PROJECT REPORT
Talha Naeem

Explanation of Problem Statement

In this project, the aim was to predict whether the customer is happy or unhappy based the star reviews they had written over a series of questions which are shown below:

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

Figure 1

We were given the datasets in the form of a CSV file, in which the first column was Y, second column was X1, third was X2 and so on. Below is the figure that shows first 50 entries of how the datasets is given to us:

	A	B	C	D	E	F	G
1	Y	X1	X2	X3	X4	X5	X6
2	0	3	3	3	4	2	4
3	0	3	2	3	5	4	3
4	1	5	3	3	3	3	5
5	0	5	4	3	3	3	5
6	0	5	4	3	3	3	5
7	1	5	5	3	5	5	5
8	0	3	1	2	2	1	3
9	1	5	4	4	4	4	5
10	0	4	1	4	4	4	4
11	0	4	4	4	2	5	5
12	0	3	2	3	3	2	3
13	0	4	4	3	4	4	4
14	1	5	2	4	5	5	5
15	0	4	2	4	5	4	3
16	0	4	1	3	3	4	3
17	1	3	2	4	3	4	4
18	0	5	3	4	5	4	5
19	1	5	1	4	3	4	5
20	0	5	1	2	4	4	5
21	0	4	2	4	4	4	4
22	1	4	2	3	3	4	4
23	0	4	3	5	5	5	4
24	0	4	3	5	5	5	4
25	1	5	1	2	5	2	4

Figure 2

The total number of entries are 127, but the first row is the header, that only leaves us with 126 rows (customer reviews) of actual content. As mentioned before first column is for Y which is our target attribute indicating if customer is happy or not, second column is X1, third one is X2 and so on. What X1, X2, X3, X4, X5, and X6 represent is depicted in **Figure 1.**

The values taken by the first column is binary that is it can be either 0 (unhappy) or a 1 (happy). The rest of the columns take any integer value from 1 to 5, with 1 indicated that the customer is least satisfied with the corresponding field of the question, and 5 indicating the customer was highly satisfied with the corresponding field of the question. The level of satisfaction increases as we move from 1 to 5.

What each star indicates is pretty much standard in the entire world ans it is indicated by the following figure:

Figure 3

Coding Part

The coding was done in python, and the necessary libraries required to implement this project were pandas and certain libraries from sklearn. Refer to the coding file main.py for more information on this

Figure 3 is important because in the code I made a threshold in which I declared a rating of above 3 is considered happy for the customer in the respective field of question and any thing below it is regarded as unhappy for that customer in that respective question, e.g. customer 105^{th} gave 3 star for X1: the order was delivered on time, we can say he is happy, and gave 1 star for X4: I paid good price for the order, we can say he is unhappy. This is because I have decided to implement this project using Logistic Regression Model, in which it is vital to ‘binarize’ the dataset, that is to give it a value of 0 or a 1. Since this ‘binarization’ seem plausible, we have implemented this in our coding section whose screen shot is in the figure below:

```
#Givingthreshold that rating > 3 in each question is happy for that question
dummyX1 = my_data['X1'] >= 3
dummyX2 = my_data['X2'] >= 3
dummyX3 = my_data['X3'] >= 3
dummyX4 = my_data['X4'] >= 3
dummyX5 = my_data['X5'] >= 3
dummyX6 = my_data['X6'] >= 3
```

Figure 4

The above figure illustrates how 'binarization' was performed while coding. In my coding part I also analyzed which independent variables were not important. The was I achieved that is by removing some unnecessary independent variables (X1, X2, X3, and X6) from training variable x_trainUnaff arbitrarily using trail and error. While X4 and X5 seemed to make the impact on results. The results were compared with the original x_train in which included all independent variable and none was removed. Since I got the exact same result when some unnecessary independent variables were removed, it was established that they had no impact on results. Following figure represents the coding section of this part where I removed the independent variables that had no impact on output:

```
#Which independent variables are irrelavent
x_unaffected = my_data.drop(['Y', 'X1', 'X2', 'X3', 'X6'], axis = 1)
x_trainUnaff, x_testUnaff, y_trainUnaff, y_testUnaff = train_test_split(
                                                    x_unaffected, y,
                                                    test_size = 0.10,
                                                    random_state = 7)
logModelUnaff = LogisticRegression()
logModelUnaff.fit(x_trainUnaff, y_trainUnaff)

predictionsUnaff = logModelUnaff.predict(x_testUnaff)
resultsUnaff = classification_report(y_testUnaff, predictionsUnaff)
confMatUnaff = confusion_matrix(y_testUnaff, predictionsUnaff)
accScoreUnaff = accuracy_score(y_testUnaff, predictionsUnaff) * 100


print('The confusion matrix for relavent independent variables is \n'
      , confMatUnaff, ' and accuracy is:', accScoreUnaff, '\n')
print('The results are unaltered even if we remove X1, X2, X3, X6.'
      'Check Variable explorer for more information.Therefore, results '
      'are not dependent on them, so they should be removed.')
```

Figure 5

It can be checked by adding X4 and/or X5 on the first line of code on figure 5 on main.py to see that X4 and X5 do impact the results while rest do not.

If following line gives an error:

```
from sklearn.model_selection import train_test_split
```

replace the above line with:

```
from sklearn.cross_validation import train_test_split
```

Accuracy and Results

Following are the outputs that you should see at output terminal:

```
Python 3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Talha Naeem/Desktop/project/main.py', wdir='C:/Users/Talha Naeem/
Desktop/project')
There were no missing values in our dataset.

The confusion matrix for all independent variables is
 [[1 3]
 [1 8]]  and accuracy is  69.23076923076923

The confusion matrix for relavent independent variables is
 [[1 3]
 [1 8]]  and accuracy is: 69.23076923076923

The results are unaltered even if we remove X1, X2, X3, X6.Check Variable explorer for more
information.Therefore, results are not dependent on them, so they should be removed.
```

Now the accuracy that I achieved was 69.23 % which is a little shy of the target 73 %. This is the most accurate model that I could design that made most sense to me. Although there may be more efficient means other than Logistic Regression like NLP that could predict the customer satisfaction with a much higher precision, but it is obvious that my model is very much intuitive than any other model like NLP. So I believe that my model pretty much covers the necessities needed to predict customer happiness using simple machine learning library and basic coding that does not need high level of understanding, but basic intuitive algorithm to kick train a basic model. I made use of simple logic to convert my ratings into binary data, and used it for training. I also chose my random_state as 7 arbitrarily using trail and error, as it gave me highest accuracy. Also I chose 10 % test size as our dataset is small and we should use more data for training, and less for testing. Using the above reason I believe that my model gives reasonable results. I hope all of this serves the purpose to convince that this model is great model to predict customer happiness.