Train In Data

SMOTE-NC

SMOTE-NC

- **SMOTE - Nominal Continuous.**

- Extends the functionality of SMOTE to categorical variables

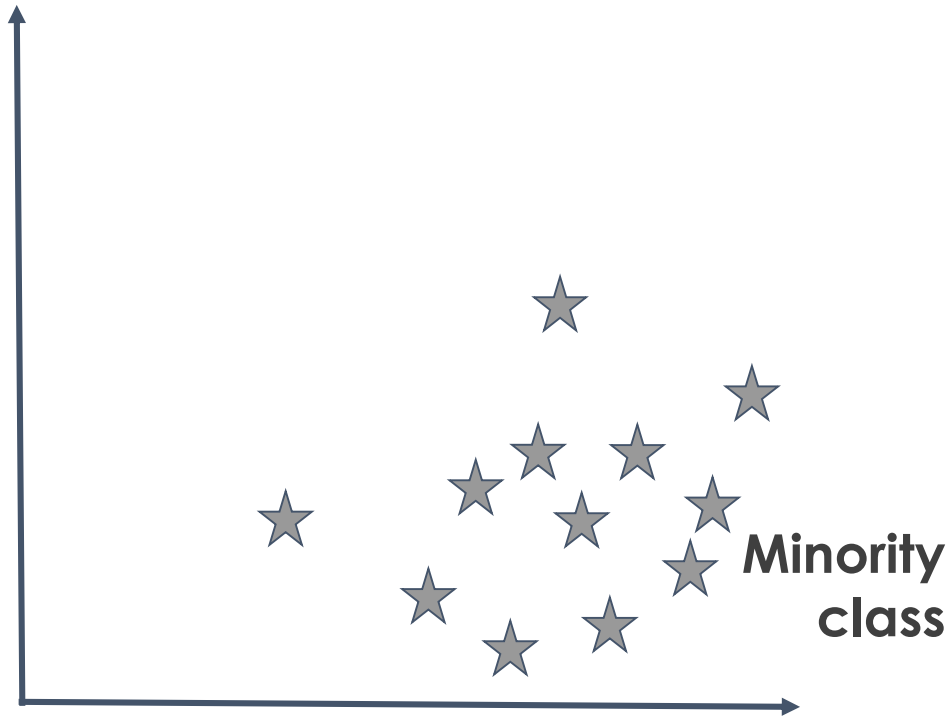
SMOTE-NC

Random Over-sampling can work with categorical data

SMOTE, its variants and ADASYN can't.

SMOTE-NC is an extension of SMOTE that makes it possible to create synthetic data from variables that are not numerical

SMOTE: how it works



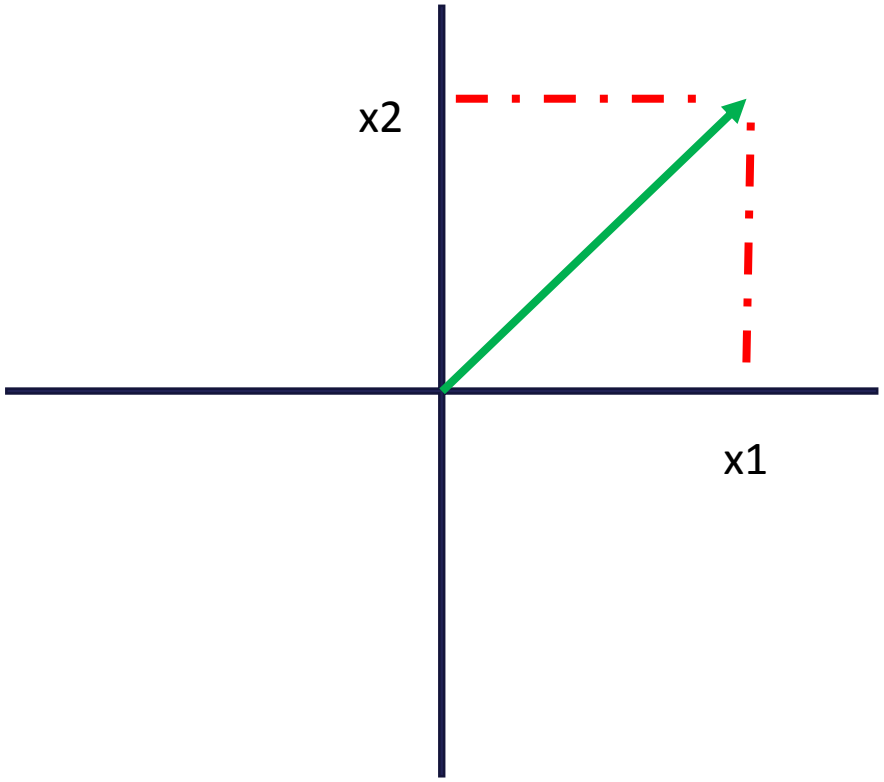
Looks only at the observations from the minority class.

Finds its k nearest neighbours

Typically k is 5

How can we calculate the distances with categorical values?

Euclidean distance, l2



$$L2 = \sqrt{x1^2 + x2^2}$$

SMOTE-NC: how it works

Area	Price	Engine	Colour	Gender
100	100	200	Black	F
90	90	200	Black	F
80	50	150	Black	M
80	60	220	Red	M
100	3	300	Green	M
70	120	450	Blue	M
55	200	450	Yellow	F

Calculate the Euclidean distances to find the K neighbours

$$L2 = \sqrt{x1^2 + x2^2 + x3^2 + x4^2 + x5^2}$$

SMOTE-NC: how it works

Calculate the Euclidean distances to find the K neighbours

Area	Price	Engine	Colour	Gender
100	100	200	Black	F
90	90	200	Black	F
80	50	150	Black	M
80	60	220	Red	M
100	3	300	Green	M
70	120	450	Blue	M
55	200	450	Yellow	F

16.29	62.13	123.48	→	62.13

Median (STDEVs)

Standard deviations

(minority class only)

$$L2 = \sqrt{x1^2 + x2^2 + x3^2 + x4^2 + x5^2}$$

SMOTE-NC: how it works

Calculate the Euclidean distances to find the K neighbours

Area	Price	Engine	Colour	Gender
100	100	200	Black	F
90	90	200	Black	F
80	50	150	Black	M
80	60	220	Red	M
100	3	300	Green	M
70	120	450	Blue	M
55	200	450	Yellow	F

16.29	62.13	123.48	→	62.13

Median (STDEVs)

$$L2 = \sqrt{x1^2 + x2^2 + x3^2 + x4^2 + x5^2}$$

Standard deviations

$$L2 = \sqrt{(100-80)^2 + (100-50)^2 + (200-150)^2 + (0)^2 + (62.13)^2}$$

SMOTE-NC: how it works

Calculate the Euclidean distances to find the K neighbours

Area	Price	Engine	Colour	Gender
100	100	200	Black	F
90	90	200	Black	F
80	50	150	Black	M
80	60	220	Red	M
100	3	300	Green	M
70	120	450	Blue	M
55	200	450	Yellow	F

16.29	62.13	123.48	→	62.13

Median (STDEVs)

$$L2 = \sqrt{x1^2 + x2^2 + x3^2 + x4^2 + x5^2}$$

Standard deviations

$$L2 = \sqrt{(80-55)^2 + (60-200)^2 + (220-450)^2 + (62.13)^2 + (62.13)^2}$$

SMOTE-NC: how it works

Calculate the Euclidean distances to find the K neighbours

Area	Price	Engine	Colour	Gender
100	100	200	Black	F
90	90	200	Black	F
80	50	150	Black	M
80	60	220	Red	M
100	3	300	Green	M
70	120	450	Blue	M
55	200	450	Yellow	F

16.29	62.13	123.48	→	62.13

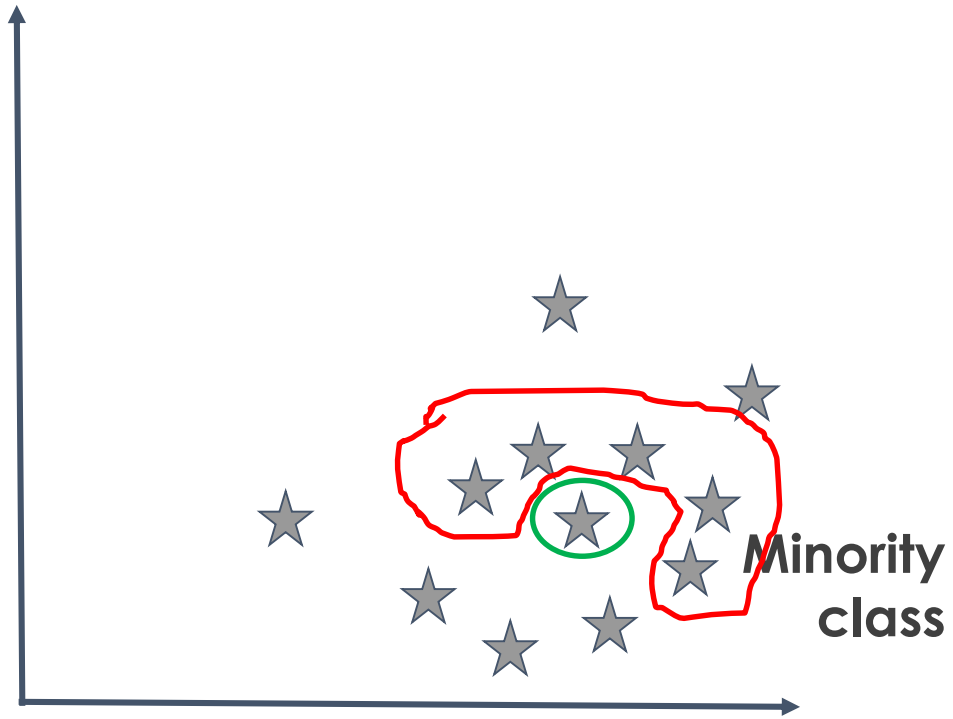
Median (STDEVs)

$$L2 = \sqrt{x1^2 + x2^2 + x3^2 + x4^2 + x5^2}$$

Standard deviations

$$L2 = \sqrt{(100-90)^2 + (100-90)^2 + (200-200)^2 + (0)^2 + (0)^2}$$

SMOTE-NC: how it works



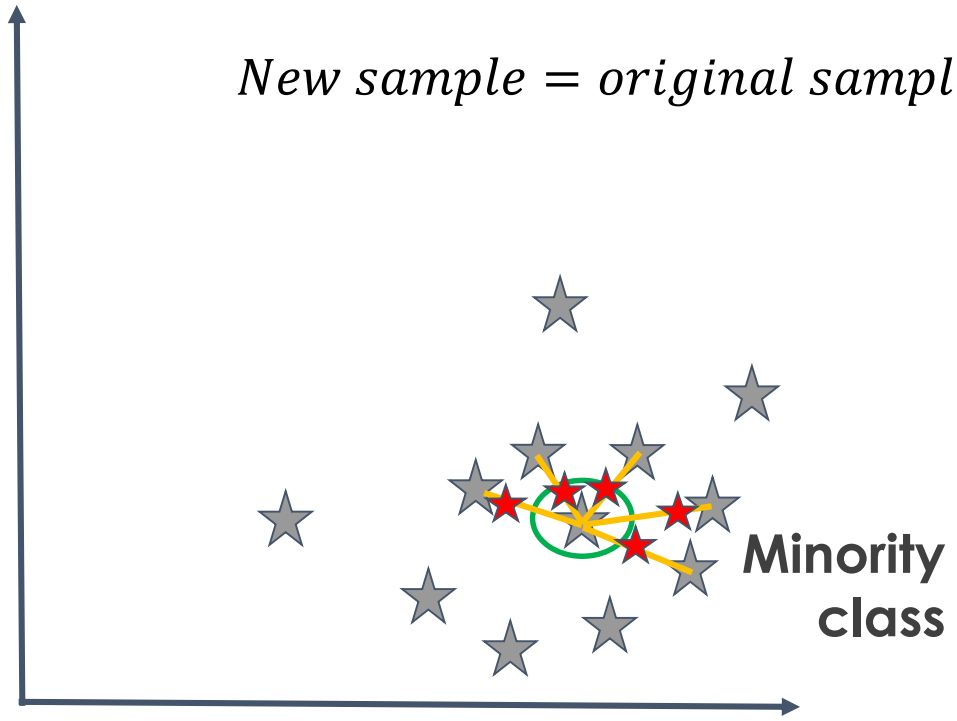
With the Euclidean distances, we can train a KNN.

And finds the k nearest neighbours of each observation

Typically k is 5

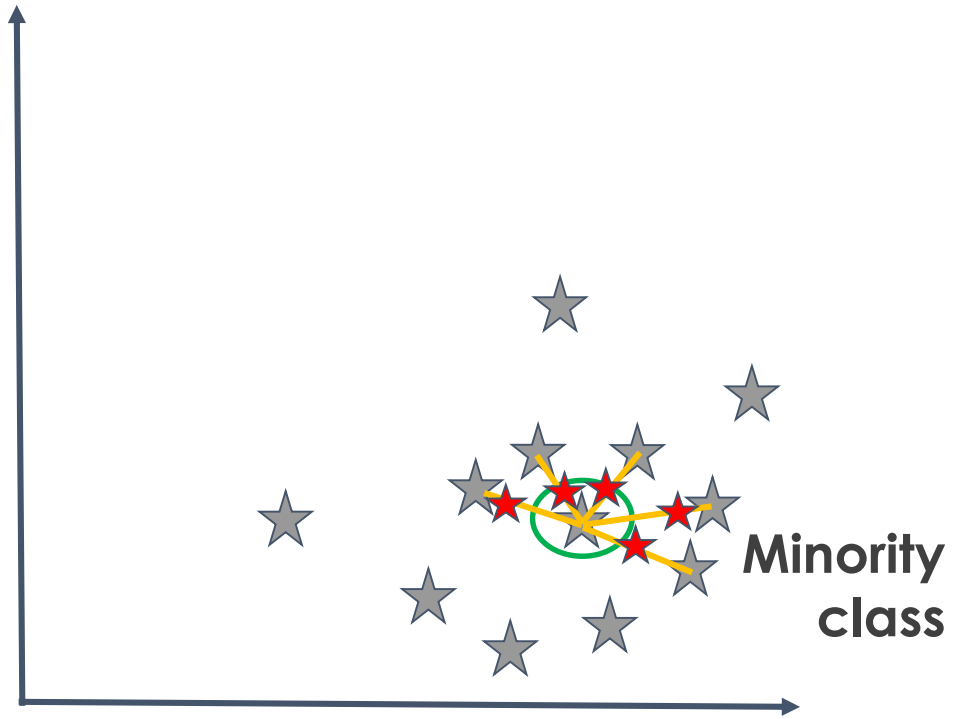
SMOTE-NC: numerical values

$$New\ sample = original\ sample\ - factor\ * (original\ sample\ - neighbour)$$



Values of numerical variables are calculated as in SMOTE

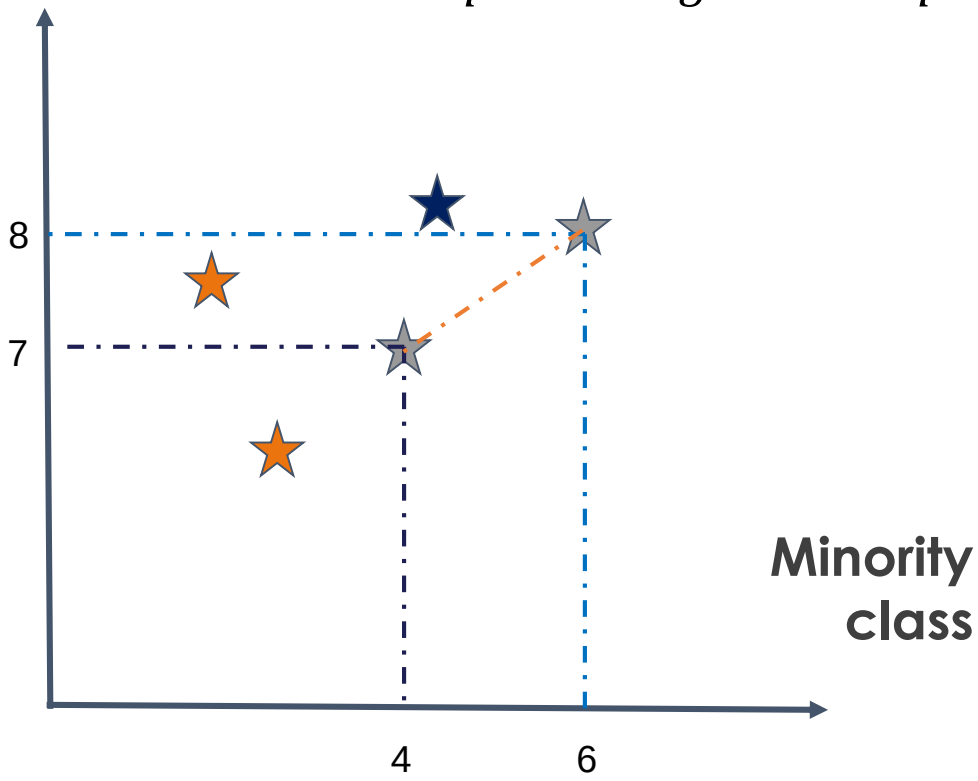
SMOTE-NC: categorical values



Values of categorical features are those shown by the majority of the neighbours

SMOTE: numerical example

$$New\ sample = original\ sample\ - factor\ * (original\ sample\ - neighbour)$$

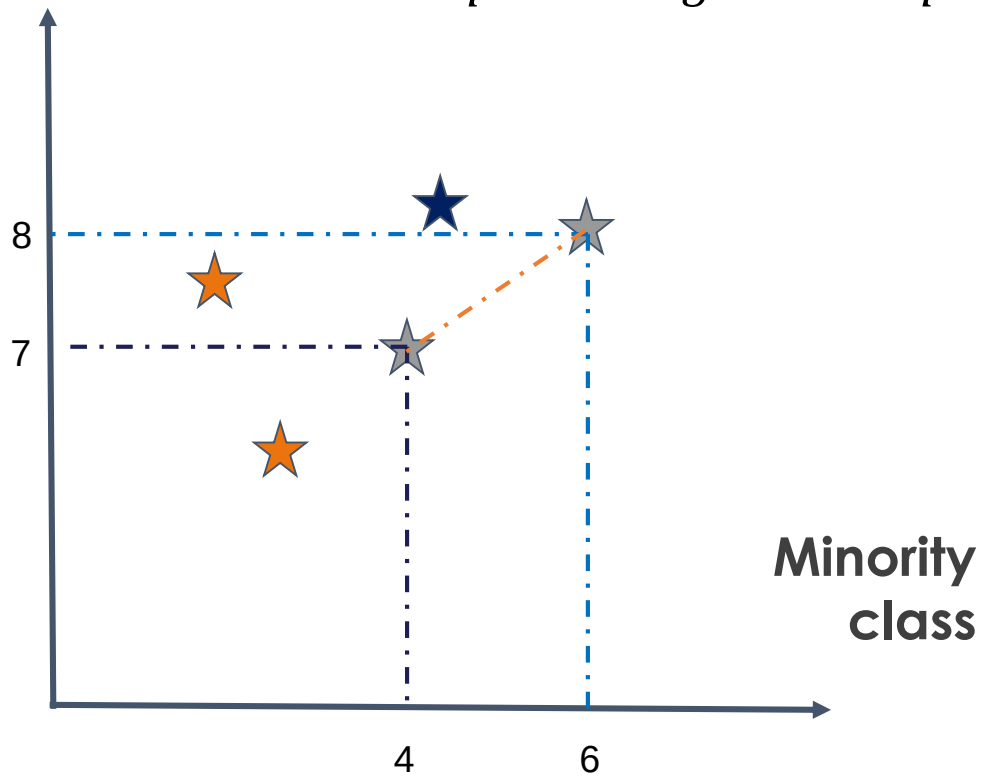


$X_{ori} = (4,7)$

$X_{neig} = (6, 8)$

SMOTE: numerical example

$$New\ sample = original\ sample - factor * (original\ sample - neighbour)$$



$X_{ori} = (4,7)$

$X_{neig} = (6, 8)$

$$New\ sample = (4,7) - 0.8 * ((4,7) - (6,8))$$

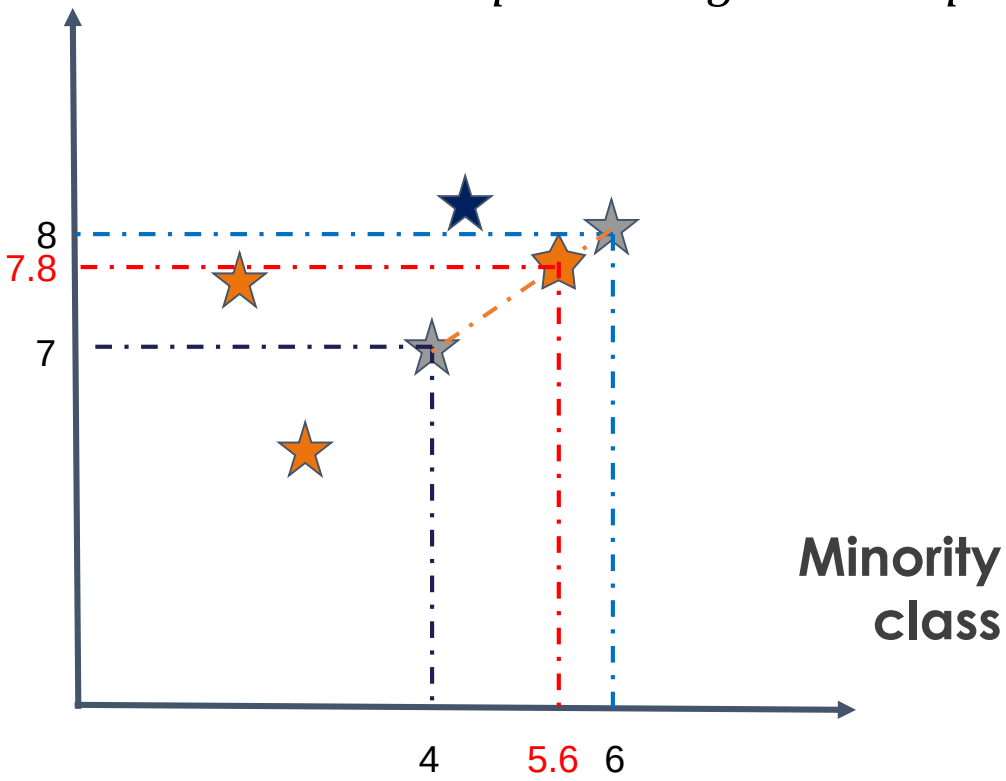
$$New\ sample = (4,7) - 0.8 * ((-2,-1))$$

$$New\ sample = (4,7) - ((-1.6, -0.8))$$

$$New\ sample = (5.6, 7.8)$$

SMOTE: numerical example

$$New\ sample = original\ sample - factor * (original\ sample - neighbour)$$



$X_{ori} = (4,7)$

$X_{neig} = (6, 8)$

$$New\ sample = (4,7) - 0.8 * ((4,7) - (6,8))$$

$$New\ sample = (4,7) - 0.8 * ((-2,-1))$$

$$New\ sample = (4,7) - ((-1.6, -0.8))$$

$$New\ sample = (5.6, 7.8)$$

Imbalanced-learn: SMOTE-NC

```
smnc = SMOTENC(
    sampling_strategy='auto', # samples only the minority class
    random_state=0,  # for reproducibility
    k_neighbors=5,
    n_jobs=4,
    categorical_features=[2,3] # indeces of the columns of categorical variables
)

X_res, y_res = smnc.fit_resample(X, y)
```

Train In Data

THANK YOU

www.trainindata.com