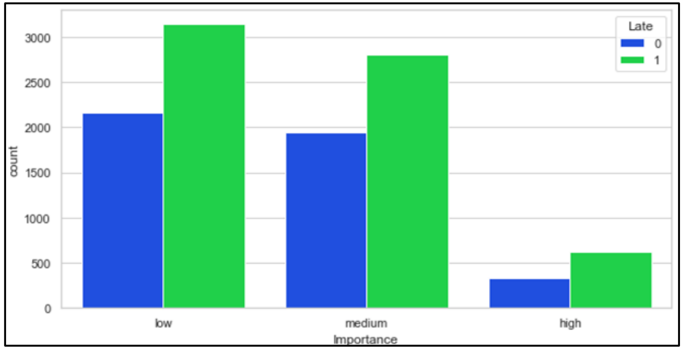
Data Understanding

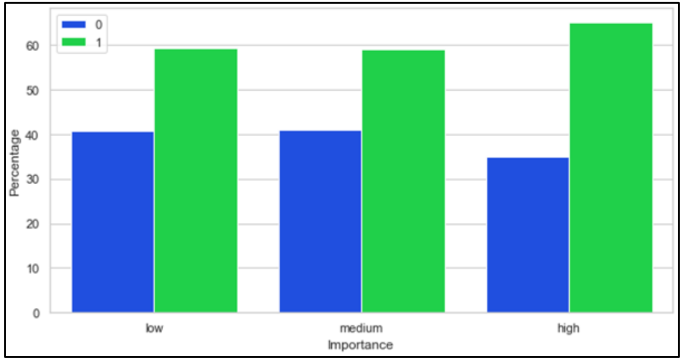
Feature

0 = On Time
1 = Late







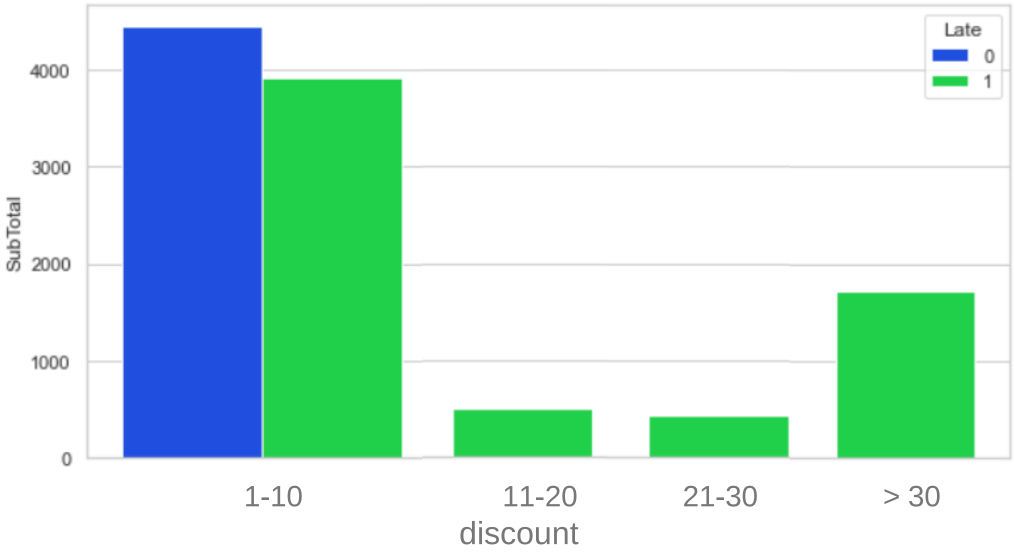


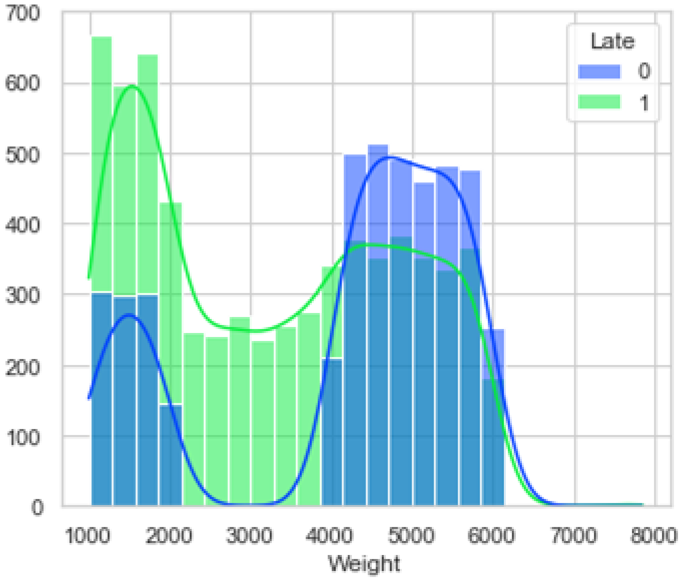
By percentage, lateness based on calls as well as based on importance is almost identical. **The more calls are the less late delivery.**

underscore

Data Understanding

Feature





- Every product that gets a discount above 10 is confirmed Late. There is an assumption that this happens in specific months, but needs further checking.
- Shipping delivery is confirmed late when the product weight is between 2-4 kg.

0 = On Time
1 = Late

underscore

Data Understanding

Feature

Conclusion: There are no redundant features as no features are have strong value above 0.7.

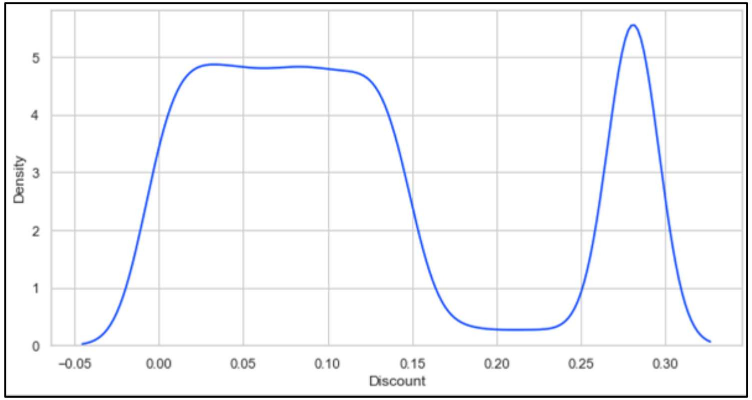
Process & Model

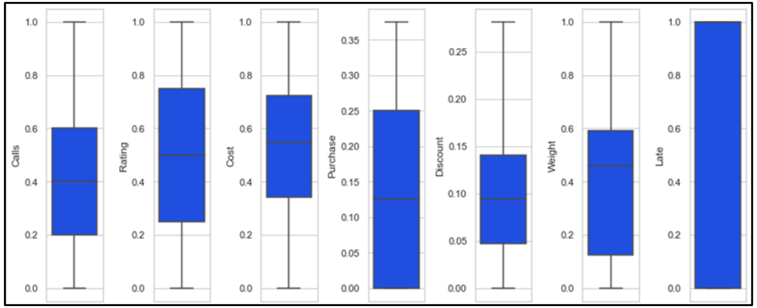
Data Processing

Missing & Duplicate	No missing & duplicate values in dataset
Outliers	Remove & replace outliers based on IQR limit
Selection	Drop 'ID' feature which has unique number
Encoding	One hot & ordinal encoding for categorical feature
Normalization	Normalization for numerical feature

Process & Model

Data Processing






```
df_project.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 9996 entries, 0 to 10998
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Calls             9996 non-null   float64
 1   Rating            9996 non-null   float64
 2   Cost              9996 non-null   float64
 3   Purchase          9996 non-null   float64
 4   Importance        9996 non-null   float64
 5   Discount          9996 non-null   float64
 6   Weight            9996 non-null   float64
 7   Late              9996 non-null   float64
 8   Warehouse_A       9996 non-null   float64
 9   Warehouse_B       9996 non-null   float64
 10  Warehouse_C       9996 non-null   float64
 11  Warehouse_D       9996 non-null   float64
 12  Warehouse_F       9996 non-null   float64
 13  Shipment_Flight   9996 non-null   float64
 14  Shipment_Road     9996 non-null   float64
 15  Shipment_Ship     9996 non-null   float64
 16  Gender_F          9996 non-null   float64
 17  Gender_M          9996 non-null   float64
```

- Replace outliers for `Discount` Feature with IQR Limit.
- Remove outliers for `Purchase` feature by IQR Limit
- Displayed in the boxplot that there are **no outliers appeared** after data processing.

Process & Model

Modeling Result

Primary : Recall
Secondary : Average Precision

underscore

	Random Forest	Logistic Regression	AdaBoost	XGBoost
Accuracy	0.67	0.64	0.66	0.64
Precision	0.84	0.69	0.78	0.72
Recall	0.57	0.73	0.60	0.68
F1-Score	0.68	0.71	0.68	0.70
ROC AUC	0.70	0.62	0.68	0.66
AP	0.74	0.67	0.70	0.68
AP Train	0.74	0.67	0.72	0.93
AP Test	0.74	0.67	0.70	0.68

Logistic Regression

	Predicted Label	
Actual Label	TRUE POSITIVE (TP) 1318 43.95%	FALSE NEGATIVE (FN) 498 16.61%
	FALSE POSITIVE (FP) 581 19.37%	TRUE NEGATIVE (TN) 602 20.07%

underscore

Process & Model

Modeling Result

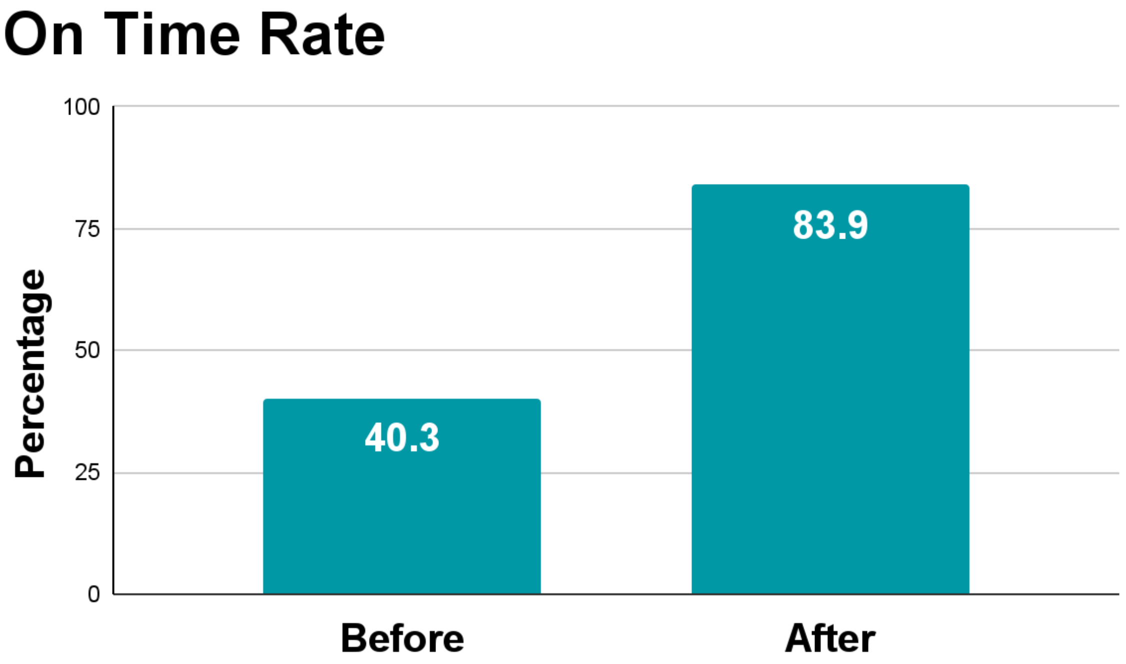
Top 5 Coefficient	
Features	**Coeff**
Discount	**8.03**
Importance	**0.144**
Rating	**0.096**
Warehouse_D	**0.05**
Gender_M	**0.04**

Top 5 coefficients show **direct relationship** to the target ‘Late’.

underscore

Recommendation

On Time Rate



On-time rate potentially increase by 108% from the previous 40.3% to become 83.9% after action based on predictive modeling

underscore

underscore

Recommendation

Potential Revenue Loss Saved

$ 196.8

Avg Revenue
Per customer

$ 1.291.729,66

Potential Revenue Loss

$ 942.964,62

73%

Potential Revenue Loss Saved

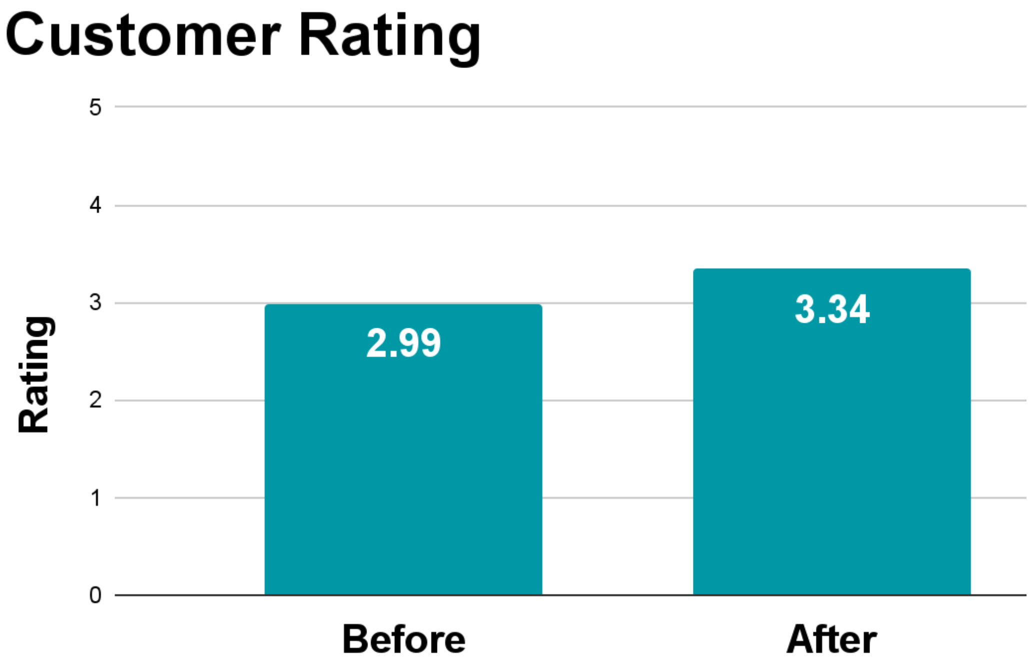
If the late customers stops shopping then the potential revenue loss for the company is approx $ 1.3 million.

But with predictive model, company can potentially save $ 942 thousand.

underscore

Recommendation

Customers Satisfaction



Our model gives an **11.7% increase** in customer rating.

It is proven with the previous average rating score of **2.99 becomes 3.34**.

That can be increased by adding 1 star to the predicted 'Late' except for customers who have given 5 stars since 5 is the maximum value can be given.

Recommendations

Short Terms

Add Estimated Package Arrived Time

Add estimated arrival time to assure the package arrived on time

Credit Points

Give credit points as a compensations to retain customer loyalty

Long Terms

More Features

Add more features to give more specific insights

Operational Audit

Perform operational audit based on the insights

THANK YOU!

Appendix
Confusion Matrix

	Predicted Class POSITIVE (Late)	Predicted Class NEGATIVE (On Time)
Actual Class POSITIVE (Late)	**TRUE POSITIVE (TP)** Late Predicted Late	**FALSE NEGATIVE (FN)** Late Predicted On Time
Actual Class NEGATIVE (On Time)	**FALSE POSITIVE (FP)** On Time Predicted Late	**TRUE NEGATIVE (TN)** On Time Predicted On Time

Primary : Recall (True Positive Rate)
Secondary : Average Precision

Appendix

Model

underscore

```python
#Splitting Feature & Target
xlr4 = df_project.drop(columns = ['Late']) #feature
ylr4 = df_project['Late'] #target
```

```python
#Splitting data Train & data Test
from sklearn.model_selection import train_test_split
xlrtrain4, xlrtest4, ylrtrain4, ylrtest4 = train_test_split
(xlr4, ylr4, test_size = 0.3, random_state = 33)
```

```python
from sklearn.linear_model import LogisticRegression
modelLR4 = LogisticRegression(random_state=33)
modelLR4.fit(xlrtrain4, ylrtrain4)
```

```
LogisticRegression(random_state=33)
```

```python
y_pred_trainLR4 = modelLR4.predict(xlrtrain4)
y_pred_trainLR4
```

```
array([0., 1., 1., ..., 0., 1., 1.])
```

```python
y_predLR4 = modelLR4.predict(xlrtest4)
y_predLR4
```

```
array([0., 1., 1., ..., 1., 0., 1.])
```

```python
modelLR4.predict_proba(xlrtest4)
```

```
array([[0.73636491, 0.26363509],
       [0.06400512, 0.93599488],
       [0.39181156, 0.60818844],
       ...,
       [0.07131088, 0.92868912],
       [0.61616682, 0.38383318],
       [0.4189624 , 0.5810376 ]])
```

```python
model_evaluation(modelLR4, y_predLR4, xlrtrain4, ylrtrain4, xlrtest4, ylrtest4)
```

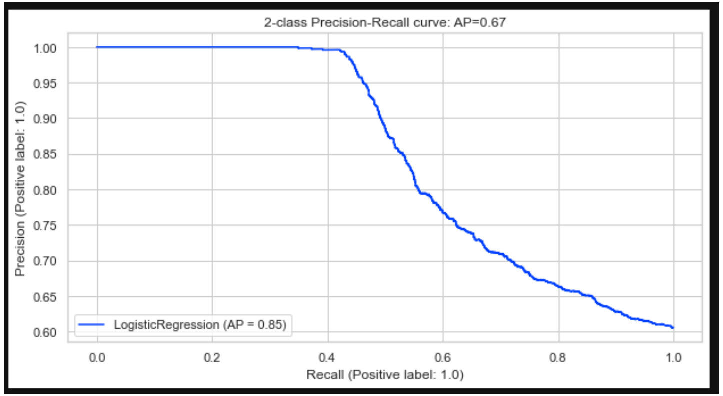
```
Accuracy : 0.640
Precision : 0.694
Recall : 0.726
F-1Score : 0.710
ROC AUC : 0.617
AP : 0.670
```

```python
print('AP test score : ',average_precision_score(ylrtest4, y_predLR4))
print('AP train score : ',average_precision_score(ylrtrain4, y_pred_trainLR4))
```

```
AP test score :  0.6697762992800597
AP train score :  0.6653809655739695
```

```python
print('train Accuracy : ',modelLR4.score(xlrtrain4, ylrtrain4))
print('test Accuracy : ',modelLR4.score(xlrtest4, ylrtest4))
```

```
train Accuracy :  0.640988995283693
test Accuracy :  0.640213404468156
```



underscore

Appendix

Coefficient Logistic Regression

```
print(modelLR4.intercept_)
```

```
[1.23464873]
```

```
print(modelLR4.coef_)
```

```
[[-0.71546752  0.09661052 -0.31429438 -2.20297555  0.1440757   8.03198014
  -2.40729906 -0.0680354   0.02322063  0.03060397  0.05114368 -0.0352052
   0.01004352 -0.03501893  0.02670308 -0.04108585  0.04281352]]
```

	Feature	Coefficient
0	Discount	8.031980
1	Importance	0.144076
2	Rating	0.096611
3	Warehouse_D	0.051144
4	Gender_M	0.042814
5	Warehouse_C	0.030604
6	Shipment_Ship	0.026703
7	Warehouse_B	0.023221
8	Shipment_Flight	0.010044
9	Shipment_Road	-0.035019
10	Warehouse_F	-0.035205
11	Gender_F	-0.041086
12	Warehouse_A	-0.068035
13	Cost	-0.314294
14	Calls	-0.715468
15	Purchase	-2.202976
16	Weight	-2.407299

Appendix

On Time Rate Growth Calculation

EXISTING	#	%
Delivery	10.999	100%
Late	6.563	59.7%
On Time	4.436	40.3%

AFTER MODEL PREDICTION	var	#	%
Delivery	a	10.999	100%
Late	b	6.563	59.7%
Predicted Late	c	4.791	73%
Predicted on Time	d	1.772	27%
Late After Prediction	e (b-c)	1.772	16.11%
On Time	f	4.436	40.3%
On Time After Prediction	g (f+c)	9.227	83.89%
On Time Growth Rate	4.436 to 9227 = 108%		

Appendix

Potential Revenue Loss Saved Calculation

	Delivery a	Cost b	Discount c	Revenue d (b - c)	Avg Revenue e (d / a)
Delivery	10.999	Cost Feature	Discount Feature	Revenue	$196.8

	Delivery a	Avg Revenue b	Potential Revenue c (a * b)	% b
Late	6.563	$196.8	$ 1.291.729,66	100%
Predicted On Time	1.772		$ 348.765,04	27%
Predicted Late	4.791		$ 942.964,62	73%

underscore

Appendix

Rating Growth Calculation

	Delivery a	Rating b	Avg Rate c (b / a)
Delivery	10.999	32.893	2.99
Predicted Late *Customers potentially increase their Rating by 1 (except if the customer already gave Rating = 5)*	4.791 - 20% = 3833	32.893 + 3833 = 36.726	3.34
Rating Growth Rate	2.99 to 3.34 = 11.7%		

	Late	Rate 5
Customers	6.563	1.317 (20%)

20% from all Late customers give 5 rating

underscore