Data Analyst Take Home Test

August 20 2021

Pranshu Kumar Premi

Summary

The main goals of the assignment are –

1. find and specify a target group for the promotion
2. Build a machine learning model in Python for predicting user's probability of conversion.

The analysis involved Exploratory Data Analysis of all the datasets provided for deriving insights.
It was found that the users bought products in form of gems, passes and value packs.
The spendevents data helped find out that majority of users fell in 'earnGemsCounter' category, followed by 'premiumChoice' and lastly 'IAP'.

Another 'spendtype' category named 'value pack' was found, however, entries with this category only had negative amount that affected the company bank account.
The 'users' dataset helped find out US, UK, Canada, Australia, and Philippines as top 5 countries with maximum users, and iPhone7, iPhone8, and iPhone6 as top 3 phone models used for gaming.

Basically, all datasets were explored for checking datatypes, null values if any, data dimensions, and variables' statistics summary.

For the first problem statement, we targeted the users with spendType as 'earnGemsCounter','IAP', and 'valuepack' who are making gems flow out of company's bank account. The users that selected 'premiumChoice' will not be considered as they already are contributing towards positive amount.

For the second problem statement, the dataset 'spendevents.csv' labelled as 'gem3' seemed a decent dataset to be used for training the machine learning model. we are basically dealing with a classification problem where we analyze whether the target group will end up spending gems or not i.e., falling in the 'premiumChoice' spendtype category or not.

We selected the 'spendType' feature as the target variable.
The 'spendType' variables has basically 4 classes, and we were trying to predict the probability of user falling into the 'premiumChoice' category in the future.

We could have considered this as a situation for multiclass classification, but we are more interested in finding people that may end up purchasing gems using their own account, and the all-other classes except 'premiumChoice' do not contribute positively to the 'amount' feature, so we considered all those classes as class **' 0 '** and 'premiumChoice' as class **' 1 '** for classification.

Data preprocessing included converting relevant date features to correct datatypes format, feature engineering for reducing feature cardinality, dropping non-required features and Label Encoding categorical features.

We split the data into 80% as train data and 20% as test data.
We utilized StandardScaler() function to rescale all features for a relatively normal distribution of values.

We selected Logistic Regression as our initial Classfication Machine Learning algorithm, achieved a prediction score of around 87%
We utilized cross-validation technique for assessing model effectiveness, but found the mean accuracy score close to the original model accuracy. We utilized the predict_proba() function from Scikit-Learn to get prediction probabilities.

We also made initial mode's performance comparisons with other classification algorithms including K-Nearest Neighbors and Naïve Bayes classifier.
We evaluated model results based on metrics including accuracy score, precision, recall, F1 score, F2 score.

Below contains the tabular representation of model names with respective performance metrics –

index	Model	Accuracy	Precision	Recall	F1 Score	F2 Score
0	K-Nearest Neighbours	0.906503	0.708141	0.493400	0.581581	0.525256
1	Naive Bayes	0.906503	0.708141	0.493400	0.581581	0.525256
2	Logistic Regression	0.871547	0.549356	0.136996	0.219303	0.161196