Gamers Acquisition Report

TRUEDATA

Presented by: Puhsin Huang

OUR TEAM

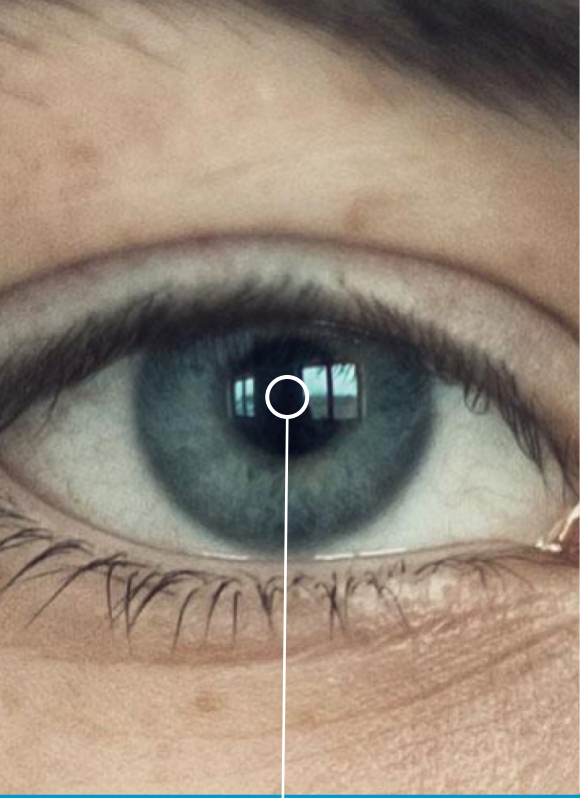
Puhsin Huang
Data Science Analyst

Puhsin brings over 3 years of data-driven marketing experience to TrueData. Having worked at Ipsos and RAPP, Puhsin is experienced in a variety of analytical tasks, ranging from crafting complex SQL queries, creating Tableau/PowerBI dashboards, conducting ad-hoc market research, and building machine learning models via Python.

Puhsin holds an MS in Business Analytics from the University of Southern California.

AGENDA

1. Business Background & Goal
2. Data Description & EDA
3. Solution Structure & Model Details
4. Result Interpretation
5. Suggestions & Next Steps



1 Business Background & Goal:

A prospective mobile gaming client is looking to acquire new casual gamers for their jigsaw puzzle games.

TrueData's goal is to **build the best model possible to predict the users who are likely to have jigsaw puzzles installed.**

Target variable is "**desired_outcome**" , which is defined as:

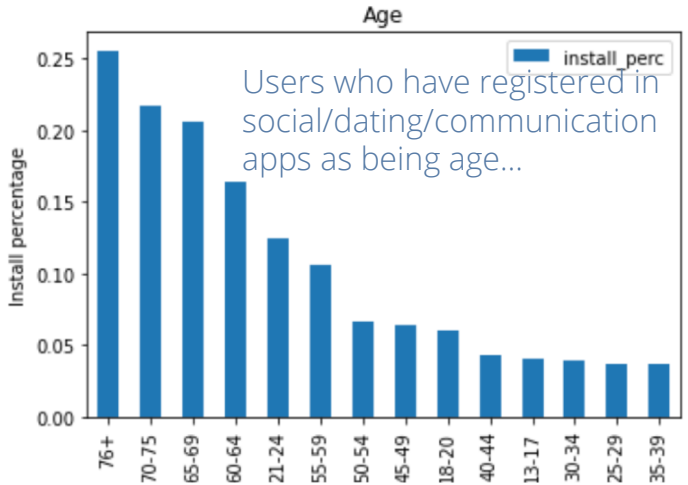
- 1 = Users who currently have jigsaw puzzle casual games installed
- 0 = Does not have jigsaw puzzle casual games installed

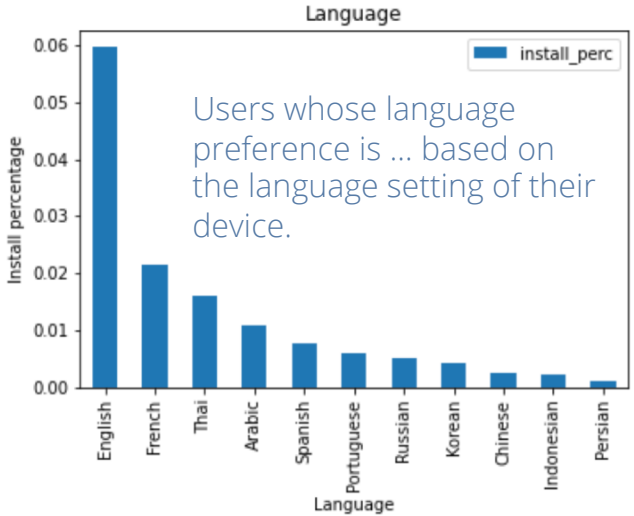
2

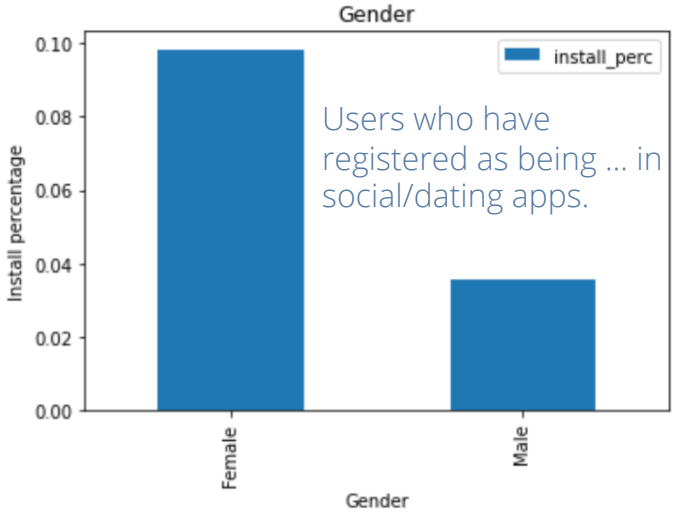
Data Description & EDA

- Number of records: 955189
- Number of fields: 9292
- Except for maid (Mobile Advertising ID), all fields are binary since one-hot encoding was performed.
- Target variable "**desired_outcome**":
  - 899,972 zeros
  - 55,217 ones

→ **Class Imbalance**

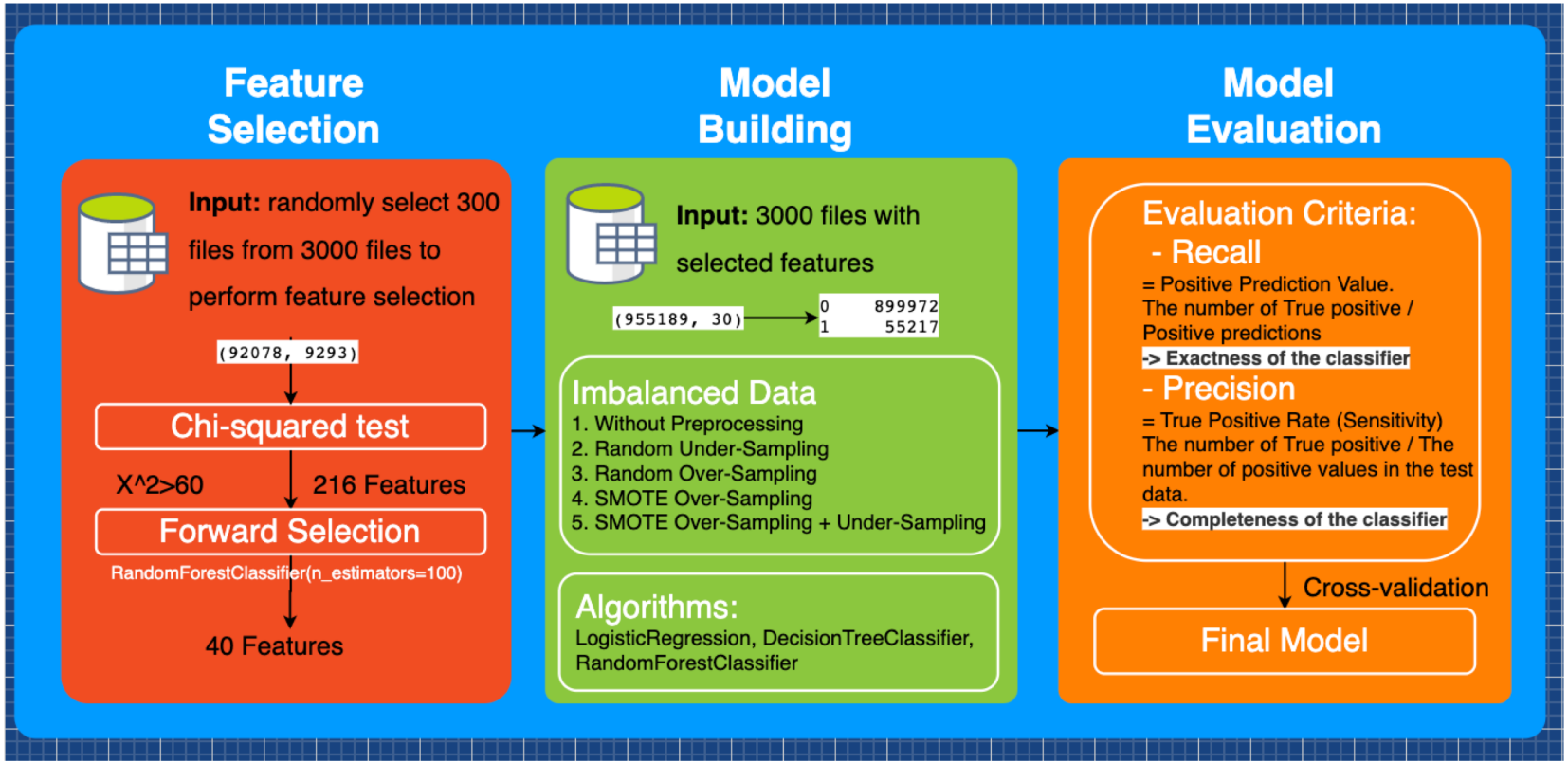






* Noted that the demographic groups are not representative to the whole population

3 Solution Structure & Model Details



4 Result Interpretation

Predict all zeros

```
[[179993      2]
 [ 11042      1]]

precision=0.33
recall=0.0
accuracy=0.94
```

Accuracy is not the best metric to use when evaluating imbalanced datasets as it can be misleading.
The classifier will always "predicts" the most common class without performing any analysis of the features and it will have a high accuracy rate.
In our case...

Recall:
The % of the installation correctly identified among all predictions.

Precision:
The % of the installation the model identified among all installation.

With limited budget, I suggest to use **Recall** as our main evaluation criteria to maximize the possibility of installation among people we target.

Meaning ...
Out of 100 people we target, 30 of them are highly possible to install the game.

Final Model
- Feature Used: 40
- Algorithm: LogisticRegression()
- Accuracy: 0.873
- Precision: 0.21
- **Recall: 0.433**

5 Suggestions & Next Steps

- APP installation related fields are particularly important for the model. While targeting customers, the company can find people who have similar behavior to its current customers in terms of these fields.

Important features:

APP201882	Users who currently have the FreeCell Solitaire app installed on their mobile device or tablet
APP201971	Users who currently have the Charm King,Ñ¢ app installed on their mobile device or tablet
APP203911	Users who currently have the WordWhizzle Themes app installed on their mobile device or tablet
APP205410	Users who currently have the Hidden City¬Æ: Hidden Object Adventure app installed on their mobile device or tablet
APP205601	Users who currently have the WordBlobs app installed on their mobile device or tablet

- Collecting more records with the target game installed is a crucial step to improve the performance of the model. (more balanced data)
- If internal installation/ purchase data is available, the RFM (Recency, Frequency, Monetary value) customer segmentation can be further conducted to identify high, medium, and low values customers.
- Individual model can be further built for each tier of customer. More precise strategy can then be implemented on customers identified by each model.

* The full list of features used by the model is listed in the appendix

5 Suggestions & Next Steps

Ways to further improve the model:

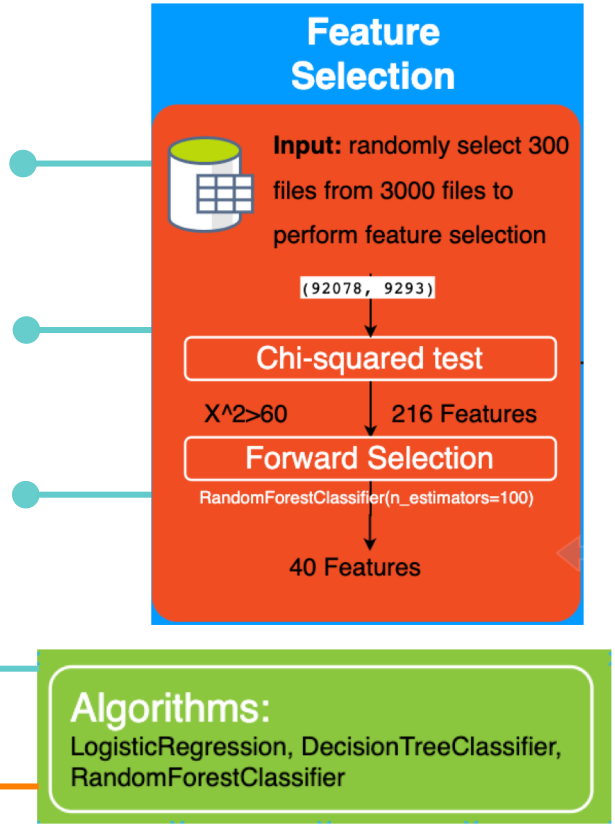
Use all 3000 files to perform feature selection/
Do bootstrapping when randomly selecting files.

Try Mutual Information Feature Selection.

Extract More features. Ex: 100 Features.
Try using 10, 20, ...100 features to fit the model.

Penalized-SVM increase the cost of classification mistakes on the minority class.

Instead of the classification approach, try calculating the possibility of each customer installing the app.



* Suggestions in orange are provided by Will


THANK YOU!
Puhsin Huang
TRUEDATA

APPENDIX

Presented by: Puhsin Huang

SIMPLIFY GROWTH WITH TRUEDATA

APPENDIX

Feature Used

Column ID	Column Description
APP201882	Users who currently have the FreeCell Solitaire app installed on their mobile device or tablet
APP201971	Users who currently have the Charm King,Ñ¢ app installed on their mobile device or tablet
APP203911	Users who currently have the WordWhizzle Themes app installed on their mobile device or tablet
APP205410	Users who currently have the Hidden City¬Æ: Hidden Object Adventure app installed on their mobile device or tablet
APP205601	Users who currently have the WordBlobs app installed on their mobile device or tablet
APP207449	Users who currently have the Wooden Block Puzzle Extreme app installed on their mobile device or tablet
APP207825	Users who currently have the Word Search Puzzles app installed on their mobile device or tablet
APP208412	Users who currently have the Pyramid Solitaire app installed on their mobile device or tablet
BEST_MASSAGE_INTERNATIONAL_90	Users who have visited a/an BEST_MASSAGE_INTERNATIONAL_90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
ATLANTA_CUSTOM_SHIRT_90	Users who have visited a/an ATLANTA_CUSTOM_SHIRT_90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
CAVE_CREEK_GOLF_COURSE_180	Users who have visited a/an CAVE_CREEK_GOLF_COURSE_180 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
CORNER_BAKERY_CAFE_90	Users who have visited a/an CORNER_BAKERY_CAFE_90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
EASTMORELAND_GOLF_COURSE_90	Users who have visited a/an EASTMORELAND_GOLF_COURSE_90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
DELAWARE_CANAL_STATE_PARK_30	Users who have visited a/an DELAWARE_CANAL_STATE_PARK_30 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
CHARLOTTE_COUNTRY_CLUB_90	Users who have visited a/an CHARLOTTE_COUNTRY_CLUB_90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
EARTHBOUND_TRADING_CO__90	Users who have visited a/an EARTHBOUND_TRADING_CO__90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}

APPENDIX

Feature Used

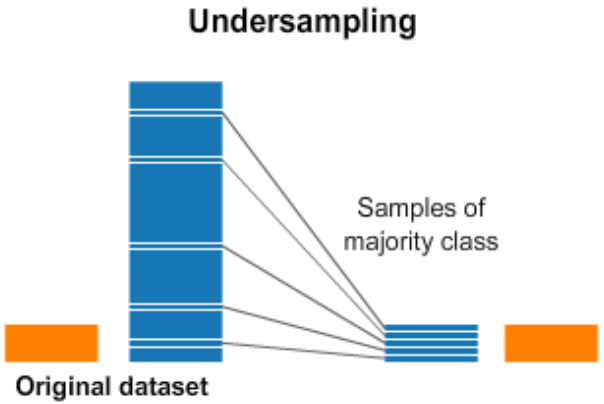
Column ID	Column Description
DALLAS_NATIONAL_GOLF_CLUB_180	Users who have visited a/an DALLAS_NATIONAL_GOLF_CLUB_180 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
FRESHII_30	Users who have visited a/an FRESHII_30 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
FOUR_WINDS_CARRIAGE_COMPANY_30	Users who have visited a/an FOUR_WINDS_CARRIAGE_COMPANY_30 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
GEORGIA_CAPITOL_MUSEUM_90	Users who have visited a/an GEORGIA_CAPITOL_MUSEUM_90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
GOODWILL_INDUSTRIES_90	Users who have visited a/an GOODWILL_INDUSTRIES_90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
HRD100031	Users who have a/an Motorola device
HRD100101	Users who have a/an iPad device
JACKSON_HEWITT_TAX_SERVICE_90	Users who have visited a/an JACKSON_HEWITT_TAX_SERVICE_90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
MRG100044	Users who currently have Maps & Navigation apps installed on their mobile device or tablet
PILOT_FLYING_J_180	Users who have visited a/an PILOT_FLYING_J_180 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
SPEEDWAY_180	Users who have visited a/an SPEEDWAY_180 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
SUNOCO_180	Users who have visited a/an SUNOCO_180 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
TARGET_90	Users who have visited a/an TARGET_90 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}
TD_BANK_30	Users who have visited a/an TD_BANK_30 in the last x days, where x is based on suffix values: {30: 0-30 days, 90: 31-90 days, 180: 91-180 days}

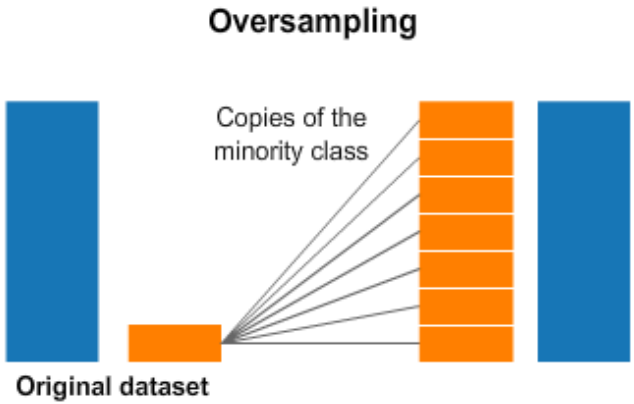
- Noted that fields 'PRD000006','STATE_GA','STATE_VA','STATE_WA', 'STATE_OR', 'STATE_NE', 'STATE_SC', 'STATE_AR','STATE_NV', 'STATE_MS', 'SUNOCO_180' are not listed in the variable dictionary so the meanings are unknown.

Models performance with different parameters combination (cv=3)

		Random Under Sampling			Data without preprocessing		
	model	recall	precision	accuracy	recall	precision	accuracy
0	LogisticRegression()	0.433	0.21	0.873	0.04688771	0.57117232	0.94287099
1	DecisionTreeClassifier()	0.43	0.213	0.873	0.05156019	0.59138755	0.94311388
2	RandomForestClassifier(max_depth=2, random_state=42)	0.38	0.22	0.88	0	0	0.94219259
3	RandomForestClassifier(max_depth=2, n_estimators=200, random_state=42)	0.383	0.217	0.883	0	0	0.94219259
4	RandomForestClassifier(max_depth=2, n_estimators=500, random_state=42)	0.38	0.22	0.883	0	0	0.94219259
5	RandomForestClassifier(max_depth=2, n_estimators=1000, random_state=42)	0.377	0.22	0.883	0	0	0.94219259
6	RandomForestClassifier(max_depth=3, random_state=42)	0.38	0.22	0.88	3.62E-05	0.66666667	0.94219469
7	RandomForestClassifier(max_depth=3, n_estimators=200, random_state=42)	0.387	0.213	0.88	3.62E-05	0.66666667	0.94219469
8	RandomForestClassifier(max_depth=3, n_estimators=500, random_state=42)	0.387	0.22	0.883	9.06E-05	1	0.94219783
9	RandomForestClassifier(max_depth=3, n_estimators=1000, random_state=42)	0.387	0.217	0.88	0.00012677	1	0.94219992
10	RandomForestClassifier(max_depth=4, random_state=42)	0.387	0.22	0.88	0.00036221	0.95833333	0.94221248
11	RandomForestClassifier(max_depth=4, n_estimators=200, random_state=42)	0.397	0.217	0.88	0.00036221	0.96296296	0.94221248
12	RandomForestClassifier(max_depth=4, n_estimators=500, random_state=42)	0.393	0.217	0.883	0.00077874	0.96969697	0.94223656
13	RandomForestClassifier(max_depth=4, n_estimators=1000, random_state=42)	0.393	0.217	0.883	0.00130394	0.95713634	0.94226378

Undersampling / Oversampling





Advantages

•It can help improve **run time and storage problems** by reducing the number of training data samples when the training data set is huge.

Disadvantages

•It can **discard potentially useful information** which could be important for building rule classifiers.

•The sample chosen by random under-sampling may be a **biased sample.**

Advantages

•Unlike under-sampling, this method leads to **no information loss.**

•Outperforms under sampling

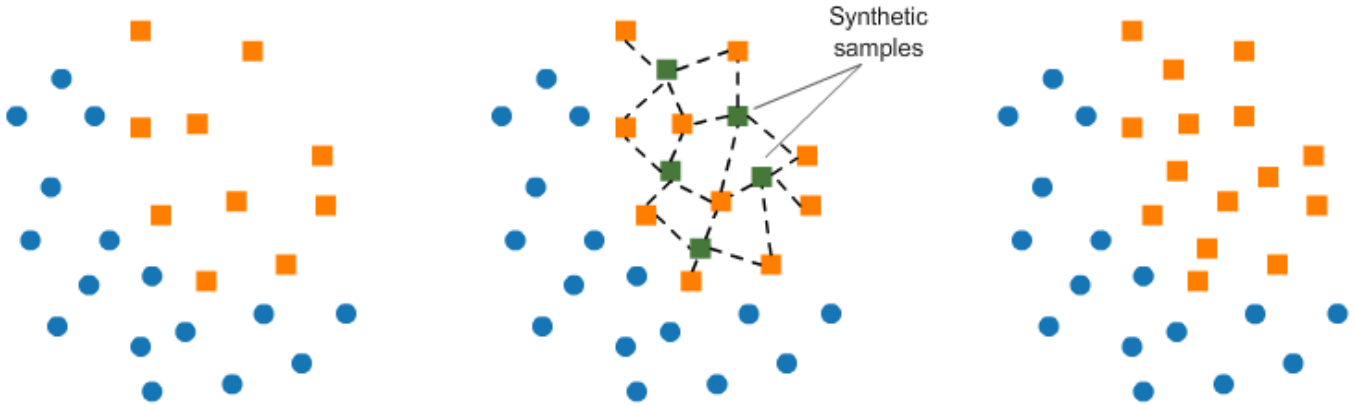
Disadvantages

•It increases the likelihood of **overfitting** since it replicates the minority class events.

APPENDIX

SMOTE

SMOTE (Synthetic Minority Oversampling Technique) works by randomly picking a point from the minority class and computing the k-nearest neighbors for this point. The synthetic points are added between the chosen point and its neighbors.



APPENDIX

Penalize Algorithm

Penalized-SVM **increase the cost of classification mistakes on the minority class.**
During training, I use the argument **class_weight='balanced'** to penalize mistakes on the minority class by an amount proportional to how under-represented it is.

```
# load library
from sklearn.svm import SVC


# we can add class_weight='balanced' to add panalize mistake
svc_model = SVC(class_weight='balanced', probability=True)


svc_model.fit(x_train, y_train)


svc_predict = svc_model.predict(x_test)# check performance
```

APPENDIX

Confusion Matrix

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

	Pred non-Install (0)	Pred install (1)
Real non-install (0)	TN	FN
Real install (1)	FP	TP