Sampling
&
Dimensionality
Reduction

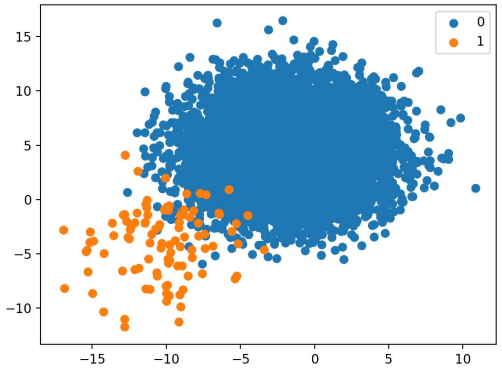
Agenda

- Data imbalance
- Exempel
- Mätvärden
- Sampling
  - Undersampling
  - Oversampling

- Dimensionality Reduction
  - Principal Component Analysis
  - Kernel PCA
  - Linear Discriminant Analysis

Data imbalance

- Obalans.
- Minoritet och majoritet.
- Binära eller multipla klasser.



*"An imbalanced classification problem is an example of a classification problem where the distribution of examples across the known classes is **biased** or **skewed**."*

- Jason Brownlee

- De flesta algoritmer antar att det inte finns några stora skillnader mellan storleken av klasserna.
- Detta antagande gör att modellen inte blir lika mycket tränad på minoritetsklassen. Detta är ett problem då det är ofta just den som vi är intresserade av.

Källa: https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/

Imbalance - Exempel

- E-mail och spam
- Kreditkortsbedrägerier
- Feldetektering av maskinkomponenter
- Detektering av nätverksfel

- Ofta många fler exempel av vanliga mail än spam. (1:9)
- Kreditkort - ännu större skillnad (50:1)
- Nätverksfel
  - Oväntat beteende som leder till lägre effektivitet och resurstillgänglighet.
  - Vill filtrera/välja ut de mest brådskande larmen.
  - Vill hitta orsaken och själva felet så fort som möjligt.
  - AI kan hjälpa till att hitta, eller t.o.m. förutspå, felen tidigare.

Källa: https://www.aismartz.com/blog/network-fault-detection-using-machine-learning/

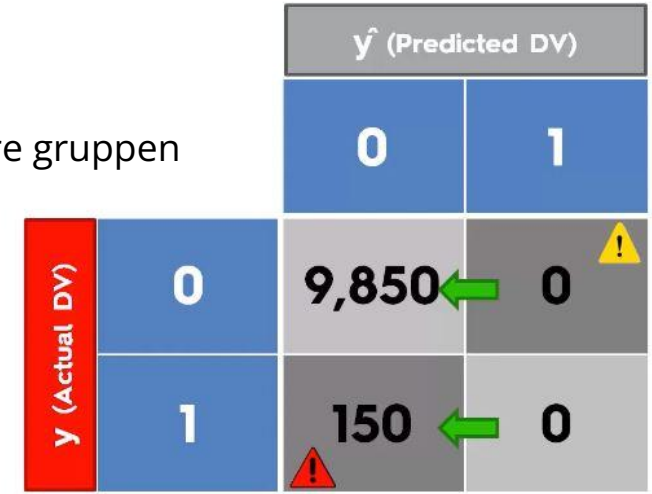
Mätvärden - Accuracy Paradoxen

Bias i data

Klassificerar alla i den större gruppen

Hur kan vi se detta?
- `classification_report`

Fyra värden:
- Precision
- Recall
- F1
- Support



Scenario 1:

Accuracy Rate = Correct / Total
AR = 9,800/10,000 = 98%

Scenario 2:

Accuracy Rate = Correct / Total
AR = 9,850/10,000 = 98.5%

- Macro average - Treat all classes equally. Average of individual class scores.
- Weighted average - Take frequency of the classes into consideration.

Mätvärden

- **Precision** score
  - Hur många “svar” av de som förutspåddes är relevanta (“true”).
    - $\frac{\text{Correct positive guesses}}{\text{Total positive guesses}}$

- **Recall**
  - Andel av alla relevanta “svar” som modellen förutspådde.
    - $\frac{\text{Correct positive cases}}{\text{All positive labels}}$

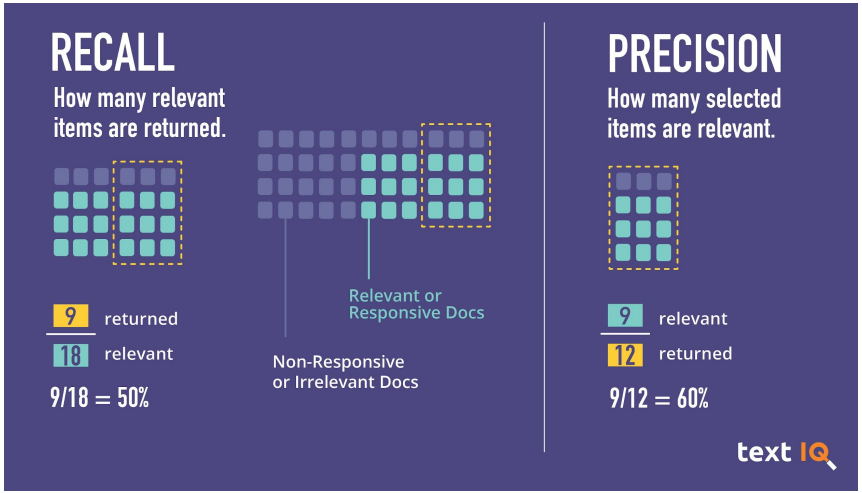
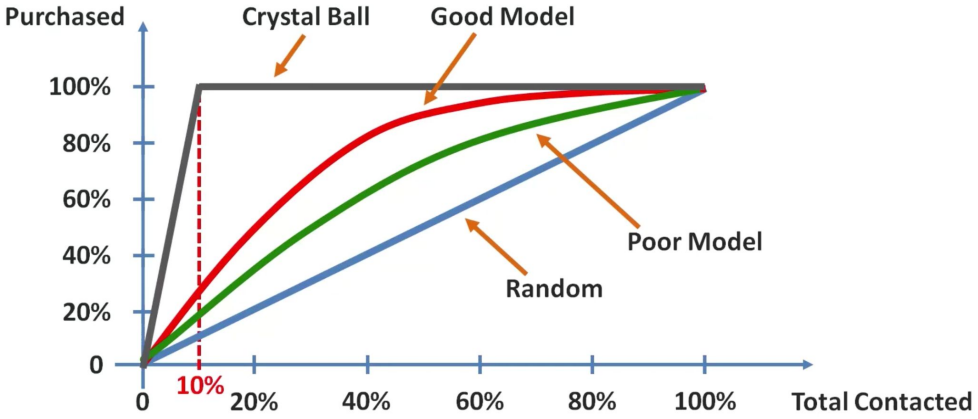


Bild ref. Measure ai

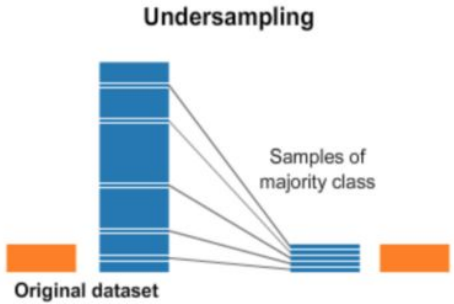
Mätvärden

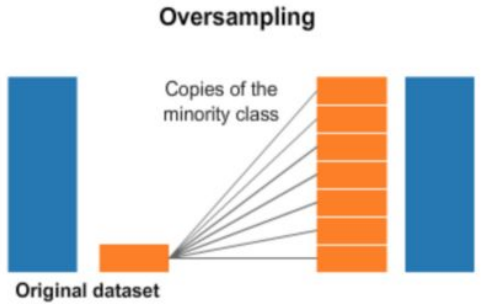
- **Precision** (noggrannhet) - tp/(tp+fp)
- **Recall** - tp/(tp+fn)
- **F-score** - Viktat medelvärde av precision och recall
- **CAP** - Cumulative Accuracy Profile



Sampling

Idén bakom sampling är att jämna ut denna obalans, vilket man gör på två olika sätt:





- Under Sampling
  - Slumpmässigt väljer ut samma antal exempel ur majoritetsklassen som vi har i minoritet.
- Over Sampling
  - Genererar nya exempel för minoritetsklassen med olika tekniker.
  - Duplicering av befintlig data.
  - Skapar nya som gränsar till befintlig data.

Dela alltid upp i tränings och test data först innan du samplar träningsdatan.

Bibliotek: `imblearn`

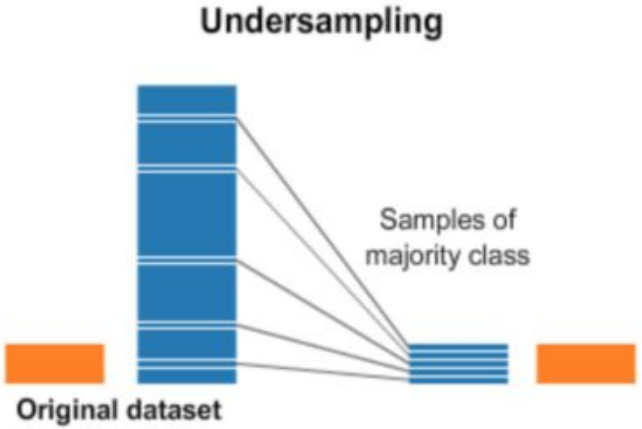
Källa: https://www.kdnuggets.com/2019/05/fix-unbalanced-dataset.html

Sampling - Undersampling

Eliminerar vissa exempel av majoritetsklassen från träningsdata.

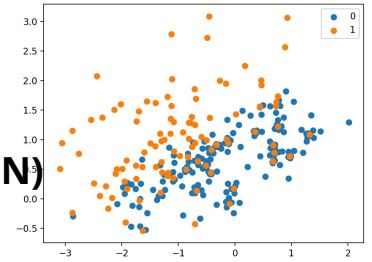
Random under sampling

- + Enkelt och effektivt.
- + All data är original.
- - Riskerar att förlora viktig information som kan avgöra gränsen mellan klasserna.
- - Ofta mycket mindre data överlag.



Källa: https://www.kdnuggets.com/2019/05/fix-unbalanced-dataset.html

Sampling - Undersampling



Near Miss

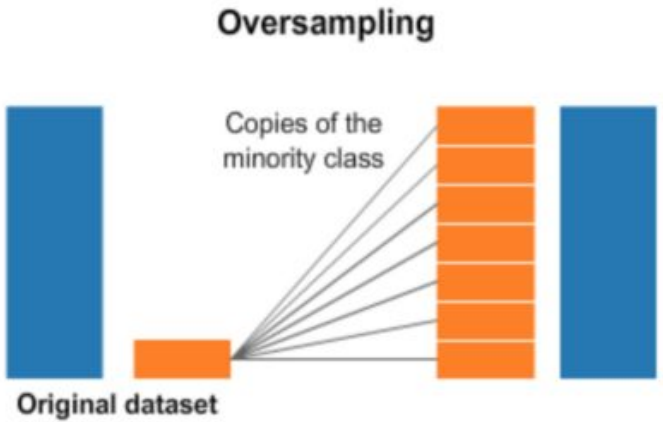
- Väljer datapunkter baserat på avståndet mellan majoritet och minoritet exempel.
- Tre versioner:
  1. Behåller exempel med det lägsta medelavståndet till närmaste minoritetsklassdatapunkt.
  2. Väljer exempel med det lägsta medelavstånd till minoritets- klassens mest avlägsna datapunkter.
  3. Behåller det närmaste exempel från majoritetsklassen för varje minoritetsklassdatapunkt.

Condensed Nearest Neighbor (CNN)

- Söker en delmängd som inte resulterar i någon förlust i modellprestanda.
- Man börjar med alla minoritetspunkter i ett “lager” och sedan används KNN algoritmen för att klassificera en punkt i taget från majoritetsklassen. Om en (majoritets) punkt inte kan klassificeras så läggs den till “lagret”.
- Skapar inte ett perfekt balanserat dataset.
- Fokuserar på gränsfallen.
- Kan vara långsam.

Sampling - Oversampling

- Fokuserar på att skapa nya punkter för minoritetsklassen för att öka dess storlek.
- Vill matcha majoritetsklassens storlek.
-
- Generera nya exempel med hjälp av en algoritm.

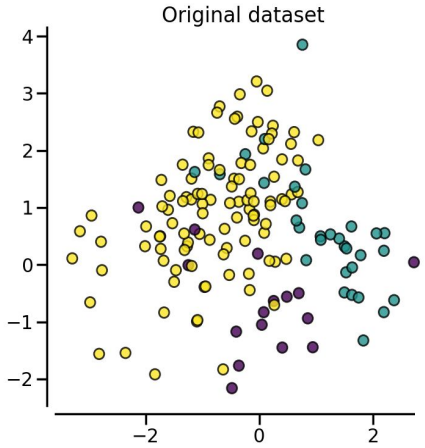


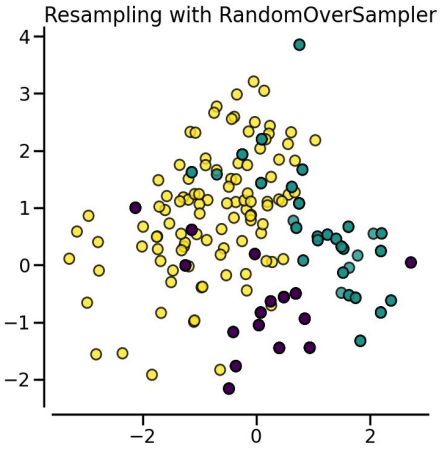
Källa: https://www.kdnuggets.com/2019/05/fix-unbalanced-dataset.html

OverSampling - Random

RandomOverSampler

- Duplicering av slumpmässiga punkter i minoritetsklassen.



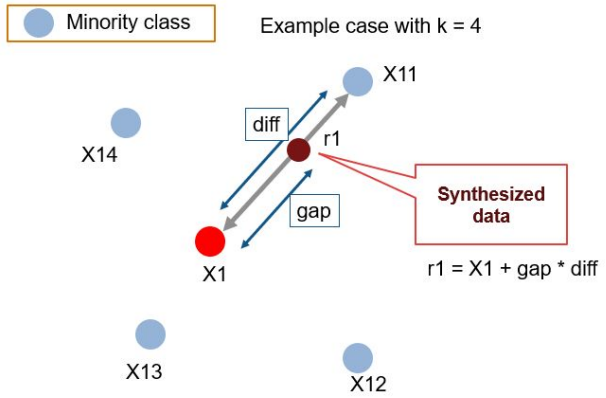


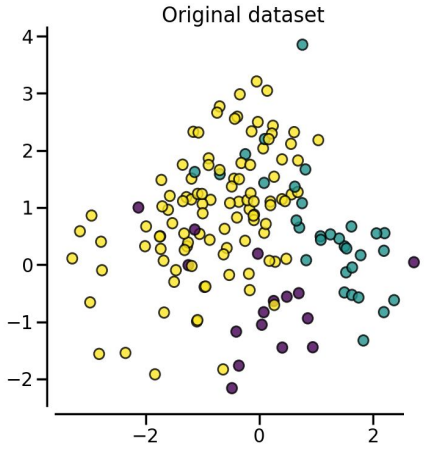
RESTORY...

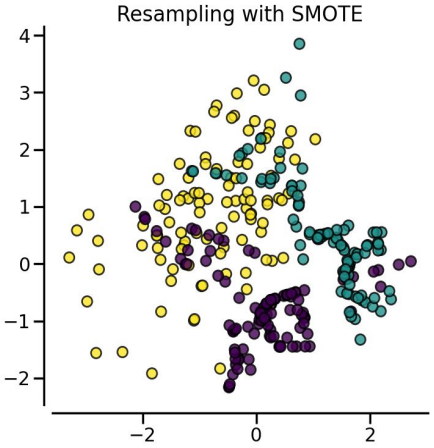
OverSampling - SMOTE

Synthetic Minority Oversampling Technique

- Skapar en ny minoritetspunkt mellan två andra med hjälp av interpolering.
- Lika men ej duplicerade.
- Kan leda till “noisy data”, d.v.s. skräpdata.





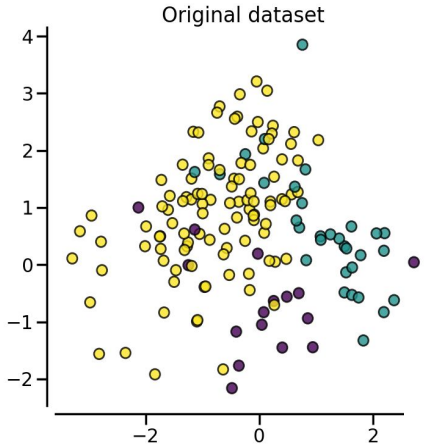


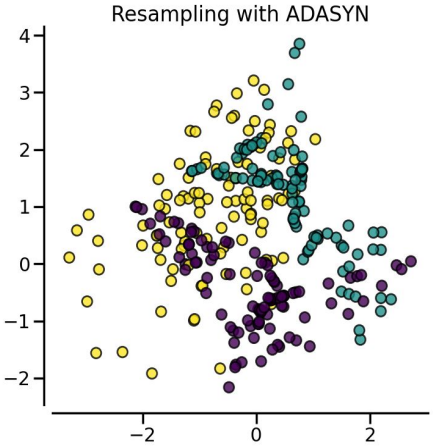
RESTORY...

Sampling - ADASYN

Adaptive Synthetic Sampling

- Skapar nya punkter inom minoritetsklassen bredvid exempel som blev felklassificerade när en KNN algoritm användes.
- Fokuserar på de svårare fallen. (outliers)

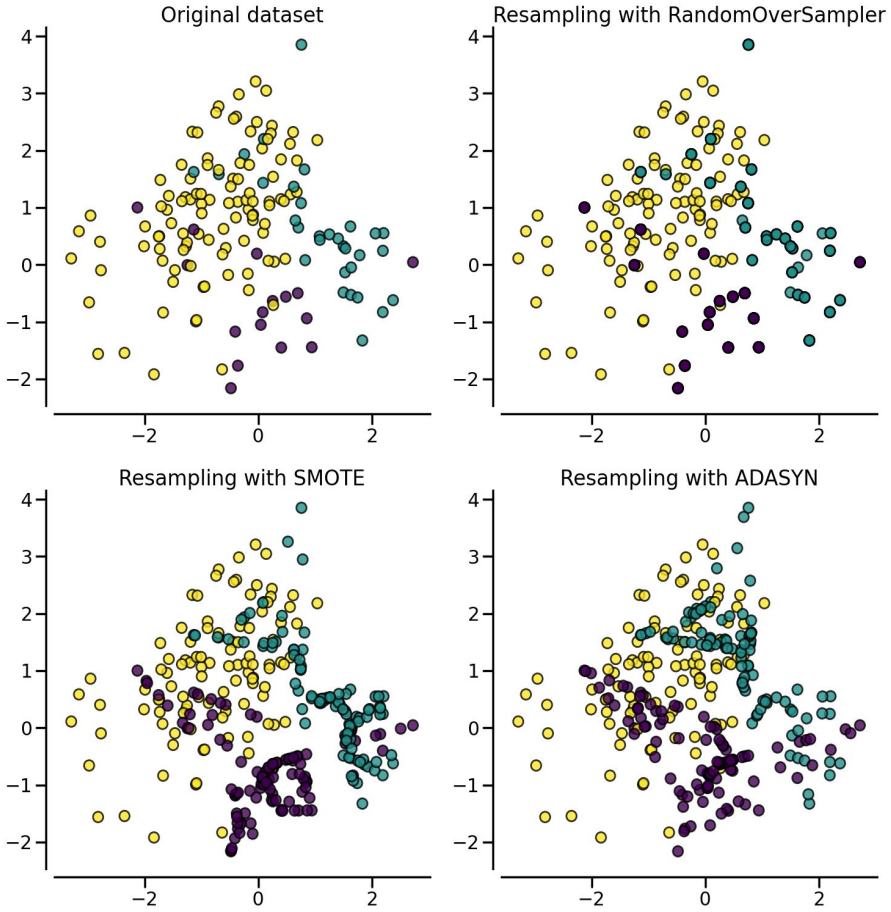




RESTORY...

Sammanfattning

- Finns fler
- Finns varianter
- Ofta kombination
  - med andra samplingsalgoritmer av samma slag
  - av under- och oversampling



Dimensionsreducering

PCA

- En av, om inte den, de mest använda oövervakade (unsupervised) algoritmer.
- Identifiera mönster i data.
- Upptäcka korrelationer mellan variabler (så att vi kan ta bort en - dimensionality reduction).
- Maximera variansen.
- Skapa nya “features”. Dessa kallas “principal components”
- Gör denna analys innan du tränar din modell.
- Fungerar med olika modeller.

- Kan välja hur många features vi vill ha ut
  - Börja med 2.
  - Öka tills du får bra resultat.
- Samma transformation måste göras på både tränings- och testdata.

- Används för:
  - Visualisering
  - Feature extraction
  - Noise filtering
- Inom:
  - Finans
  - Genanalys

PCA

- Standardize the data.
- Obtain the Eigenvectors and Eigenvalues from the covariance matrix or correlation matrix, or perform Singular Vector Decomposition.
- Sort eigenvalues in descending order and choose the k eigenvectors that correspond to the k largest eigenvalues where k is the number of dimensions of the new feature subspace ($k \leq d$)/.
- Construct the projection matrix $\mathbf{W}$ from the selected k eigenvectors.
- Transform the original dataset $\mathbf{X}$ via $\mathbf{W}$ to obtain a k-dimensional feature subspace $\mathbf{Y}$

https://plot.ly/ipython-notebooks/principal-component-analysis/

PCA kernel

- Båda importeras från `sklearn.decomposition`
  - `PCA`
  - `KernelPCA`
- En kernel, dvs en funktion, används för att göra delningen lättare.
- Precis som i SVM kernel.

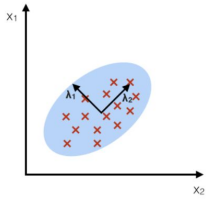
- Måste ange vilken kernel som parameter

LDA

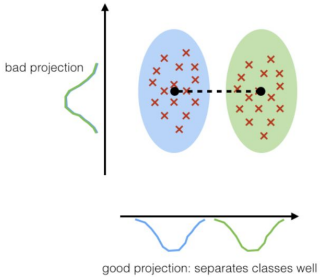
Linear Discriminant Analysis

- Upptäcka korrelationer mellan variabler
- Bibehålla information om klasskillnaderna.
- Övervakad (supervised) p.g.a. relationen till den beroende variabeln (y).

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



LDA

Summarizing the LDA approach in 5 steps

Listed below are the 5 general steps for performing a linear discriminant analysis; we will explore them in more detail in the following sections.

1. Compute the d-dimensional mean vectors for the different classes from the dataset.
2. Compute the scatter matrices (in-between-class and within-class scatter matrix).
3. Compute the eigenvectors ($\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_d$) and corresponding eigenvalues ($\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_d$) for the scatter matrices.
4. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix $\boldsymbol{W}$ (where every column represents an eigenvector).
5. Use this $d \times k$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: $\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{W}$ (where $\boldsymbol{X}$ is a $n \times d$-dimensional matrix representing the n samples, and $\boldsymbol{y}$ are the transformed $n \times k$-dimensional samples in the new subspace).

Källa:
https://sebastianraschka.com/Articles/2014_python_lda.html#principal-component-analysis-vs-linear-discriminant-analysis

Länkar

- Macro vs. weighted average
- Intro to imbalanced classification
- Overcoming imbalance
- Handling imbalance
- Analysis of imbalanced datasets
- Artikel: Oversampling in multi-class
- Multiclass with imbalanced dataset
- How to build a spam classifier
- PCA visualised
- LDA more info
- How to fix an Unbalanced dataset
- Classification accuracy for imbalanced class distributions
- Handle imbalanced data
- Undersampling algorithms
- SMOTE
- Practical - imbalanced data
- Why is data imbalance important
- What is synthetic data

Övningar

- Sampla Churn dataset - klassificera igen
- Gör en PCA och LDA på ett tidigare exempel av regression (titta på när vi jämförde dem för att se vilken som förbättrats mest)
- Gör en PCA och LDA på ett tidigare exempel av klassifikation.
- Installera imbalanced-learn:

```
pip install imbalanced-learn
```