Do It Skills
do it. enjoy it.

SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

IMBALANCED DATSET

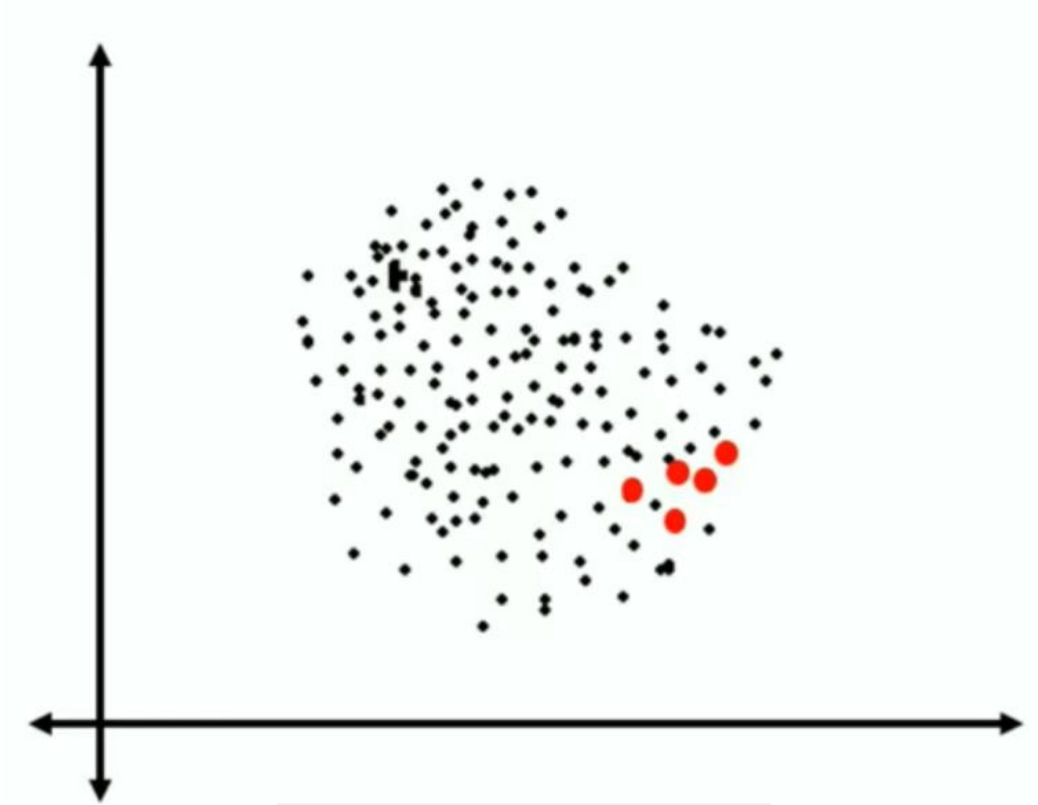
- This dataset is **unbalanced.**

```
data.head()
```

	buying	maint	doors	persons	lug_boot	safety	outcome
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

Before SMOTE : Counter({'unacc': 839, 'acc': 282, 'good': 48, 'vgood': 40})

IMBALANCED DATSET



Two Class Classification

No-Fraud → 99.5%
Fraud → 0.5%

- Presence of minority class in the dataset
- Challenges related Imbalanced Dataset
  - Biased predictions
  - Misleading accuracy
- Some Examples
  - Credit card frauds
  - Manufacturing defects
  - Rare diseases diagnosis
  - Natural disasters
  - Enrolment to premier institutes

HOW TO SOLVE THE PROBLEM?

- Balance the classes by Increasing minority or decreasing majority

- Random Under-Sampling
  - Randomly remove majority class observations
  - Helps balance the dataset
  - Discarded observations could have important information
  - May lead to bias

Total Observations = 1,000
Fraudulent = 10 or 1%
Normal = 990 or 99%

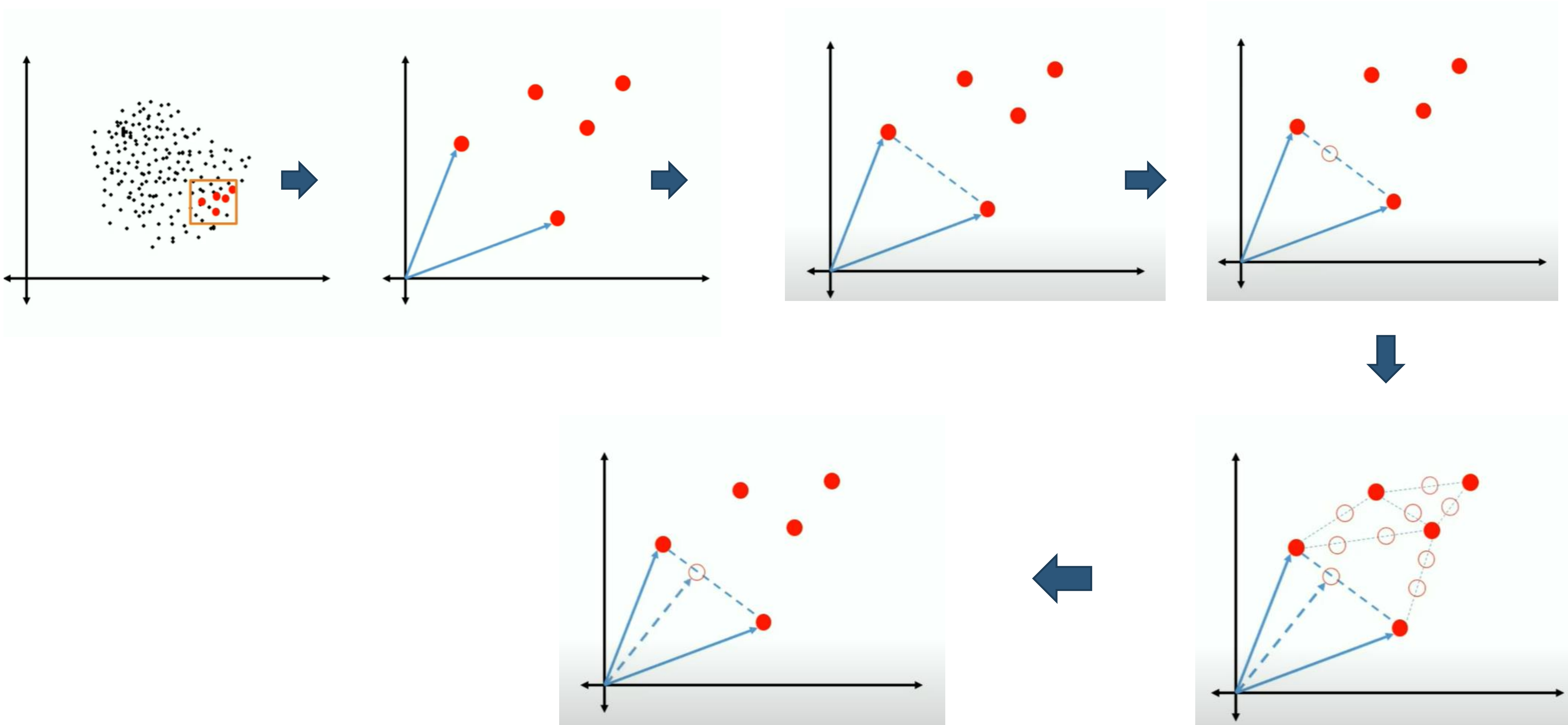
Reduce normal to 90
Fraudulent = 10 or 10%

- Random Over-Sampling
  - Randomly add more minority observations by replication
  - No information loss
  - Prone to overfitting due to copying same information

Total Observations = 1,000
Fraudulent = 10 or 1%
Normal = 990 or 99%

Increase fraudulent by 100
Fraudulent 110 or 10%

SMOTE



HOW TO SOLVE THE PROBLEM?

- Synthetic Minority Oversampling Technique
- Creates new “Synthetic” observations
- SMOTE Process
  - Identify the feature vector and its nearest neighbour
  - Take the difference between the two
  - Multiply the difference with a random number between 0 and 1
  - Identify a new point on the line segment by adding the random number to feature vector
  - Repeat the process for identified feature vectors

HOW TO SOLVE THE PROBLEM?

x belongs to A

- **Step 1:** Setting the minority class set **A**, for each $x \in A$, the **k-nearest neighbors of x** are obtained by calculating the **Euclidean distance** between **x** and every other sample in set **A**.
- **Step 2:** The sampling rate **N** is set according to the imbalanced proportion. For each $x \in A$, **N** examples (i.e x1, x2, ...xn) are randomly selected from its k-nearest neighbors, and they construct the set A_1 .

x belongs to A

- **Step 3:** For each example $x_k \in A_1$ (k=1, 2, 3...N), the following formula is used to generate a new example:

$$x' = x + rand(0,1) * \mid x - x_k \mid$$

in which rand(0, 1) represents the random number between 0 and 1.

SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

- **Imbalanced classification** involves developing predictive models on classification datasets that have a **severe** class imbalance.
- The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class, although typically it is **performance on the minority class that is most important.**
- One way to solve this problem is to oversample the examples in the minority class.
- The simplest approach involves **duplicating examples in the minority class**, **although these examples don't add any new information to the model.**
- This can balance the class distribution but does not provide any additional information to the model.
- An improvement on duplicating examples from the minority class is to **synthesize new examples from the minority class.**
- This is a type of **data augmentation** for **tabular data** and can be very **effective** and is referred to as the **Synthetic Minority Oversampling Technique** or **SMOTE** for short.

SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

- SMOTE works by selecting examples that are close in the **feature space**, **drawing a line between the examples** in the feature space and **drawing a new sample at a point along that line.**
- Specifically, a **random example** from the **minority class is first chosen.**
- Then **k of the nearest neighbors** for that example are found (typically k = 5).
- A **randomly selected neighbor** is **chosen** and a **synthetic example** is created at a **randomly selected point between the two examples** in feature space.

BALANCED DATSET

- This dataset is **balanced.**

```
data.head()
```

	buying	maint	doors	persons	lug_boot	safety	outcome
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

Before SMOTE : Counter({'unacc': 839, 'acc': 282, 'good': 48, 'vgood': 40})
After SMOTE : Counter({'acc': 839, 'unacc': 839, 'vgood': 839, 'good': 839})

Do It Skills
do it. enjoy it.

Do It Skills
do it. enjoy it.

THANK YOU

ARUNKG99@GMAIL.COM

WWW.DOITSKILLS.COM