



ADASYN



ADASYN

- Uses a weighted distribution of the minority class according to how difficult the observations are to be learned or classified.
- More synthetic data is generated from samples that are harder to classify



ADASYN vs SMOTE

- **SMOTE:** uses all samples from the minority class to create the synthetic data.
- **ADASYN:** uses more samples that are harder to classify and less that are easy to classify to create the synthetic data



ADASYN vs SMOTE

- **SMOTE**: uses only minority class to train the KNN.
- **ADASYN**: uses all samples to train KNN

ADASYN vs SMOTE

- **SMOTE**: interpolates between samples of minority class.
- **ADASYN**: interpolates between sample of minority class and neighbour from minority or majority

ADASYN: how it works

- **Step 1:** determine the balancing ratio.

$$R(x) = \frac{X(\text{minority})}{X(\text{majority})}$$

ADASYN: how it works

- **Step 2:** determine the number of samples to generate.

$$G = (X(\text{majority}) - X(\text{minority})) * \text{factor}$$

- If $X_{\text{maj}} = 900$ and $X_{\text{min}} = 100 \rightarrow G = 800$
- Factor is 1 for full balancing, or balancing ratio of 1

ADASYN: how it works

- Step 3:

1. train KNN using **entire dataset** (not only minority class as SMOTE)
2. Find K closest neighbours for each sample of minority class
3. Determine the weighting r

$$r = \frac{D}{K}$$

D = neighbours from the majority class

K = neighbours

ADASYN: how it works

- **Step 4:** Normalise r

$$r_{norm} = \frac{r}{\text{sum}(rs)}$$

- Remember that there is 1 r value per observation of the minority class

ADASYN: how it works

- **Step 5:** Calculate the number of synthetic examples that need to be generated for each observation of the minority class

$$g_i = r_i \times G$$

r_i = weight for observation i

G = number of samples to generate

g_i = number of samples to generate from observation i

ADASYN: how it works

- **Step 6:** For each minority class example x_i , generate $g1$ synthetic samples

$$\text{New sample} = \text{minority sample} - \text{factor} * (\text{minority sample} - \text{neighbour})$$



The neighbour can be from the majority or the minority class.

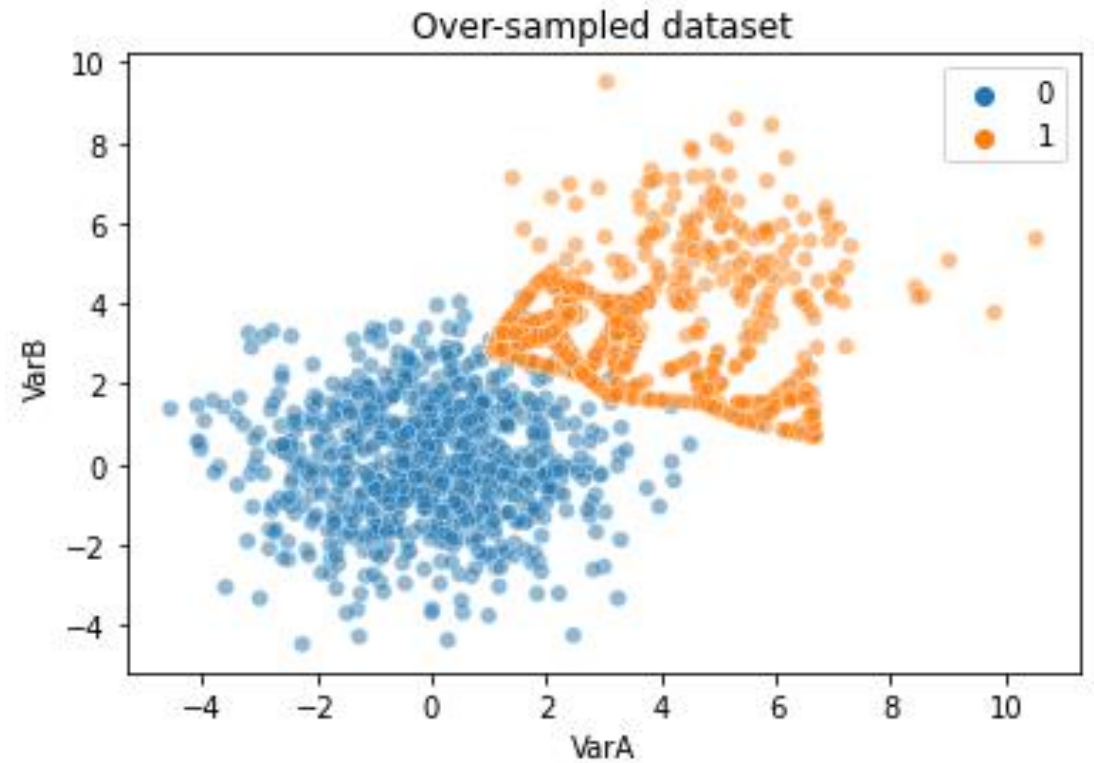
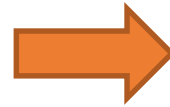
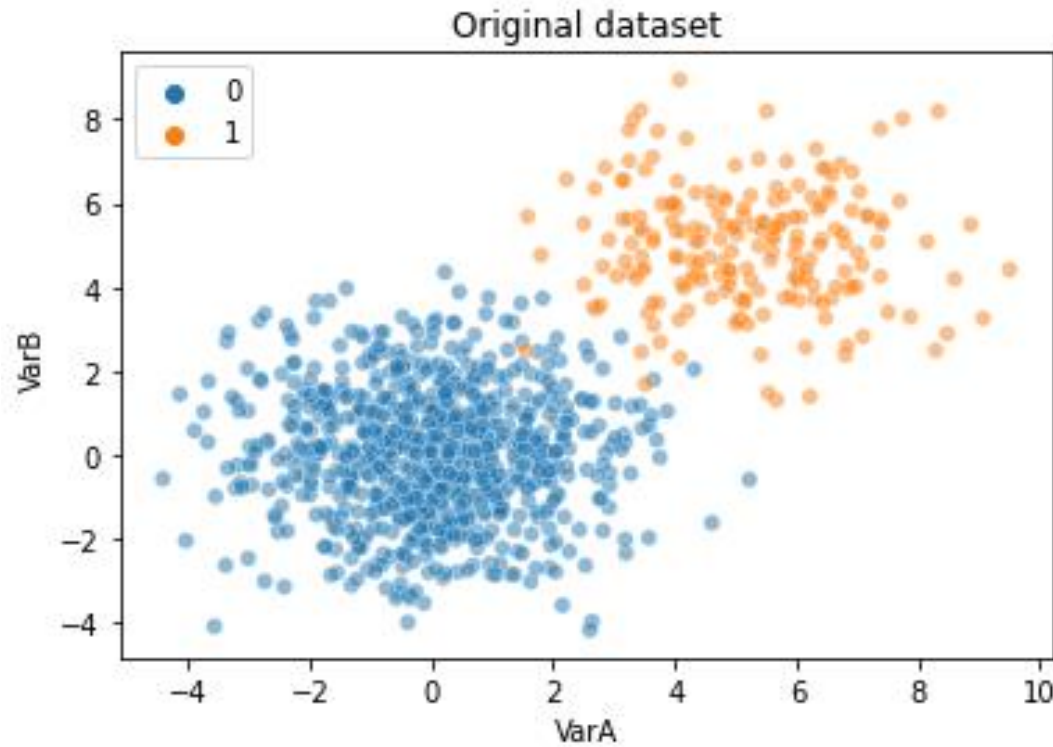
The KNN is trained on the entire dataset

Different from SMOTE!

Imbalanced-learn: ADASYN

```
▶ ada = ADASYN(  
    sampling_strategy='auto',  # samples only the minority class  
    random_state=0,  # for reproducibility  
    n_neighbors=5,  
    n_jobs=4  
)  
  
X_res, y_res = ada.fit_resample(X, y)
```

Imbalanced-learn: ADASYN



THANK YOU

www.trainindata.com