York B2E Capstone Report

Jenny Huang

February 20, 2024

1 Introduction

The objective of this model is to predict whether a Google Analytics hit will have an action of ”Add to Cart”. Using the Google Analytics BigQuery dataset, this model was developed and trained on 1 months’ worth of data, with the intention of being transferred to a years’ worth of data.

2 Data Explanation and Exploration

2.1 Dataset Explanation

This dataset was pulled from the BigQuery public data on Google Analytics, under the dataset name ‘bigquery-public-data.google_analytics_sample.ga_sessions_*‘. The dataset contains ”hits” stored in tables divided by day from August 2016 to August 2017 along with date, user, and traffic information. The columns ’eventinfo.eventAction’ and ’eCommerceAction.action_type’ contain information about the target feature, which is whether a hit is an ”Add to Cart” action. For proof of concept, data from October 2016 was used to train the model and data from March 2017 was used to validate and evaluate the model’s performance.

2.2 Data Preprocessing

Entries will null values were filtered out. Columns with ¿50% null values or only 1 unique value were also filtered out. Duplicate values and redundant columns were also removed.

Missing values were imputed with the mean (float), median (int), or mode (categorical).

2.3 Exploratory Data Analysis (EDA)

After filtering out entries with null target values, there were 119376 hits and 28 columns for October 2016. The following query was used to select this data:

```
SELECT date, CONCAT(fullVisitorId, visitId) AS unique_id, totals, trafficSource, device, geoNetwor
     CASE
     WHEN eventInfo.eventAction = ’Add to Cart’ OR eCommerceAction.action_type = ’3’ THEN 1
     ELSE 0
     END AS is_addtocart
     FROM ‘bigquery-public-data.google_analytics_sample.ga_sessions_*‘,
     UNNEST(hits) as hits LEFT JOIN UNNEST(product) as product
     LEFT JOIN UNNEST(promotion) AS promotion
     WHERE _TABLE_SUFFIX BETWEEN FORMAT_DATE(’%Y%m%d’, ’{year}-{month}-01’)
     AND FORMAT_DATE(’%Y%m%d’, ’{year}-{month}-{max_days}’)
     AND (eventInfo.eventAction IS NOT NULL OR eCommerceAction.action_type != ’0’)
     ORDER BY date, unique_id, hitNumber ASC;
```

The proportion of hits that are ”Add to Cart” is 7.6

The relationship of other metrics, such as path_depth and timeOnSite, to event action were also explored with notable correlation.

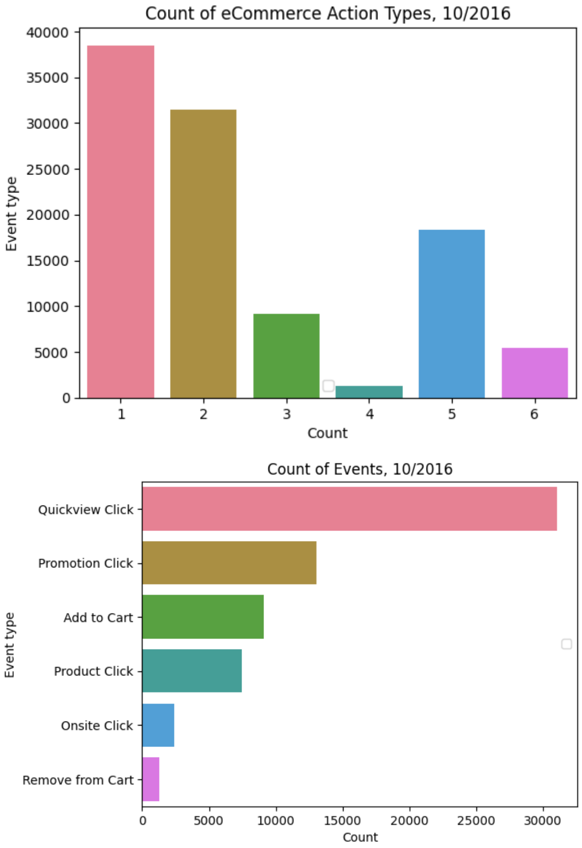


Figure 1: Histograms for the eventinfo.eventAction and eCommerceAction.action_type columns. 3 = Add to Cart

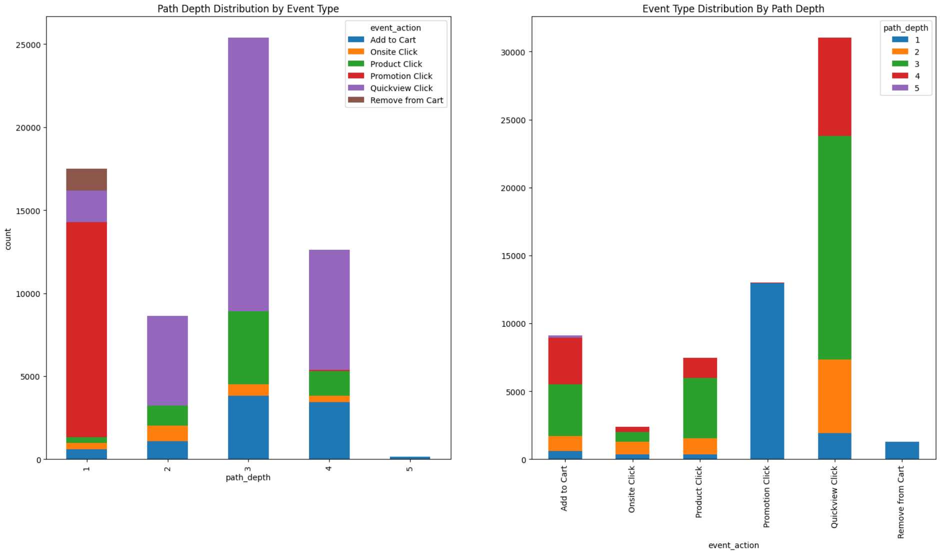
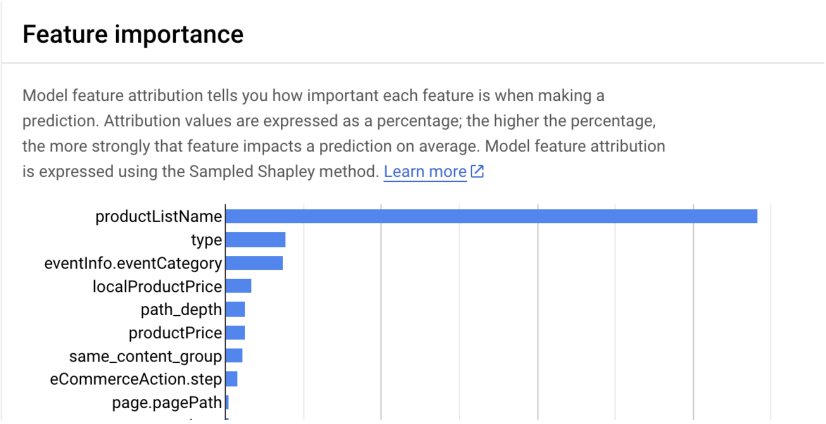


Figure 2: Path depth distributions by event type.


Feature importance

Model feature attribution tells you how important each feature is when making a prediction. Attribution values are expressed as a percentage; the higher the percentage, the more strongly that feature impacts a prediction on average. Model feature attribution is expressed using the Sampled Shapley method. Learn more

Figure 3: Feature importance based on initial AutoML training job.

2.4 Feature Selection and Engineering

Four features were engineered (see eda.ipynb and data_ingestion.ipynb), including a binary encoding of the target (is_addtocart) that combines the information from the 'eventinfo.eventAction' and 'eCommerceAction.action_type' features. A value of 1 is "Add to Cart", while a value of 0 is all other actions. is_addtocart was used as the target feature, and related features were dropped to prevent data leakage while training.

Using AutoML's feature evaluation, the 10 most importance features were used to train the final iteration of the model. (see Model Training and Figure /reffig:feat_import)

3).

These features were utilized in the final iteration of the model.

3 Model Training

3.1 AutoML

The model was implemented using Vertex AI's AutoML. A workbench instance was created where the model was created and trained (see model_training.ipynb). The cleaned datasets were uploaded to a Google Storage Bucket, where they could then be used to create an AutoMLTabularDataset.

The AutoMLTabularTrainingJob handles data transformation, splitting, model selection, hyperparameter tuning, and validation. The two model types tested were neural network (nn) and boosted gradient trees (boosted tree).

A training/testing/validation split of 80/10/10% was used on the Oct 2016 data. The optimization metric was ROC AUC. This metric measures the area under the curve between the true positive rate against the false positive rate; the greater the area, the better the performance.

After training, the model was deployed through an online prediction endpoint for further external validation.

3.2 Model Performance

The model trained with the 10 best features achieved perfect accuracy on the testing set of 10636 samples. This is a signal of overfitting; however, the model also performed similarly on unseen data from march 2017 of size 35149 rows also achieved near-perfect performance, only misclassifying 1 entry.

4 Cost Analysis

This model was developed and deployed with Vertex AI and utilized BigQuery and Jupyter Notebooks.

Item	**Cost ($)**
Vertex AI Instance	13.24
Node hours	42.84
BigQuery	3.01
Notebooks	8.24
Total	**67.33**

Table 1: Costs Table

5 Conclusion

This model was able to successfully predict when a hit will have an action of ”Add to Cart”. However, the unusually high performance rate indicates this model may be overfit, although the model still performed well on unseen data. Further validation with the whole year’s dataset or batch predictions are necessary to fully determine whether the model is overfit.

The cost of the model could be optimized by stopping the Vertex AI instance more often when idle, and by minimizing the number of trainings and deployments, which appear to be the most cost-intensive.