American Express - Arrival Probability

AI Studio Final Presentation

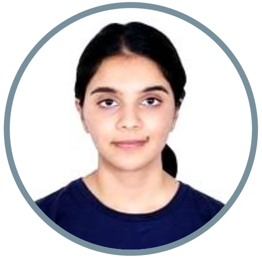
Break Through Tech AI @ UCLA
Dec 6, 2022

Our Team

Yitian Yan
Computer Science @ USC

Jade Eng
Computer Information Systems @ CPP

Jane Zou
Statistics @ UCLA



Shruthi Srinarasi
AI Studio TA

Saurabh Gupta
Challenge Advisor

Akram Alim
Challenge Advisor

Presentation Agenda

1. **Project Overview:**
   a. Goals
   b. Business Impact
   c. Our Approach
   d. Resources
2. **Data Preprocessing:**
   a. EDA
   b. Feature Engineering
3. **Model Selection and Evaluation:**
   a. Insights
   b. Takeaways
4. **Final Thoughts**
   a. What We Learned
   b. Potential Next Steps

AI Studio Project Overview

Machine learning challenge to **maximize** American Express’ revenue by targeting the **highest ROI customers** in campaign duration

Problem Statement

Our Goal

1. Efficient use of **GitHub** (structured development engineering practices), CI/CD
2. Efficient use of SciKit/libraries to learn and understand **model building**
3. Cover all phases of model building exercise
4. Learn to **think outside the box collaboratively** to come up with alternative solutions

Business Impact

- Determine the profile of customers who are predicted to be **high ROI**
- **Maximize investment** of ad campaign funds more effectively

Our Approach

Milestone	Completion Date
1. Team Building: get to know how to collaborate effectively	08/19
2. Business Understanding: build business sense for AI Studio project, align project expectations	09/04
3. Data Understanding and Preparation: preprocess raw data	10/09
4. Modeling and Evaluation: choose model candidates, feature selection and hyperparameter techniques	11/06
5. Iterate and Prepare Your Presentation: improve selected model, prepare final presentation to Challenge Advisor and company	12/04

Resources We Leveraged

- Biweekly meetings with our **Challenge Advisors** Saurabh and Akram, and our **TA**, Shruthi
- **eCornell Machine Learning Fundamentals** Labs and Assignments
- **Documentation** on scikit learn, pandas
  - Binary classification
  - Random Forest Classifier
  - Gradient Boosting Classifier
  - Logistic Regression
  - XGBoost Model

Data Preprocessing

Exploratory Data Analysis

Account Data

- Remove NULL/NA values, replace with **mode** of column
- Combine into one dataset
- Replace duplicates with **maximum account date**
- Group by account id

```python
from functools import reduce
```

```python
#define list of DataFrames
#Replace df4 with maximums
dfs = [df1, df2, df3, maximums, df5, df6, df7, df8, df9, updated_wp1, updated_wp2]
```

```python
#merge all DataFrames into one
final_df = reduce(lambda  left,right: pd.merge(left,right,on=['ac_id'],
                                              how='left'), dfs).fillna('n/a')
```

```python
final_df.head()
```

	ac_id	cycle_dt	payment_due_dt	new_account_indicator	member_since_in_months	spend_active	is_active_balance	has_credit_limit_reached
0	AC4592fa29-384c-4d58-a74b-2ac5780e884f	3/14/18	4/8/18	0	n/a	1	0	0
1	AC4fb74ef4-3f46-4e6a-9ae7-657320b463bc	n/a	n/a	0	n/a	1	0	0
2	AC10e4b9f0-e76f-4da7-a0d3-99988ef23f08	3/7/18	4/2/18	0	185.0	1	0	0
3	ACe990a57f-8061-4303-97f6-83b6c580a5f1	n/a	n/a	0	324.0	1	0	0
4	ACcec24e2f-c5d0-49ac-ae7d-8106e50646ce	3/28/18	4/23/18	0	19.0	1	0	0

Creating the Label

Dependent Dataset

- **Combine responses** for all 30 days of June 2018 campaign
- Test models on **day 3, 10, 20, and 29**
- Label **binary indicator** of whether a customer arrived on the website

```python
filename = os.path.join(os.getcwd(), "DEPENDENT_DATA.txt")
df2 = pd.read_table(filename)
df2
```

	Unnamed: 0	ac_id	arr_ind_1	arr_ind_2	arr_ind_3	arr_ind_4	arr_ind_5	arr_ind_6	arr_ind_7	arr_ind_8	...	arr_ind_21	arr_ind_22	arr_ind_23
0	0	AC4592fa29-384c-4d58-a74b-2ac5780e884f	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	1.0	1.0	1.0
1	1	AC4fb74ef4-3f46-4e6a-9ae7-657320b463bc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	1.0	1.0	1.0
2	2	AC10e4b9f0-e76f-4da7-a0d3-99988ef23f08	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	...	1.0	1.0	1.0
3	3	ACe990a57f-8061-4303-97f6-83b6c580a5f1	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	...	1.0	1.0	1.0
4	4	ACcec24e2f-c5d0-49ac-ae7d-8106e50646ce	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	...	1.0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
897840	899995	AC3bff3721-35d8-4ef7-a5a5-753be4f67d2d	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	...	1.0	1.0	1.0
897841	899996	AC7aa46ab7-e3db-4075-b04c-b092372cf1fb	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	1.0
897842	899997	AC5d08eaa6-a54d-4cb4-a0b7-b49e9e482e08	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	...	1.0	1.0	1.0
897843	899998	ACf80349b3-0be2-4d3d-bf96-f1ee507b9c6f	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	1.0	1.0	1.0
897844	899999	AC2384f6f9-8cd5-4247-aaa9-78799e937e5e	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	...	1.0	0.0	1.0

897845 rows × 32 columns

Feature Engineering

Web Records, Purchase Records

- Calculate **maximum and minimum differences** between purchase date and web visit date, proximity from payment due date
- Count number of times a customer **visits and purchases within 30, 60, 90, 120, and 180 days** *before* dependent data time frame (June 2018)
- Combine dependent data dates with feature engineering dataset
- Randomly select **500,000 unique accounts**

	ac_id	visited_page	diff_dates	Max Date Diff	Min Date Diff	Purchase 30_x	Purchase 60_x	Purchase 90_x	Purchase 120_x	Purchase 180_x	...	arr_ind_21	arr_ind_22	arr_ind_23
0	ACe761e40e-3259-4e4f-93f9-8f2f2ed34388	REWARDS	52 days	380 days	31 days	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	1.0
1	AC612ca133-52a6-456d-a978-e6ecfa9e87d6	PRICINGENGINE	155 days	386 days	91 days	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	1.0
2	AC200056f5-32de-4cbd-927d-278f3ee18282	PRICINGENGINE	257 days	336 days	5 days	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	1.0
3	AC200056f5-32de-4cbd-927d-278f3ee18282	PRICINGENGINE	209 days	336 days	5 days	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	1.0
4	AC4c2519a1-4934-47e6-8c22-2ccfa240b586	MYACCOUNT	246 days	379 days	7 days	0.0	0.0	1.0	1.0	1.0	...	1.0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
499994	ACdc6643c3-55a3-4f1b-b07a-fd486c85fb2f	PRICINGENGINE	38 days	327 days	3 days	0.0	0.0	0.0	1.0	1.0	...	1.0	1.0	1.0
499996	AC8d97243d-d43d-4f68-bf7f-30d07c485a54	REWARDS	88 days	224 days	5 days	0.0	0.0	0.0	1.0	1.0	...	1.0	1.0	1.0
499997	AC762d9492-ad39-4d14-888a-8ac279982d85	PRICINGENGINE	223 days	349 days	47 days	1.0	1.0	1.0	1.0	1.0	...	1.0	1.0	1.0
499998	AC043ad7c3-807f-4f92-9c31-3161b69e1994	REWARDS	335 days	376 days	27 days	1.0	1.0	1.0	1.0	1.0	...	1.0	1.0	1.0
499999	AC38c6c972-5a5f-4a0f-bf9e-bad4c3c06025	REWARDS	247 days	295 days	8 days	0.0	1.0	1.0	1.0	1.0	...	1.0	1.0	1.0

498668 rows × 66 columns

Model Selection and Evaluation

Random Forest

- Best Random Forest models: **day 3 and day 29**
- Calculate weighted **precision and recall**

	Day 3	Day 29
Precision	0.87	0.99
Recall	0.91	0.99

```
Accuracy for Each Model
Day 3:   0.90986321181811
Day 10:   0.6101871038702973
Day 20:   0.8488098638194955
Day 29:   0.9863515656808114
```

Logistic Regression

- Best logistic regression models: **day 20 and day 29**
- Select **best five features** for each model
- Calculate **precision and recall** based on confusion matrix

	Day 20	Day 29
Precision	0.816	0.984
Recall	1	1

Log loss: 0.4777902940221649
Accuracy: 0.8158190579785004

	Predicted: Arrival 20	Predicted: Not Arrival 20
Actual: Arrival 20	134252	0
Actual: Not Arrival 20	30309	0

Log loss: 0.08359528337434832
Accuracy: 0.983683861911385

	Predicted: Arrival 29	Predicted: Not Arrival 29
Actual: Arrival 29	161876	0
Actual: Not Arrival 29	2685	0

Gradient Boosting Machine

- Best GBM days: **day 3 and 29**
- Testing on **learning rates** 0.05 to 1
- Ranked based on learning rate, confusion matrix **precision** and **recall**
- No difference in learning rate hyperparameter

```
Learning rate:  1
Accuracy score (training): 0.915
Accuracy score (validation): 0.914
```

Day 3

```
Learning rate:  1
Accuracy score (training): 0.983
Accuracy score (validation): 0.984
```

Day 29

	Day 3	**Day 29**
Precision	0.86	0.98
Recall	0.91	0.98

XGBoost

- Best XGBoost models: **day 3 and day 29**
- Calculate **Mean Squared Error** and **accuracy** for each model
- Default parameters from scikit learn

	Day 3	Day 29
Precision	0.92	0.98
Recall	0.91	0.98

```
Mean Squared Error for Each Model
Day 3:  0.0782872103394022l
Day 10:  0.2488198117483265 7
Day 20:  0.1497806111522862 5
Day 29:  0.0159928332917148 97
```

```
Accuracy for Each Model
Day 3:  0.914153414235450 7
Day 10:  0.5252094967823482
Day 20:  0.8162322786079326
Day 29:  0.98369601545931 3
```

Model Comparison

Model Name	Description	Pros	Cons
Random Forest	Output of multiple decision trees to reach a single result	• High accuracy for days 3, 29 • Highest precision and recall for day 29	• Low accuracy for day 10 • Computationally inefficient
Logistic Regression	Binomial estimation of probability of customer arriving to website	• High accuracy, precision, and recall for days 20, 29 • Easy to implement	• Low accuracy and low precision/recall for day 10
Gradient Boosting Machine	Using gradients in loss function, measure indicating how good model fitting data	• Computationally efficient • High precision, recall, and accuracy for day 29	• Lowest accuracy of all the models for day 10 • Confusion matrix unrepresentative of data
XGBoost	Scalable extreme gradient boosting decision tree	• High accuracy, high precision and recall, and low MSE for days 3, 29	• Low accuracy for day 10 • High MSE for days 10 and 20 • Computationally inefficient

Insights and Key Findings

- Ranked based on precision and recall
  1. Random forest
  2. XGBoost
  3. GBM
  4. Logistic Regression

- Best Model for Each Day
  - Day 3: XGBoost
  - Day 10: Random Forest
  - Day 20: Logistic Regression
  - Day 29: Random Forest

- Ranked based on accuracy
  1. Random Forest
  2. XGBoost and GBM
  3. Logistic Regression

- **Our Selected Model**
  - Random Forest
    - Highest accuracy for day 10
    - Best evaluation metrics

What We Learned

- Utilizing GitHub, Geeks for Geeks, Towards Data Science, and Sci-kit Learn **Documentation**
- **Project management:** Slack, Trello
- Gradient Boosting and XGBoost **Classification**
- Hands-on experience with **ML pipeline**

Potential Next Steps

- Test for **multiple days** apart from four models
- Classification for **days 1, 2, and 30**
- Feature engineering dataset inclusive of all **900,000 unique accounts**
- **Docker image deployment** and Medium article
- **GitHub** command practice

Questions?