Happy Customers

Background:

We are one of the fastest growing startups in the logistics and delivery domain. We work with several partners and make on-demand delivery to our customers. During the COVID-19 pandemic, we are facing several different challenges and everyday we are trying to address these challenges.

We thrive on making our customers happy. As a growing startup, with a global expansion strategy we know that we need to make our customers happy and the only way to do that is to measure how happy each customer is. If we can predict what makes our customers happy or unhappy, we can then take necessary actions.

Getting feedback from customers is not easy either, but we do our best to get constant feedback from our customers. This is a crucial function to improve our operations across all levels.

We recently did a survey to a select customer cohort. You are presented with a subset of this data. We will be using the remaining data as a private test set.

Data Description:

Y = target attribute (Y) with values indicating 0 (unhappy) and 1 (happy) customers
X1 = my order was delivered on time
X2 = contents of my order was as I expected
X3 = I ordered everything I wanted to order
X4 = I paid a good price for my order
X5 = I am satisfied with my courier
X6 = the app makes ordering easy for me

Attributes X1 to X6 indicate the responses for each question and have values from 1 to 5 where the smaller number indicates less and the higher number indicates more towards the answer.

Goal(s):

Predict if a customer is happy or not based on the answers they give to questions asked.
Success Metrics:

Reach 73% accuracy score or above

Bonus(es):

We are very interested in finding which questions/features are more important when predicting a customer's happiness. Using a feature selection approach show us understand what is the minimal set of attributes/features that would preserve the most information about the problem while increasing predictability of the data we have. Is there any question that we can remove in our next survey?

Data Analysis:

	Happy/Unhappy	On Time	As Expected	Ordered Everything	Good Price	Satisfied with courier	Easy App
0	0	3	3	3	4	2	4
1	0	3	2	3	5	4	3
2	1	5	3	3	3	3	5
3	0	5	4	3	3	3	5
4	0	5	4	3	3	3	5
5	1	5	5	3	5	5	5
6	0	3	1	2	2	1	3
7	1	5	4	4	4	4	5
8	0	4	1	4	4	4	4
9	0	4	4	4	2	5	5

Original Data

	Happy/Unhappy	On Time	As Expected	Ordered Everything	Good Price	Satisfied with courier	Easy App	Satisfaction
0	0	3	3	3	4	2	4	3.166667
1	0	3	2	3	5	4	3	3.333333
2	1	5	3	3	3	3	5	3.666667
3	0	5	4	3	3	3	5	3.833333
4	0	5	4	3	3	3	5	3.833333
5	1	5	5	3	5	5	5	4.666667
6	0	3	1	2	2	1	3	2.000000
7	1	5	4	4	4	4	5	4.333333
8	0	4	1	4	4	4	4	3.500000
9	0	4	4	4	2	5	5	4.000000

Data After Adding New Feature (Over All Satisfaction)

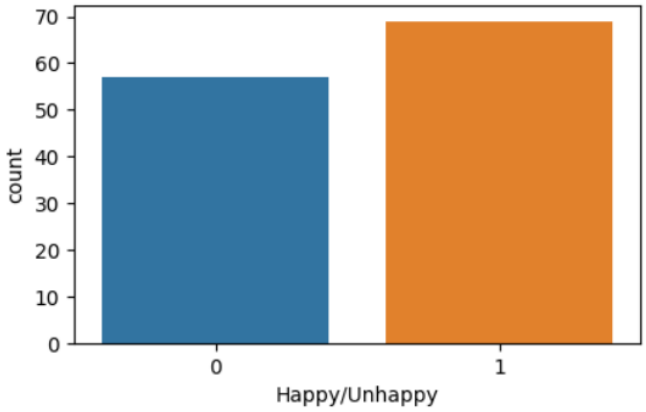
- **Statistics of the data set**

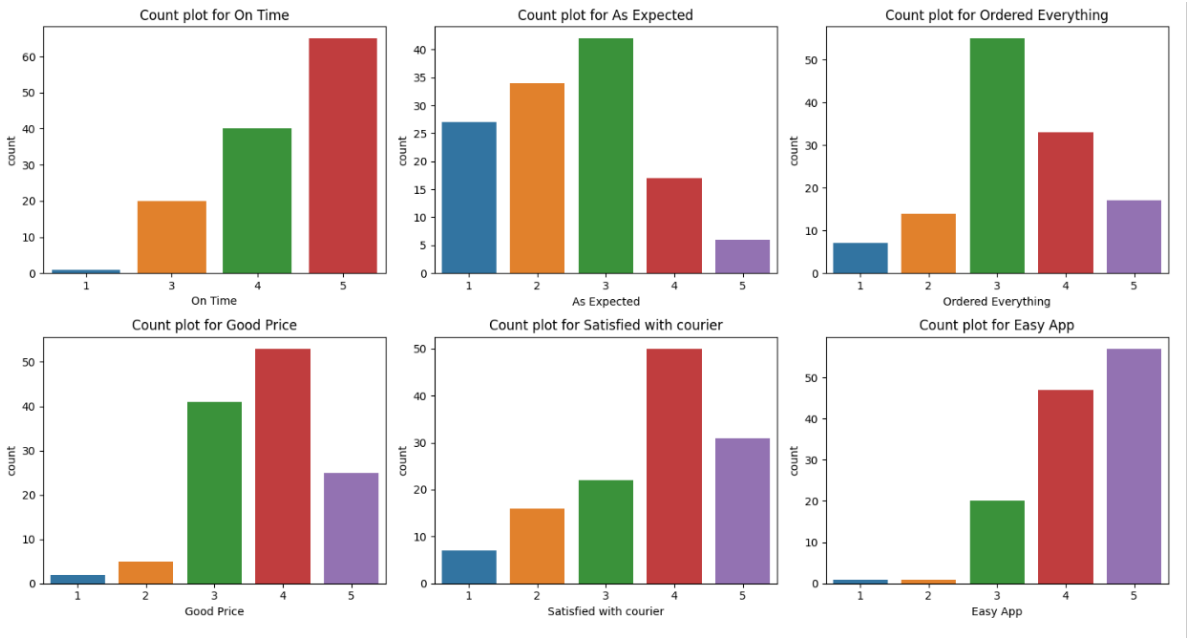
	Happy/Unhappy	On Time	As Expected	Ordered Everything	Good Price	Satisfied with courier	Easy App
count	126.000000	126.000000	126.000000	126.000000	126.000000	126.000000	126.000000
mean	0.547619	4.333333	2.531746	3.309524	3.746032	3.650794	4.253968
std	0.499714	0.800000	1.114892	1.023440	0.875776	1.147641	0.809311
min	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	0.000000	4.000000	2.000000	3.000000	3.000000	3.000000	4.000000
50%	1.000000	5.000000	3.000000	3.000000	4.000000	4.000000	4.000000
75%	1.000000	5.000000	3.000000	4.000000	4.000000	4.000000	5.000000
max	1.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000

Data Visualization:

Figure Show the total number of happy and unhappy customers.

The dataset has more happy customer but the difference is not big to effect the trained model and have biased toward happy customers





- Getting orders “On Time” and having an “Easy App” shows more satisfaction from customers.
- Getting orders “As Expected” , having a “Good Price”, being :Satisfied with courier”, and “Ordering Everything” shows that customer satisfaction has a different distribution and decline after the rating of (3) or (4) in the survey; thus more analysis/improvement is needed on these.

Model & Coclusion:

To start building the classification model, I choose 5 initial models to train using the default parameters, and the result was as follow:

- **Support vector machine (using linear kernel):** Accuracy = 78.9% and F1 score = 0.8
- **Logistic Regression:** Accuracy = 57% and F1 score = 0.63
- **Descension Tree:** Accuracy = 63% and F1 score = 0.66
- **Random Forest:** Accuracy = 57.8% and F1 score = 0.63
- **K nearest neighbour :** Accuracy = 36.8% and F1 score = 0.45

To improve the models performance and tune the parameters of each model, grid search is implemented and the result as follow:

- **Support vector machine (using linear kernel):** Accuracy = 78.9% and F1 score = 0.8
- **Logistic Regression:** Accuracy = 57% and F1 score = 0.63
- **Descension Tree:** Accuracy = 63% and F1 score = 0.66
- **Random Forest:** Accuracy = 52.6% and F1 score = 0.57
- **K nearest neighbour :** Accuracy = 47% and F1 score = 0.54

Concluded that support vector machine produced the best result with an accuracy of 79% and an F1 score of 0.8 .