Supervised Learning: Classification

IBM Machine Learning - Project
Mohammed Qasim K.
October 2022

Main Objective

- The main objective of this analysis was to classify the fictional users in the data set as likely to spend money or not using Classification Methods
- The data set was split into Training set (60%), Validation set (20%), and Test set (20%) using Train-Test-Split.

About the Data

- The data set comprises of fictional user behaviors on the Stranger social app platform.
- This data set has 3000000 records and 5 variables. During the analysis, 7 duplicates were detected and removed.

https://www.kaggle.com/datasets/sexiaoze/stranger-social-app-user-behaviour-data?select=user_id.csv

Variable name	Type	Description
user_id	int	Unique identifier for a given user
ts	object	Time Stamp of the given behavior
behaviour_type	object	A particular behavior of the app (like, disklike, etc.)
consume	int	Amount of money that occurred for those records
target_user	int	Unique identifier of the receptor *user_id*

Data Exploration

- After checking for duplicates, the EDA was conducted on the data set.
- The data description is as follows:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2999993 entries, 0 to 2999992
Data columns (total 5 columns):
 #   Column          Dtype
---  ------          -----
 0   user_id         int64
 1   ts              object
 2   behaviour_type  object
 3   consume         int64
 4   target_user     int64
dtypes: int64(3), object(2)
memory usage: 114.4+ MB
```

Data Exploration

- EDA was conducted on the *behaviour_type* column.
- The data description is as follows:

```
view          563888
dislike       448510
voiceroom     416270
like          413939
click         318675
zan           313339
post          229844
comment       197854
follow         62642
buy            35032
Name: behaviour_type, dtype: int64
```

behaviour_type	Description
dislike	the user doesn't want to match with the target user
like	the user feels ok to match the target user
zan	a like given by the user to any post
view	browse for recommended items, ads, posts, and any things you could imagine here with staying for at least a few seconds
post	put up a post
voiceroom	the user enters a voice chatting room
click	click a banner or an ad
buy	spend money on something
comment	comment a post
follow	follow a topic, a user, a group or something else

Data Exploration

- One Hot Encoded the *behaviour_type* column and summed the column based on the *user_id.*
- Dropped *ts*, *target_user* and *consume* columns.
- Encoded the buy column into a Binary decision.

0 : Not Likely to perform In app purchases
1: Likely to perform In app purchases

	user_id	buy	click	comment	dislike	follow	like	post	view	voiceroom	zan
0	10928793	0	0	0	1	0	0	0	0	0	0
1	32521526	0	0	0	0	0	0	0	1	0	0
2	63353958	0	0	0	0	0	0	0	0	1	0
3	84233255	0	0	0	0	0	0	0	1	0	0
4	98010797	0	0	0	0	0	0	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...
2999988	17775246	0	0	0	0	0	0	0	1	0	0
2999989	69288974	0	0	0	0	0	0	1	0	0	0
2999990	69497756	0	0	0	0	0	1	0	0	0	0
2999991	60503809	0	0	0	0	0	0	0	0	1	0
2999992	22359624	0	0	0	0	0	0	0	0	0	1

Data Exploration

- The new data set upon which analysis will be conducted has 15000 unique records and 11 variables.
- The target label is the *buy* column where:

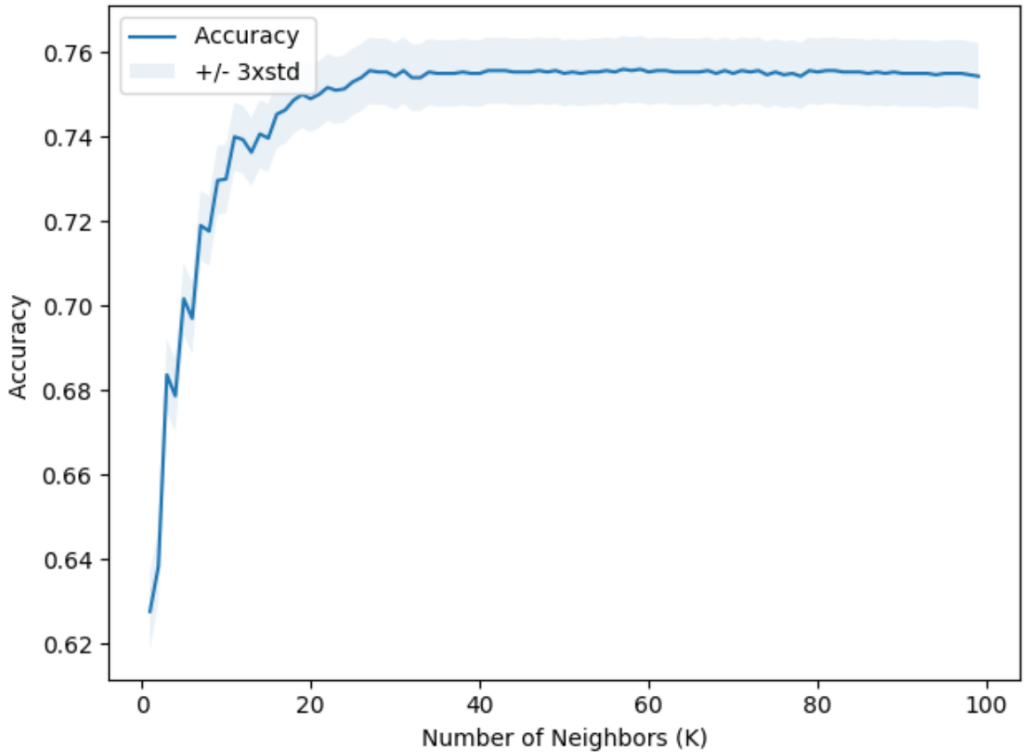
0 : Not Likely to perform In app purchases

1: Likely to perform In app purchases

user_id	buy	click	comment	dislike	follow	like	post	view	voiceroom	zan
10006437	1	11	7	26	0	9	9	24	15	25
10013978	1	13	8	21	0	19	9	32	17	21
10015055	1	18	20	29	5	33	17	37	30	21
10022355	1	23	16	33	0	37	15	39	21	16
10024335	0	22	21	29	1	24	19	34	22	26
...	...	...	...	...	...	...	...	...	...	...
99959540	1	19	5	37	3	27	12	42	21	24
99967539	1	13	15	20	0	19	13	21	19	10
99974265	1	21	15	37	3	27	15	46	29	21
99975758	1	13	1	16	0	15	12	24	21	14
99997216	1	17	11	22	0	18	7	22	26	13

Model Variations

- Implemented a K Nearest Neighbor algorithm, with a range of different K values (0 – 100) for finding the appropriate number of neighbors for clustering the dataset.




```
The best accuracy was with 0.756 with k= 57
```

Predictions on the Validation Set

The four models were then fit on the unseen Validation Set and the appropriate evaluation metrics were calculated for each model.

- K Nearest Neighbor with *n_neighbors* = 57
- Decision Tree with an *Entropy* criterion
- Support Vector Machine with a *Sigmoid* kernel
- Logistic Regression with C = 0.01

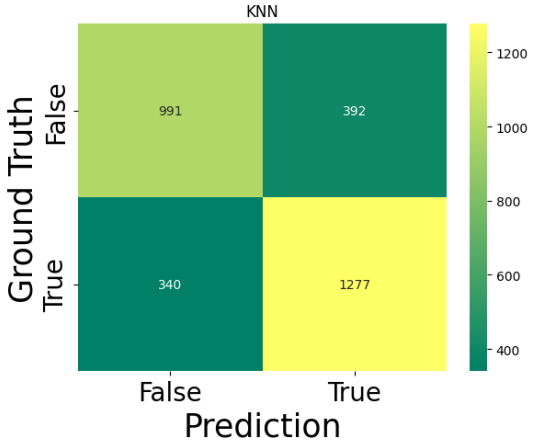
Comparing the Metrics

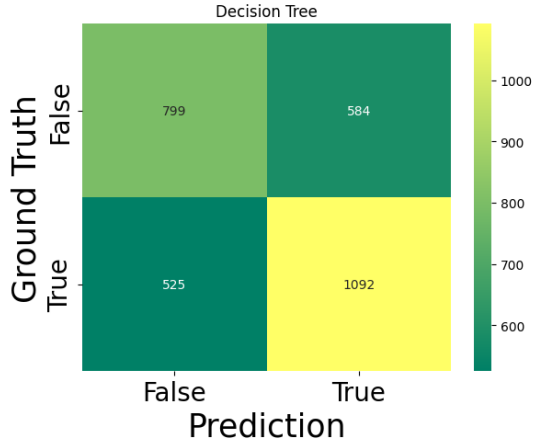
- The metrics show that KNN and Logistic Regression performed decently with no signs of overfitting.
- SVM performed the worst on the validation set and is probably not fit for the data set.
- The best classification model is KNN (Jaccard score = 0.635640 & F1-score = 0.755593)

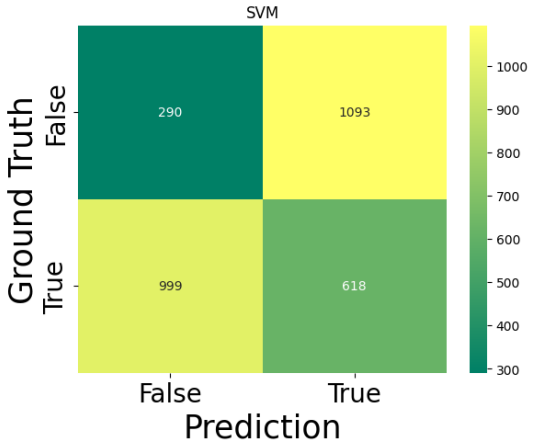
Algorithm	Jaccard	F1-score	LogLoss
KNN	0.635640	0.755593	NA
Decision Tree	0.500683	0.633944	NA
SVM	0.228044	0.300249	NA
Logistic Regression	0.622556	0.748980	0.557872

Comparing the Metrics

- The confusion matrix shows that KNN and Logistic Regression performed decently with less False Positives & Negatives.
- SVM performed the worst on the validation set.
- The best classification model is KNN (True Positives= 0.79 & True Negatives = 0.72)







Predictions on the Test Set

The four models were then fit on the unseen test Set and the appropriate evaluation metrics were calculated for each model.

- K Nearest Neighbor with *n_neighbors* = 57
- Decision Tree with an *Entropy* criterion
- Support Vector Machine with a *Sigmoid* kernel
- Logistic Regression with C = 0.01

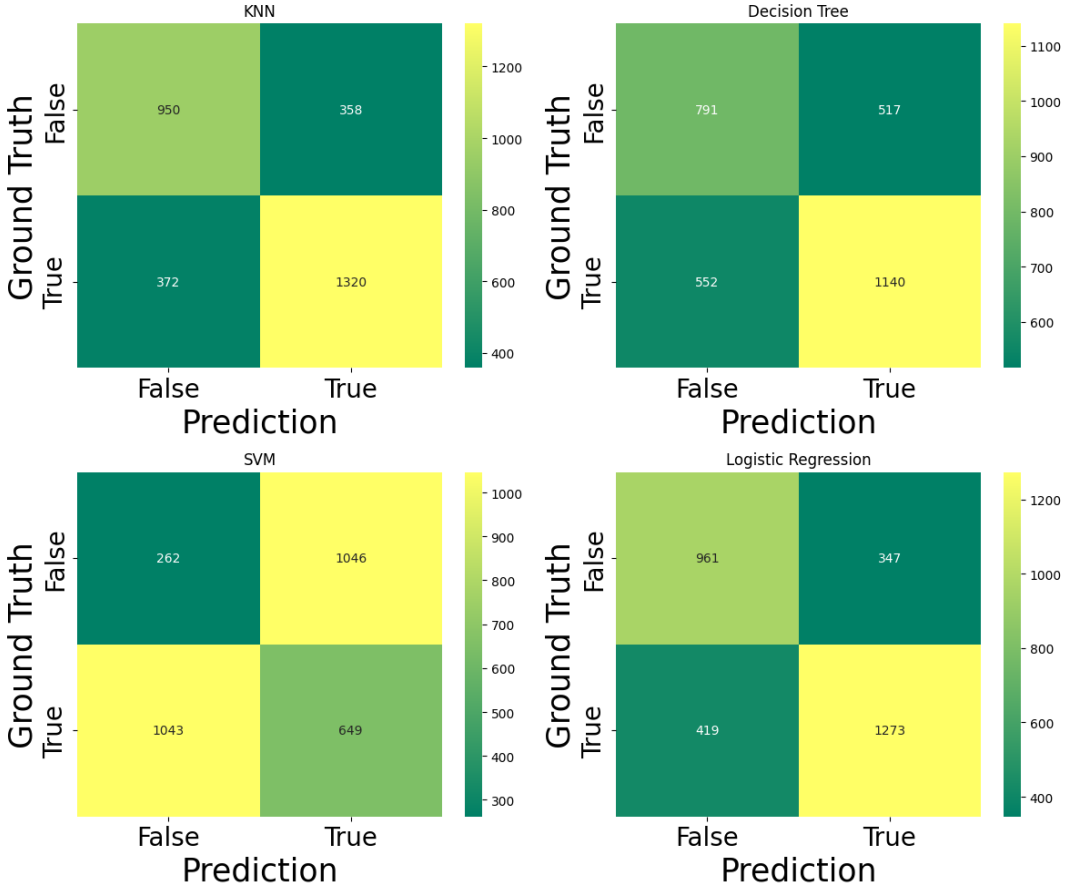
Comparing the Metrics

- The metrics show that KNN and Logistic Regression performed decently with no signs of overfitting.
- SVM performed the worst of the test set and is probably not fit for the data set.
- The best classification model is KNN (Jaccard score = 0.643902 & F1-score = 0.756809)

Algorithm	Jaccard	F1-score	LogLoss
KNN	0.643902	0.756809	NA
Decision Tree	0.510840	0.639547	NA
SVM	0.237034	0.303575	NA
Logistic Regression	0.624326	0.745311	0.559229

Comparing the Metrics

- The confusion matrix shows that KNN and Logistic Regression performed decently with less False Positives & Negatives.
- SVM performed the worst on the test set.
- The best classification model is KNN (True Positives= 0.78 & True Negatives = 0.73)



Conclusion

The analysis shows that Feature Engineering has a large effect on the model performance. The best The analysis shows that Feature Engineering has a large effect on the model performance. The best classification model for this application is K Nearest Neighbors (KNN) . Although performance by the model wasn't great improvements can be made by choosing a dataset which is not synthetic, trying other classification methods not employed in this analysis or adding/generating more features which wasn't done for the sake of simplicity and limited computational resources.

Jupyter Notebook can be found here: https://github.com/moqa19/IBM-Machine-Learning/blob/main/ML%203.ipynb