FourthBrain

Team GroupBy

Toyosi Bamidele , Uchenna Mgbaja

Outline

- Problem
- Solution
- Data + Model
- Demo
- MLE Stack
- Conclusions (and lessons learned)
- Future Work

Problem

Understanding customer behavior in the e-commerce space, a business area altered during the pandemic due to increased demand for online purchases, improving the customer experience, to ensure customer retention and product monetization is critical. The main goal is optimizing the customer journey and shopping experience using a predictive model and recommendation system

Solution

1.Predictive Model:Uplift model(One and Two model approach)

- What customers are likely to convert?
- Who should we target primarily?

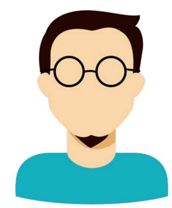
2.Recommendation System

- What products should we recommend to our users based on their purchase history?
- What products should we recommend to users based on items pairs from past basket purchases

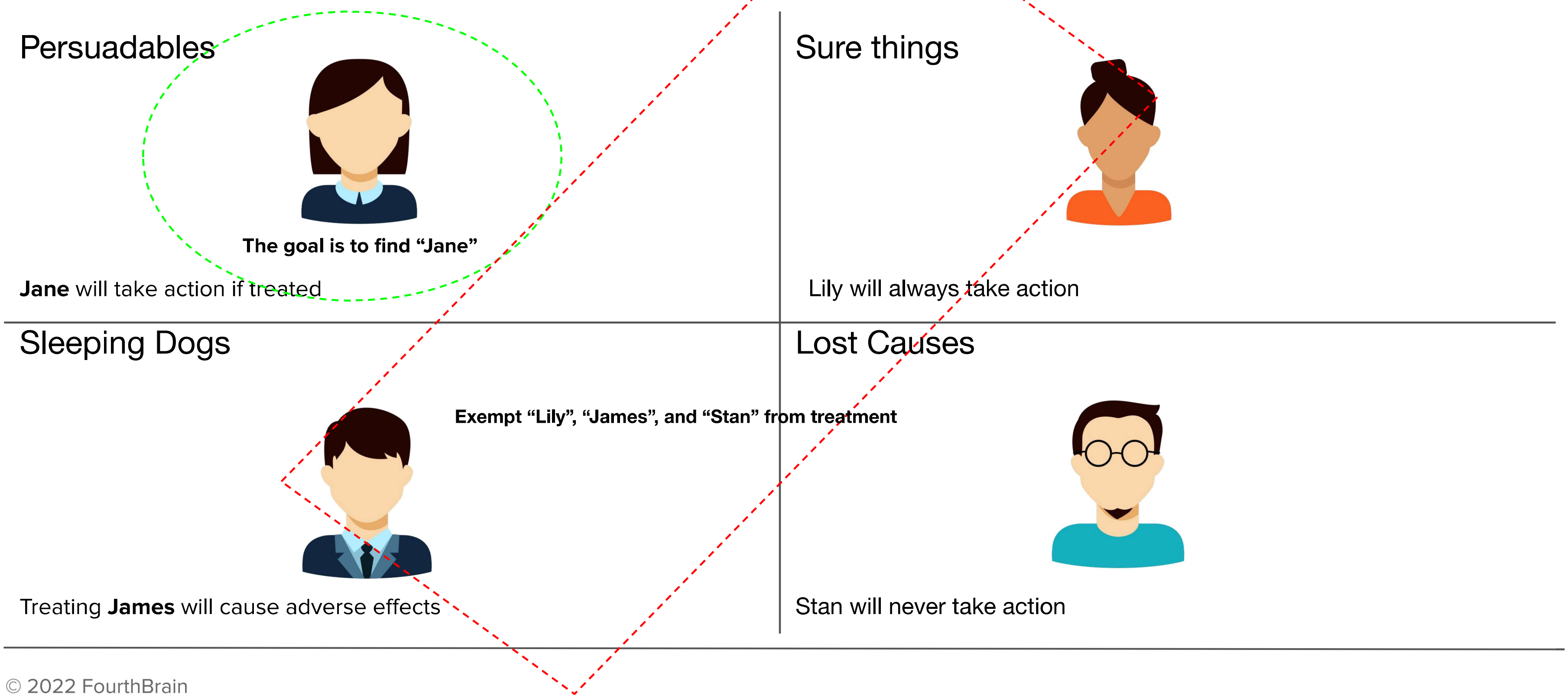
Uplift modeling

Uplift models helps in identifying **users that are more likely to take action or respond positively after treatment exposure** like a marketing campaign or promotional offer

Classic Uplift Segments

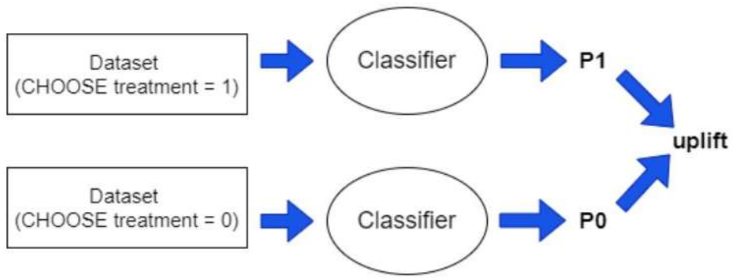
Persuadables  **Jane** will take action if treated	**Sure things**  **Lily** will always take action
Sleeping Dogs  Treating **James** will cause adverse effects	**Lost Causes**  **Stan** will never take action

Classic Uplift Segments



Uplift Model

Two Model



The training process:

$$model^T = fit\begin{pmatrix} x_{11} & \cdots & x_{1k}, & y_1 \\ \vdots & \ddots & \vdots & \cdots \\ x_{p1} & \cdots & x_{pk} & y_p \end{pmatrix}, \quad model^C = fit\begin{pmatrix} x_{11} & \cdots & x_{1k}, & y_1 \\ \vdots & \ddots & \vdots & \cdots \\ x_{q1} & \cdots & x_{qk} & y_q \end{pmatrix}$$

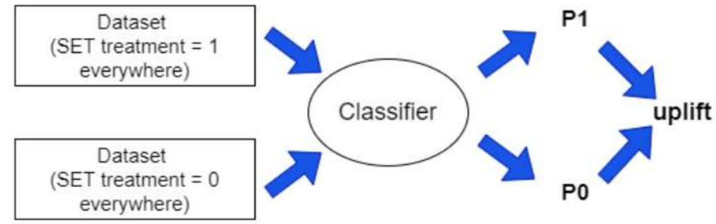
X_{train_treat} Y_{train_treat} $X_{train\_control}$ $Y_{train\_control}$

The process of applying the model:

$$\begin{matrix} model^T \\ predict \\ proba \end{matrix}\begin{pmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mk} \end{pmatrix} - \begin{matrix} model^c \\ predict \\ proba \end{matrix}\begin{pmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mk} \end{pmatrix} = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}$$

X_{test} X_{test} $uplift$

One Model



The training process:

$$fit\begin{pmatrix} x_{11} & \cdots & x_{1k} & w_1 & y_1 \\ \vdots & \ddots & \vdots & \cdots, & \cdots \\ x_{n1} & \cdots & x_{nk} & w_n & y_n \end{pmatrix}$$

X_{train} W_{train} Y_{train}

The process of applying the model:

$$\begin{matrix} predict \\ proba \end{matrix}\begin{pmatrix} x_{11} & \cdots & x_{1k} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{m1} & \cdots & x_{mk} & 1 \end{pmatrix} - \begin{matrix} predict \\ proba \end{matrix}\begin{pmatrix} x_{11} & \cdots & x_{1k} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ x_{m1} & \cdots & x_{mk} & 0 \end{pmatrix} = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}$$

X_{test} W_1 X_{test} W_0 $uplift$

Solution Architecture for Uplift Model

Data Storage

Treatment History
Control History
User Profiles

- Source: Google Analytics Reports

Data Processing

Data Cleansing
Feature Engineering
Dat Merge
Train/Test Split
Treatment/Control Split

- Platforms used: Jupyter Notebooks, Google Colab, VScode
- Libraries: Scikit learn
- EDA: Matpltolib, Seaborn

Model Building

Model Preparation
Model Train
Model Selection
Version Control

- Models: Logistic Regression
- Uplit Modelling: Two Model & Single Model Approach
- AutoML: TPOT
- Model Explainability: SHAP & LIME reports
- Version Control: Github

Model Deploy

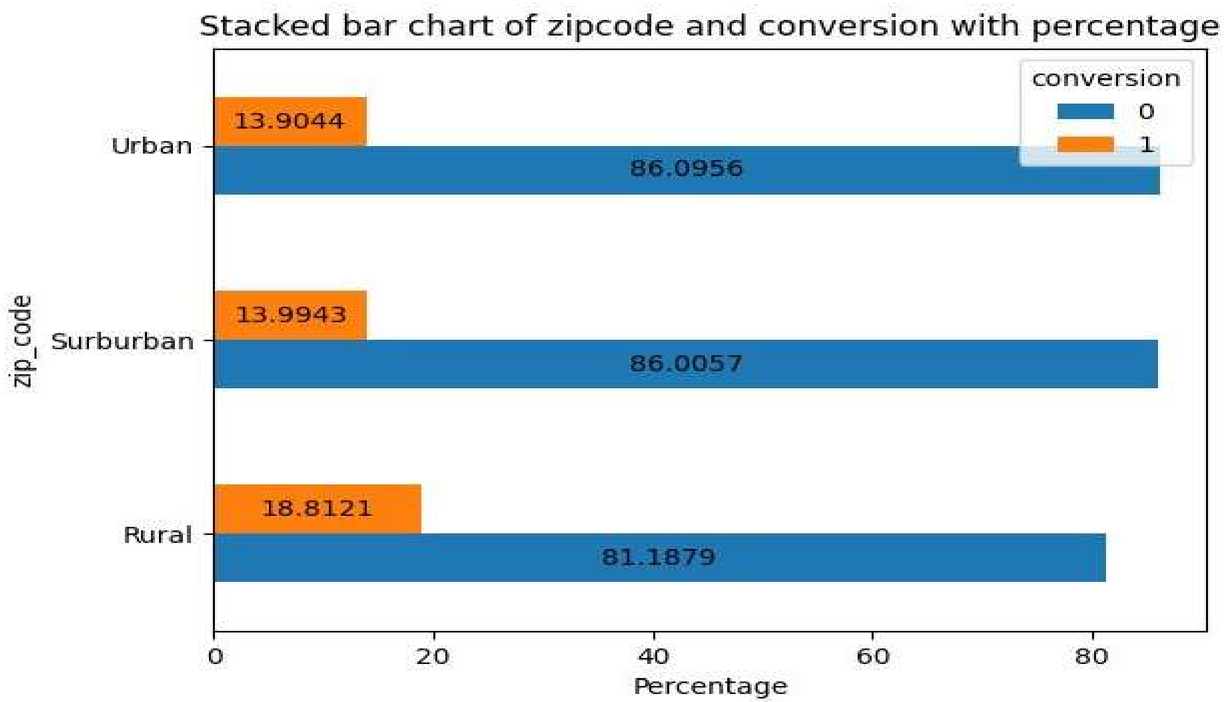
Inference
API
Web Service
Model Monitoring

- Streamlit, FASTAPI
- Monitoring: MLflow

Data Source

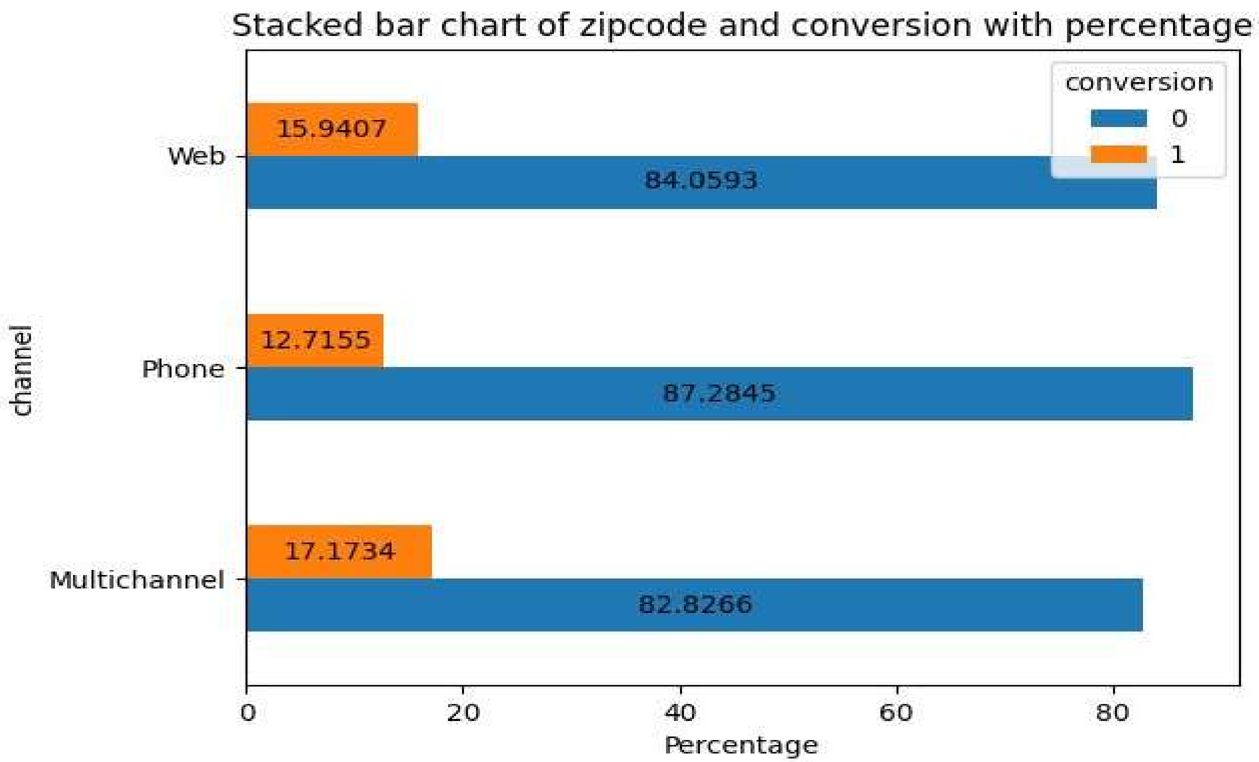
- The dataset contains 64,000 customers who last purchased within twelve months. The customers were involved in an e-mail marketing campaign
- 1/3 were randomly chosen to receive an e-mail campaign featuring a Discount offer
- 1/3 were randomly chosen to receive an e-mail campaign featuring a Buy One Get One offer
- 1/3 were randomly chosen to not receive an e-mail campaign.
- Goal:
- 1. Did the treatment have an impact?
- 2. Which campaign performed better?

EDA: Zipcode

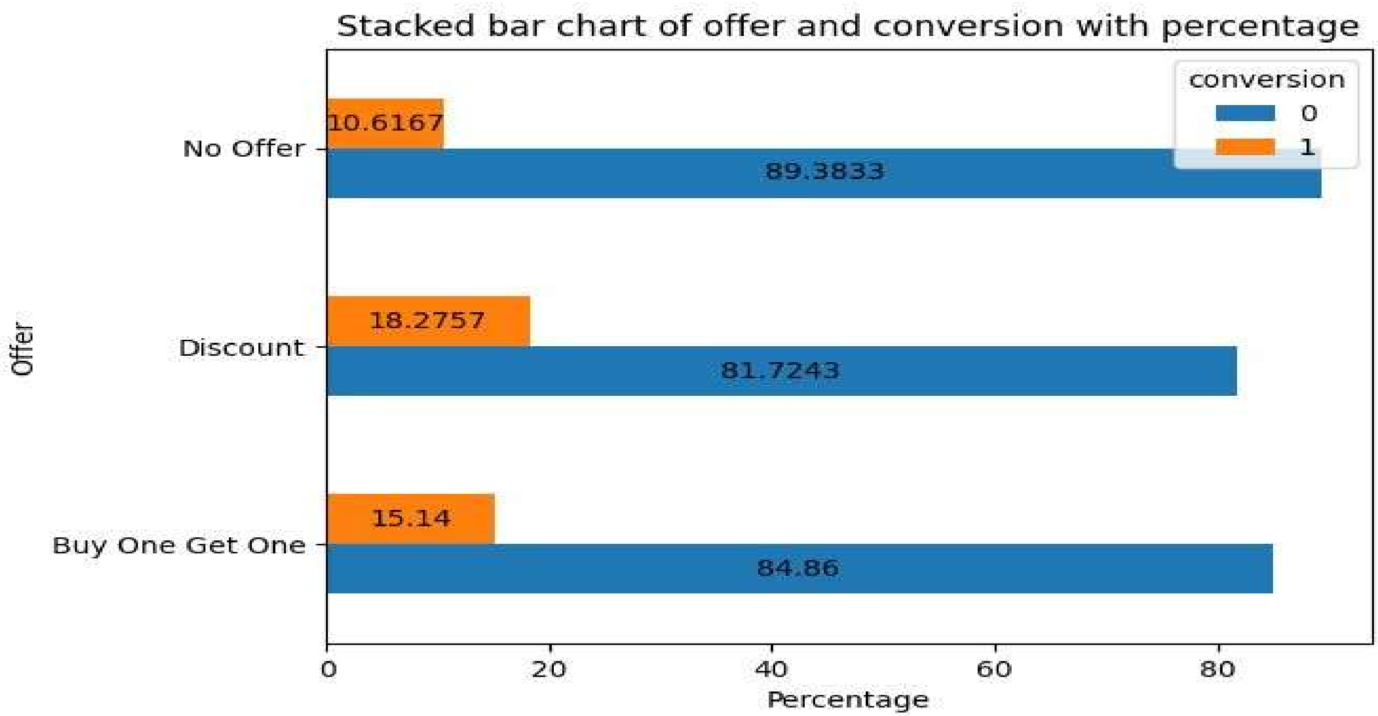


Stacked bar chart of zipcode and conversion with percentage
conversion
0
1
13.9044
86.0956
13.9943
86.0057
18.8121
81.1879
Urban
Surburban
Rural
zip_code
0
20
40
60
80
Percentage

EDA: Channels



EDA: Impact of Treatment Conversion



Model Selection

- Base Model: Logistic Regression
- Ensemble Model: XGBoost
- Uplift Model: Two Model Approach vs Single Model Approach

Logistic Regression Results

-

```
accuracy: 0.85325
precision: 0.5
recall: 0.0008517887563884157
f1 score: 0.0017006802721088437
confusion matrix:
[[13650     2]
 [ 2346     2]]
```

XGBClassifier Results: Similar to LR

```
accuracy: 0.8515
precision: 0.1956521739130435
recall: 0.0038330494037478705
f1 score: 0.007518796992481203
confusion matrix:
[[13615    37]
 [ 2339     9]]
```

Possible Issues & Solutions

- Biased data
- Drop Duplicates
- Set Class weight to “balanced” for models
- New Results: Slightly better

```
Accuracy:  0.6053870292887029
Precision:  0.21499380421313508
Recall:  0.5891341256366723
F1 Score:  0.31502496595551527
Confusion Matrix:
 [[3936 2534]
 [ 484  694]]
```

AutoML Implementation: TPOT

```
Successfully installed deap 1.3.3 stopit 1.1.2 tpot 0.11.7 update-checker 0.18.0 xgboost 1.7.4
```

In [62]:

```python
%%time
from tpot import TPOTClassifier
tpot = TPOTClassifier(generations=10,
                      population_size=16,
                      scoring=None,# YOUR CODE HERE
                      verbosity=2,
                      random_state=42)
tpot.fit(X.values, y.values)
print(f"Tpop score on test data: {tpot.score(X, y):.2f}")
tpot.export('tpot_uplift.py')
```

```
Optimization Progress:   0%|          | 0/176 [00:00<?, ?pipeline/s]
Generation 1 - Current best internal CV score: 0.8459124452228648

Generation 2 - Current best internal CV score: 0.8459124452228648

Generation 3 - Current best internal CV score: 0.8459124452228648

Generation 4 - Current best internal CV score: 0.8459124452228648

Generation 5 - Current best internal CV score: 0.8459124452228648

Generation 6 - Current best internal CV score: 0.8459124452228648

Generation 7 - Current best internal CV score: 0.8459124452228648

Generation 8 - Current best internal CV score: 0.8459124452228648

Generation 9 - Current best internal CV score: 0.8459909073648813

Generation 10 - Current best internal CV score: 0.8459909073648813
```

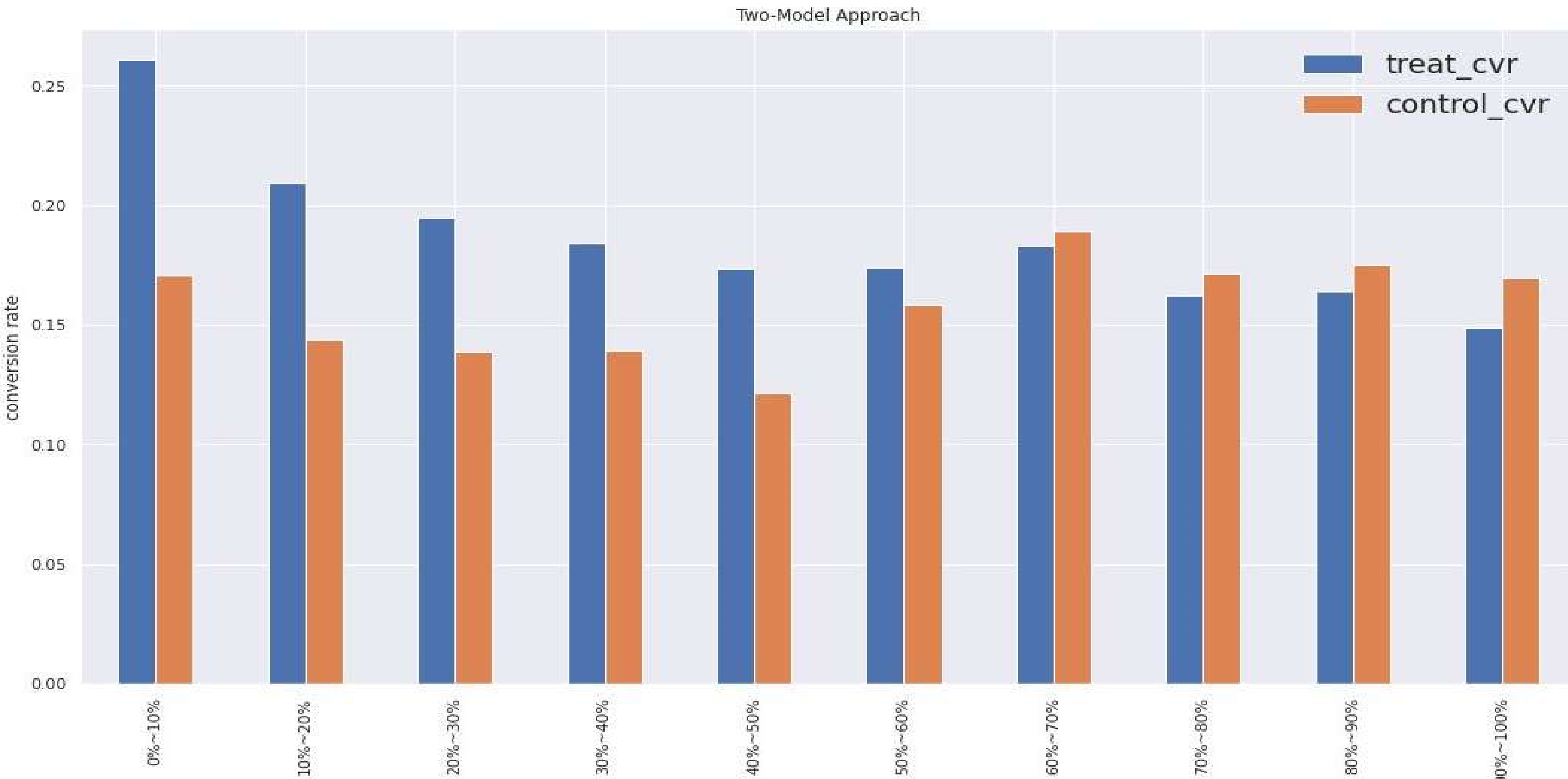
Uplift Implementation: Two Model

-

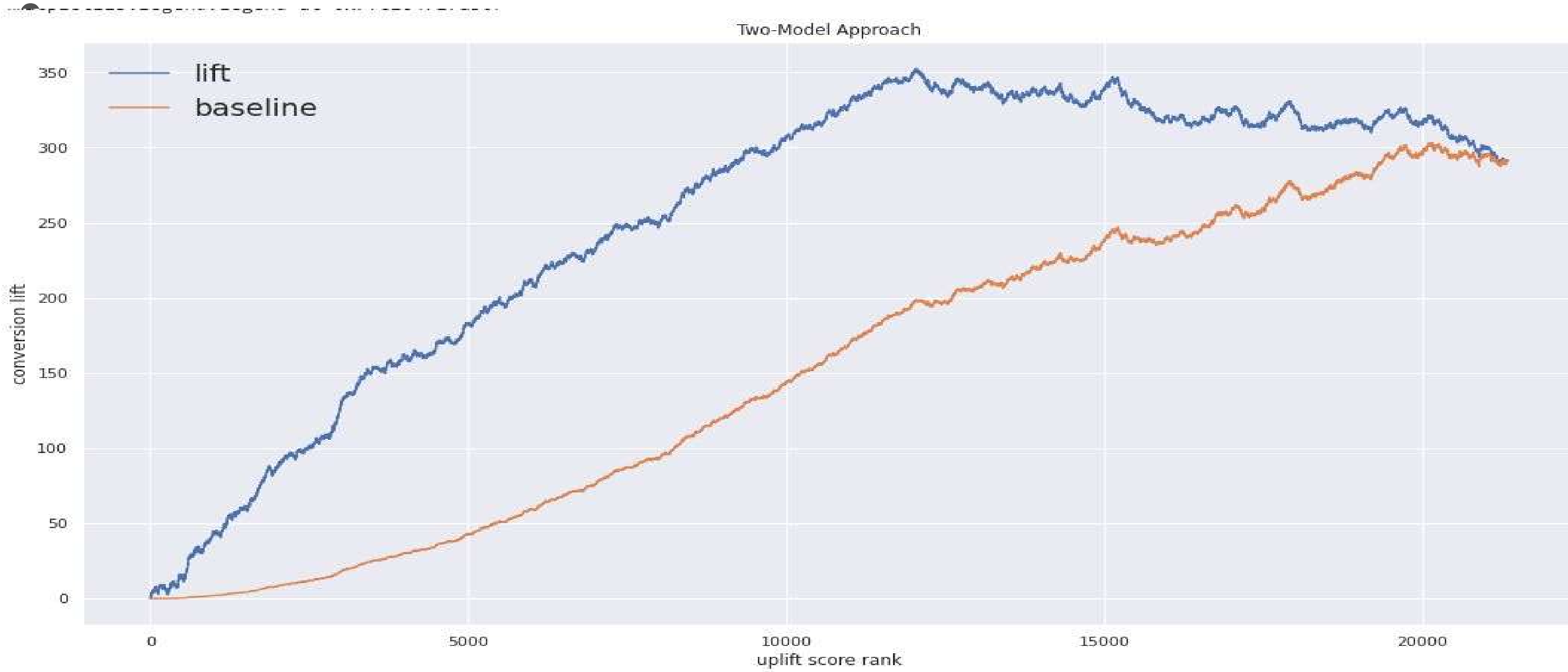
```
3] print('treat accuracy: ', sum(model_treat.predict(X_test)==y_test)/len(y_test))
   print('control accuracy: ', sum(model_control.predict(X_test)==y_test)/len(y_test))
```

```
treat accuracy:  0.8285941818522509
control accuracy:  0.8285941818522509
```

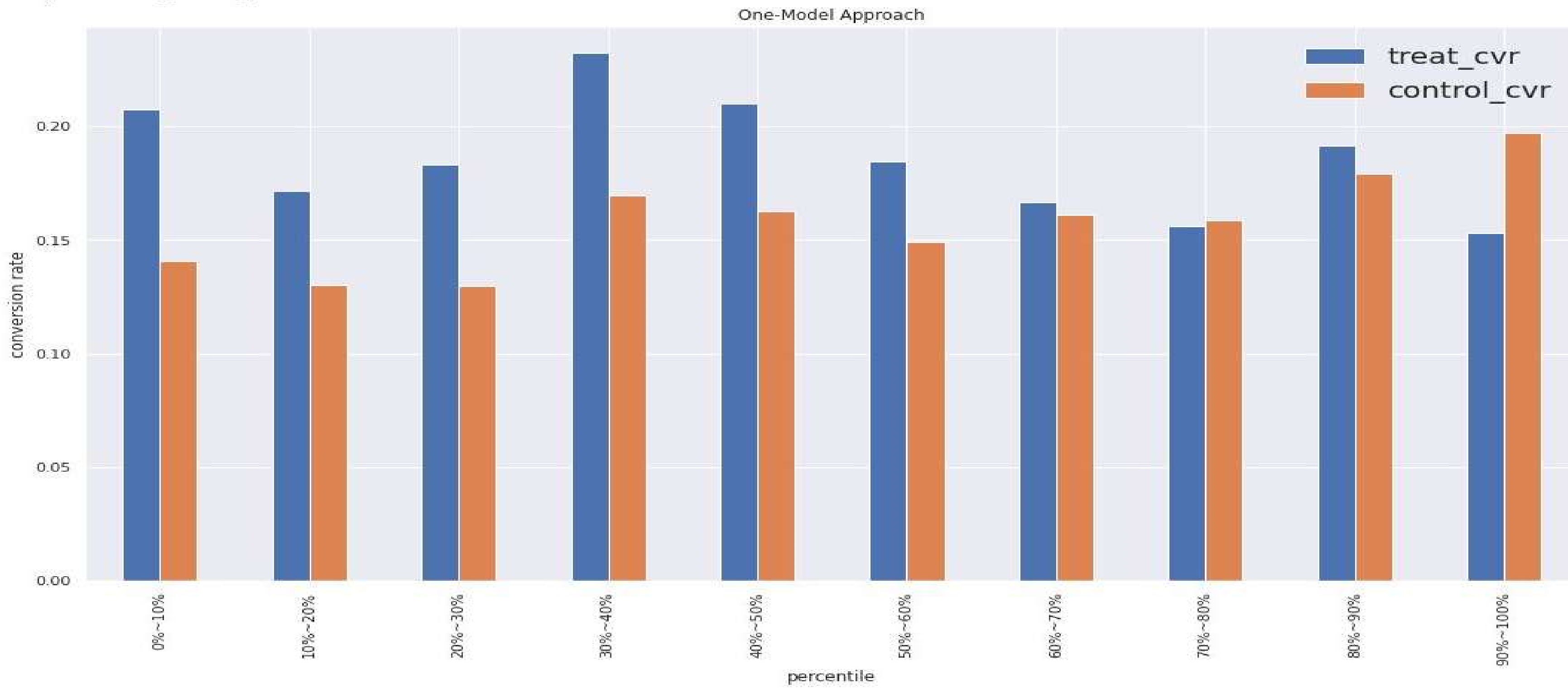
Uplift Implementation: Two Model



Uplift Implementation: Area Under the Uplift Curve

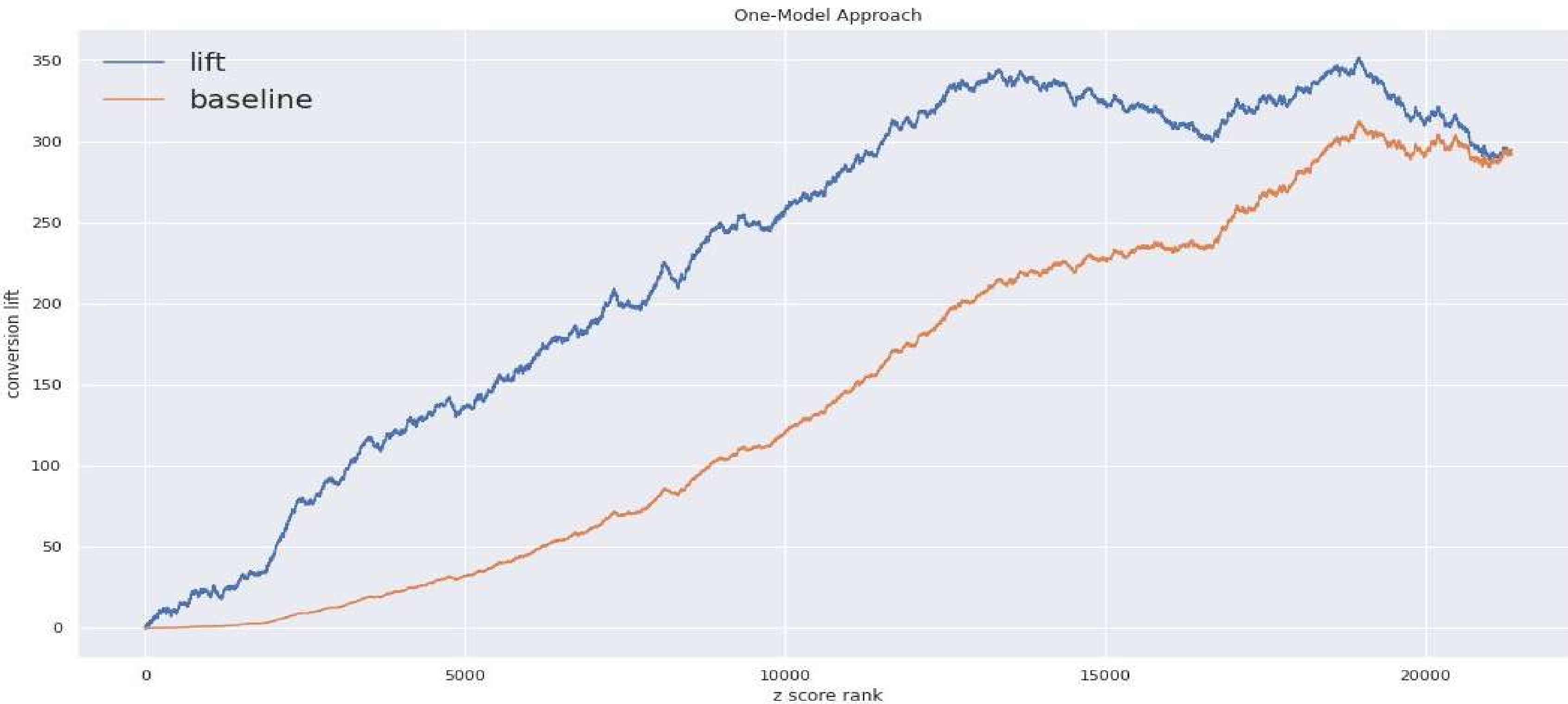


Uplift Implementation: One Model

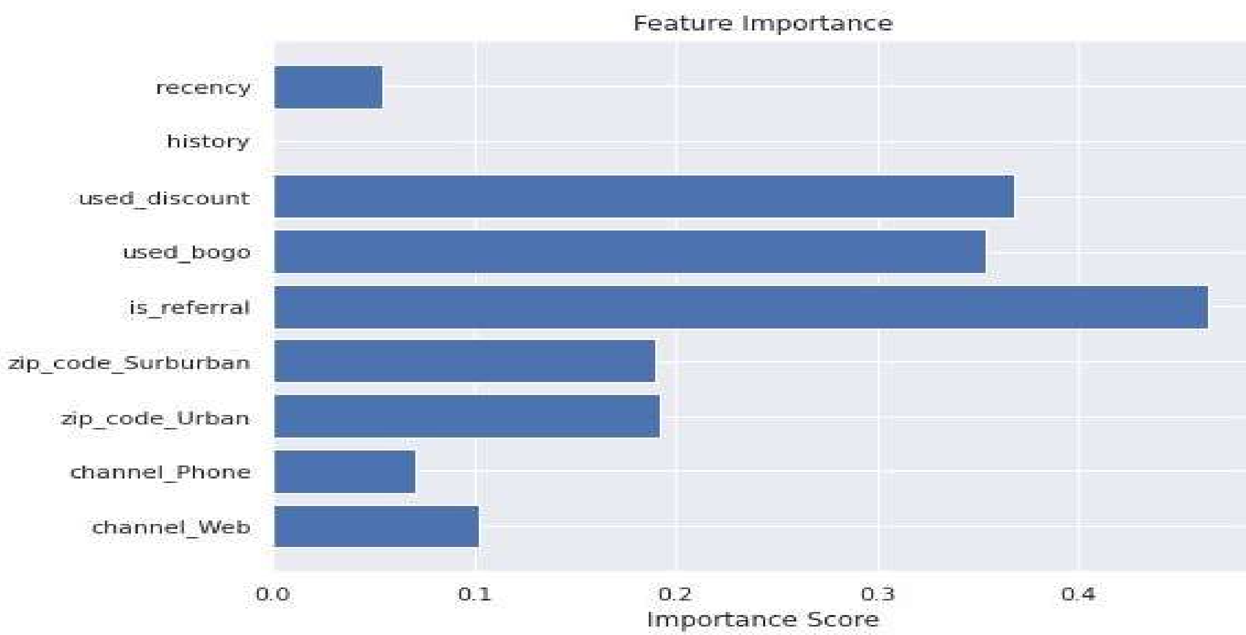


Uplift Implementation: Area Under the Uplift Curve

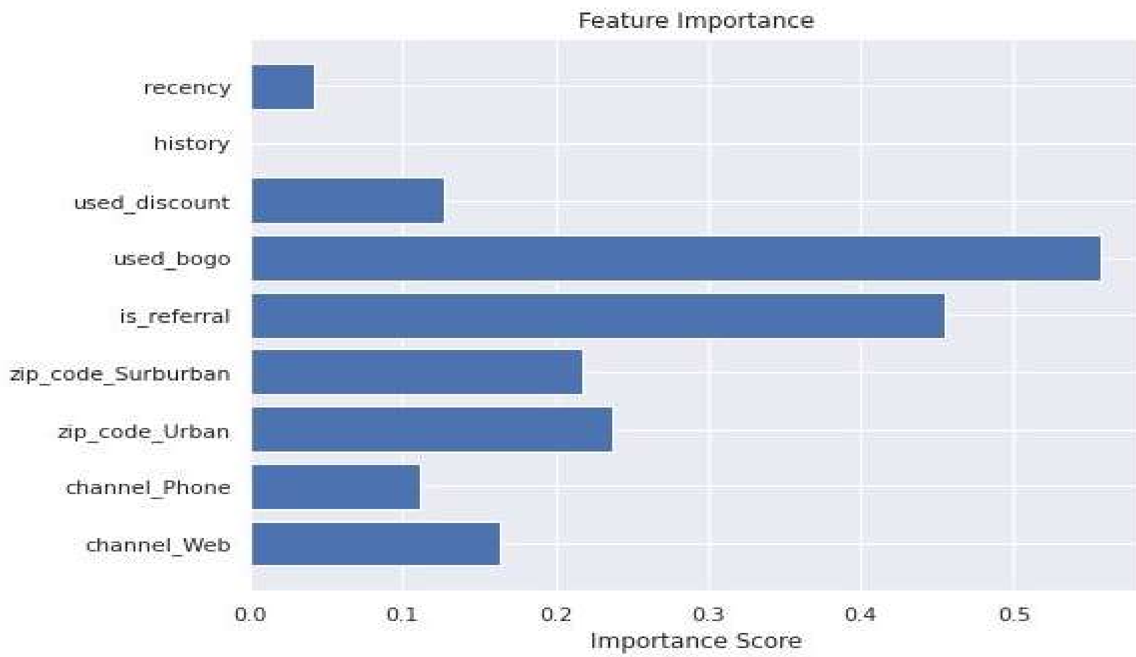
-



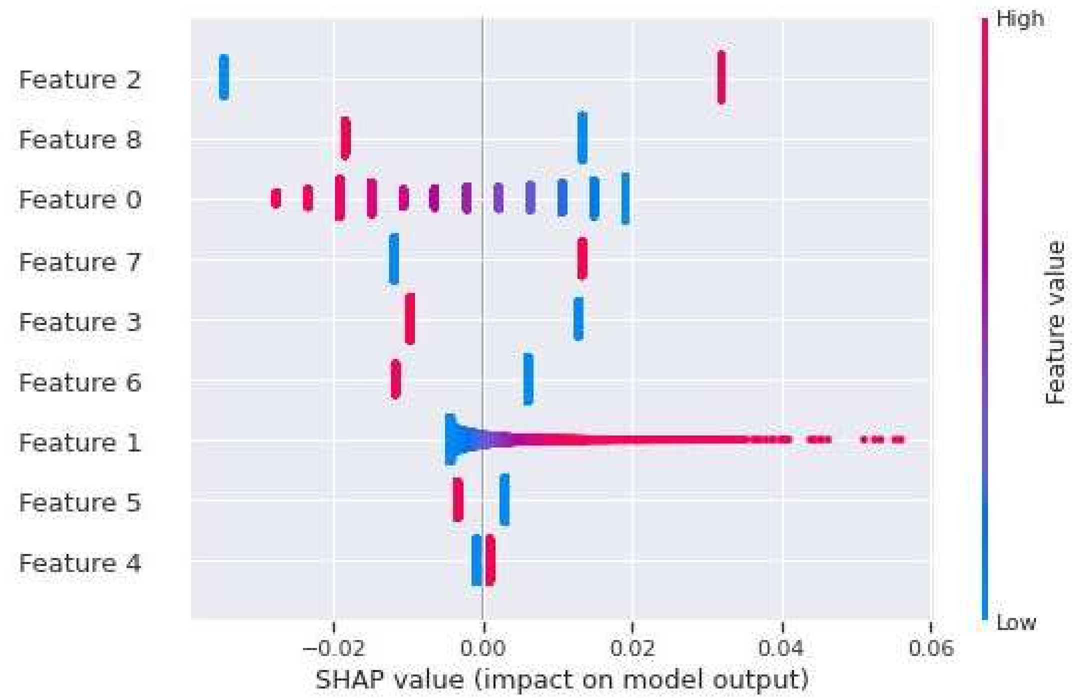
Explainability/Interpretability: Two Model:Treatment



Explainability/Interpretability: Two Model:Control

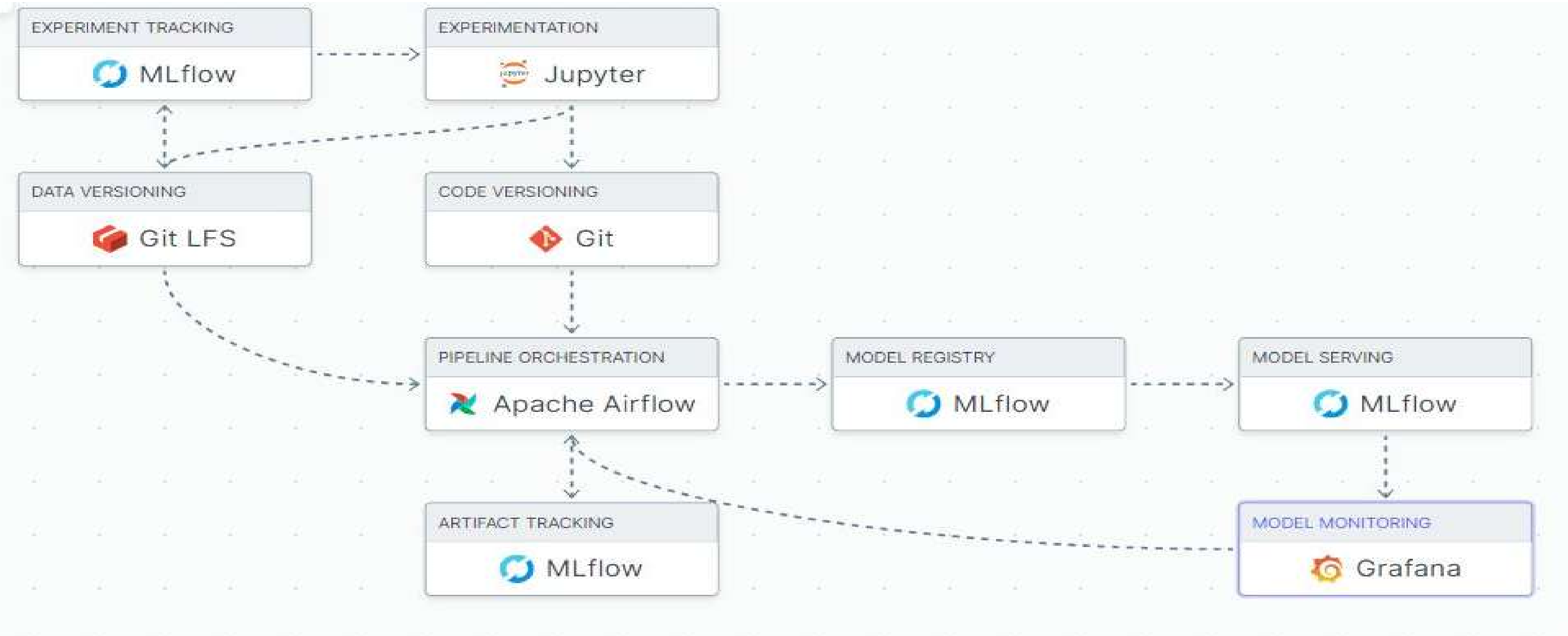


Explainability/Interpretability: SHAP: Single Model



	features	weights
1	history	0.000021
4	is_referral	0.001751
0	recency	0.004254
5	zip_code_Surburban	0.006383
6	zip_code_Urban	0.017707
3	used_bogo	0.022559
7	channel_Phone	0.025144
8	channel_Web	0.031803
2	used_discount	0.066748

MLE Stack


EXPERIMENT TRACKING
MLflow
EXPERIMENTATION
Jupyter
DATA VERSIONING
Git LFS
CODE VERSIONING
Git
PIPELINE ORCHESTRATION
Apache Airflow
MODEL REGISTRY
MLflow
MODEL SERVING
MLflow
ARTIFACT TRACKING
MLflow
MODEL MONITORING
Grafana

Model Performance Comparison

- Logistic Regression and XGBClassifier yielded high Accuracy scores and low Precision , Recall & F1 scores until the class-weights were balanced
- AutoML yielded high accuracy scores but this could be biased as TPOT did not have the class_weight feature
- Comparison of the uplift models: Two Model & Single Model had similar outcomes as can be seen in the density plots
- Uplift scores were significant for both Discount offer and BOGO offer with Discount being the most effective

Business Value

- Increased revenue
- Improved customer engagement
- Reduced marketing costs
- Improved customer retention
- Better decision-making

Conclusions & Future Work

- Overview of how the model can be improved
- Explanation of how the model can be integrated into the business process
- Optimize Uplift Model
- Recommendation System
- MLE Stack Optimization

Thank You! Questions?