Capstone Proposal: Starbucks Project

Domain Background

The Starbucks project contains simulated data which replicates consumer behavior on the Starbucks app. The data provided includes transaction, demographic and offer data. The data itself only represents one product though in reality, Starbucks has many products. Businesses such as Starbucks which increasingly acquire, communicate with, and retain their consumers through virtual interactions are now in a position to learn from data factors which influence the behavior of their customers and better serve them to increase profits. Many business analysts use tools like SQL and Excel to analyze data sets and arrive at conclusions including heuristics for customer segmentation. Running classification algorithms might improve upon the applicability of heuristics at a granular level. For example, instead of a broad statement advocating showing a specific type of coupon to certain demographics, leveraging machine learning models might help us determine whether to show someone with a specific set of attributes a promotion based on their likelihood of viewing it and acting upon it.

As someone interested in the business applicability of machine learning techniques, I want to effectively leverage computational power and data to better serve customers and drive sales.

Problem Statement

Starbucks would like to glean insights from this data to better serve its customers and grow revenue. More specifically, it wishes to understand different segments of its customer base and how they react to offers so that it can better target them to increase spend. This is a classification problem predicting if an individual will make a purchase after having viewed a promotion based upon provided demographic data.

Datasets and Inputs

The data is contained in three JSON files [portfolio, transcript, and profile] respectively. The portfolio file contains information about the different types of offers. The profile dataset contains information on each customer such as their age, gender, income, and an id. The transcript file contains a record of all transactions and offer events. The Profile and transcript dataframes can be merged or "joined" at columns *profile.id* and *transcript.person*. The dataset was obtained from Starbucks which has provided simulated data to better personalize its reward offerings and drive business. The portfolio file will be used to understand the different types of offers and make some initial conjectures about what we believe we might find exploring the data. An interesting initial step might be to create several plots and see if new predictive features can be created by clustering the data. It is important to remember the data provided is in a relational format and combining it into one dataset means we will have to organize it further. For example, we will have to create a new categorical feature which we can populate with whether the person had viewed a promotion before making a transaction (and which type).

Solution Statement

To create "personas", I'll apply data exploration and Visualization methods to understand consumer behavior regarding coupon use.

To predict potential responses to a specific type of promotion, it will be useful to employ classification techniques. This can be used to determine the likelihood of someone viewing the coupon or even acting upon it (two different dependent vars for separate classification problems. The results from this step will be helpful to expand upon the consumer personas I've developed and see whether individual responses can be predicted based on the heuristics developed.

Benchmark Model

Traditionally in the business domain, segmenting groups of customers and targeting them differently works. Tools like SQL are very useful for analyzing and discovering insights about differences between different groups. I'll use classification to create a more granular and individualized approach built on the benchmark heuristics generated during data exploration phase. The results will be benchmarked against a random choice model as well the results of a persona based decision.

During an A/B testing phase, a company might show different groups different coupons as and compare treatment results to that of a control group. During such an RCT, there are certain assumptions underlying the different groups, namely that the random choice reduces bias and that the impact of a treatment can be judged from the results and statistical methods. Stratified random sampling takes this a step further by creating subpopulations with certain characteristics. To effectively compare performance of our individualized classifier against the personas, we'll have to examine how our classifier did with different subpopulations. For example, if one persona shows that 50% women with over $50000 in disposable income made a purchase after viewing offer A, we will take the assumption that the company expects this to be representative of future behavior and compare it to the actual results from the test data; that if all women are shown offer A, 50% of women making above $50000 will make a purchase and 50% will not.

Evaluation Metrics

For a regression model, RMSE is a desired metric to evaluate. For classification we might look at a confusion matrix and think about accuracy, precision, kappa stat, and recall.

$$\mathrm{RMSE}_{fo} = [\sum_{i=1}^{N} (z_{f_i} - z_{o_i})^2 / N]^{1/2}$$

Where:

- Σ = summation ("add up")
- $(z_{f_i} - Z_{oi})^2$ = differences, squared
- N = sample size.

RMSE:

One python implementation using sklearn would look as follows:

```
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse = sqrt(mean_squared_error(y_actual, y_predicted))
print(rmse)
```

Accuracy: Accuracy is the sum of true positives and true negatives divided by the sum of false positives and false negatives.

One implementation of accuracy in the sklearn library is accuracy score:

`sklearn.metrics.accuracy_score`(*y_true*, *y_pred*, *, *normalize=True*, *sample_weight=None*)

Conversely, I can define an accuracy function or calculate it inline. Sklearn metrics allows you to create a confusion matrix using *metrics.confusion_matrix()* and passing in the actual values, predicted values, and labels. Similarly to obtain a report which includes precision and recall, we can use *metrics.classification_report()*.

Precision can be calculated as the number of true positives divided by the sum of true positives and false positives.

Recall can be calculated as the number of true positives divided by the sum of true positives and false negatives (those which were marked negative but should have been classified as positive).

Cohen's kappa is a useful statistic to measure inter-rater agreement to correct for chance agreement and imbalanced classes which impact measures like accuracy.

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

Cohen's Kappa formula:

SKlearn allows you to calculate the kappa stat using `sklearn.metrics.cohen_kappa_score`.

A kappa stat close to zero signals little agreement and closer to one signifies almost perfect agreement.

Project Design Outline

For this project, I'll first explore the data and aim to see if there are any apparent patterns. I'll clean the data and based off initial exploration will make some changes during feature engineering. It might be helpful to join the datasets through and also create several data frames with different properties to work with during various steps for different problems after this stage. Using Data Exploration and Visualization methods, we'll come to understand the data better and stat to see patterns in consumer behavior. We'll split the data into train, validation, and test sets early on. As part of further exploration, we'll try to create representative customer segments based on specific factors such as income or age. We'll generate relevant insights on the behavior of different demographics. After we've created some profiles, we'll attempt to view relationships between different demographics and coupon use to determine which groups should be shown certain promotions. Note, the personas/heuristics developed will be built on information from train and validation data, but blind to the information from the test set.

Step 2: After we conduct our data analysis and write down our findings, we'll then create our heuristics with respective predictive statements and begin training a classification algorithm. I've considered XGBoost though will also look into other potential algorithms and methodologies to use for different steps such as potentially using auto-sklearn for hyperparameter tuning and algorithm selection. At the end, I'll compile a short executive summary for guidance along with a longer detailed project report.

Results can be compared against the findings from step 1 regarding our heuristics. To compare our heuristics and classifier, we'll compare the performance of the classifier on the test set (not validation) among certain subgroups described by heuristics to the predictive statement attached to the persona/heuristic: ex, 50% of women who make over $50000 will respond to offer A.

Step 3: Another classification problem to potentially pursue is determine is the likelihood of someone even viewing a promotion based on demographic data. We could even go a step further and create a regression problem to determine how influential viewing coupons is on total spend per customer, we could group by certain features or classes we create within features. Another interesting issue to look at is impact of promotions on customer lifetime value and return. Whether there is a strong relationship between promotions and recurring purchases to see if there are any interesting relationships there. These are potential area of interest to explore at the end and not the focus of this capstone.

References

https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/

https://intellipaat.com/community/1269/is-there-a-library-function-for-root-mean-square-error-rmse-in-python

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html

https://thedatascientist.com/performance-measures-cohens-kappa-statistic/

https://www.investopedia.com/ask/answers/032615/what-are-some-examples-stratified-random-sampling.asp

https://www.datarevenue.com/en-blog/use-ai-to-predict-who-to-target-offers-at