Prediktiv analys

FÖRELÄSNING 8

Dagens fråga

- Vilken mytisk varelse önskar du fanns på riktigt?

Dagens agenda

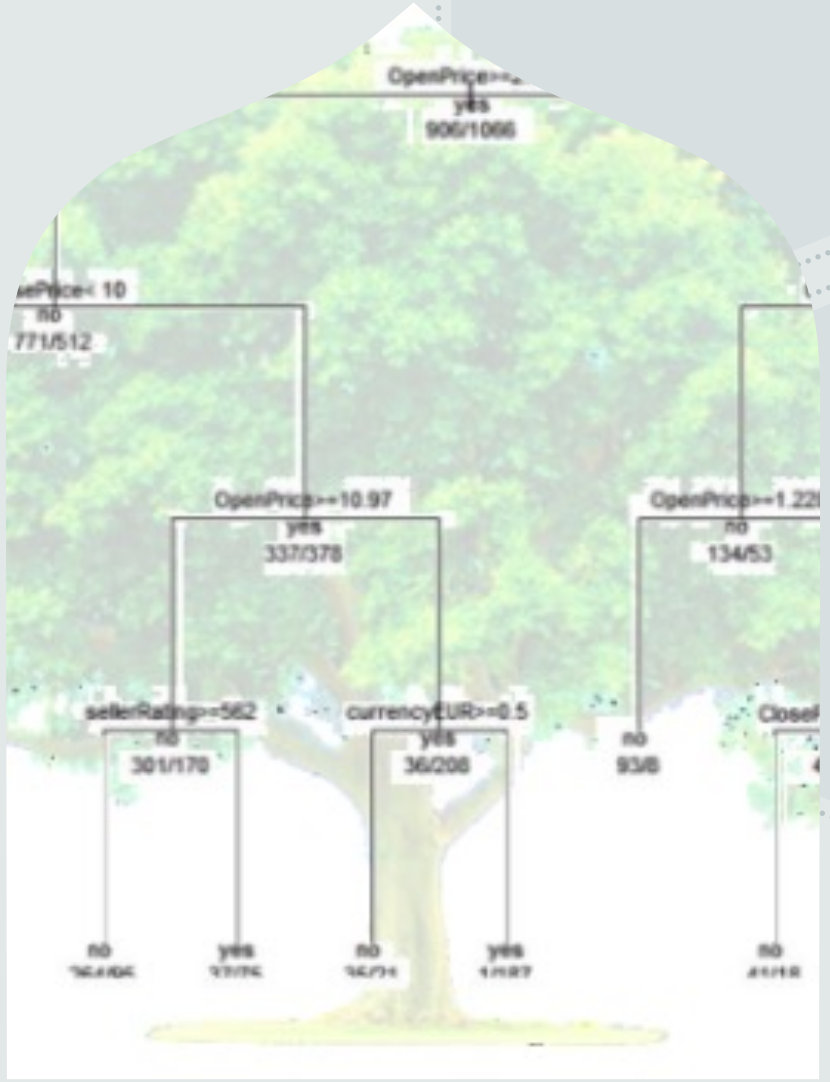
- Classification trees
- Rule inference
- Imbalanced data
- Naive Bayes

Förra föreläsning

- Gradient descent
- Evaluering av klassifikationsmodeller:
  - Error metrics
  - Confusion Matrix
  - Accuracy, precision, recall, F1

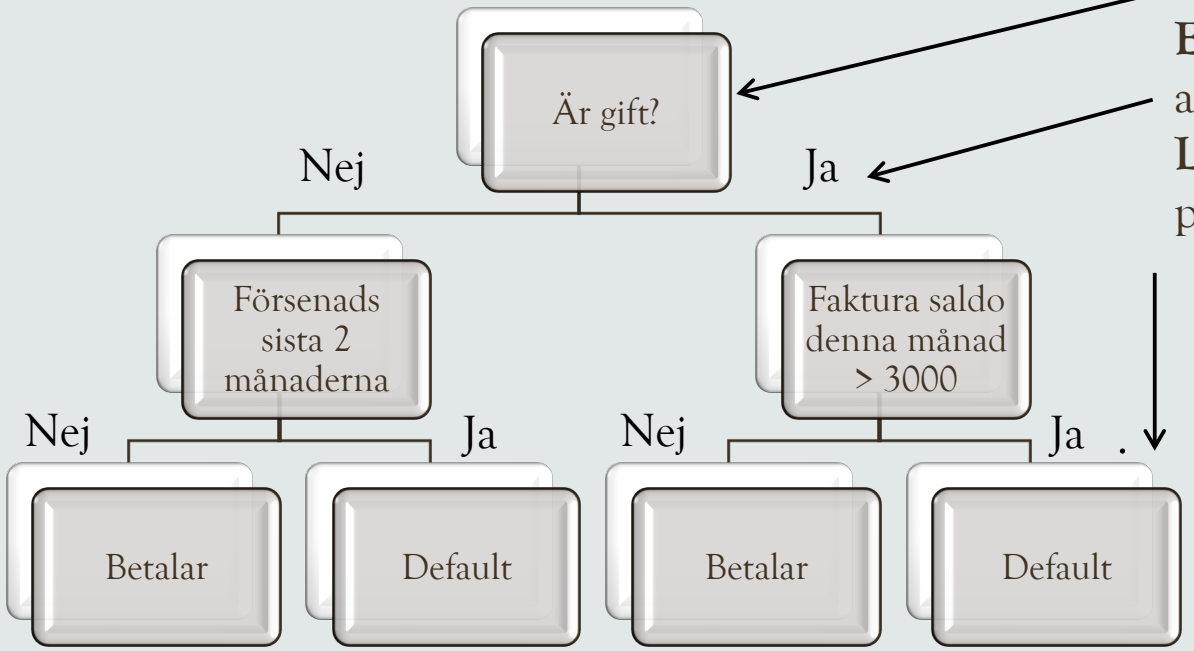
Classification Trees

- Målet är att skapa en modell som predikterar target y genom att lära sig simpla beslutsregler (if-then regler) baserat på slutsatser den drar från egenskaperna x
- Fördelar
  - Enkla att förstå och att visualisera
  - Kräver lite data förberedning
- Nackdelar
  - De tenderar att overfitta träningsdatan
  - De funkar inte bra när det är en obalans mellan positiva och negativa klasser t.ex. 90% negativa klasser och 10% positiva klasser



Classification Trees

Prediktera kredit default nästa månad:



Består av:

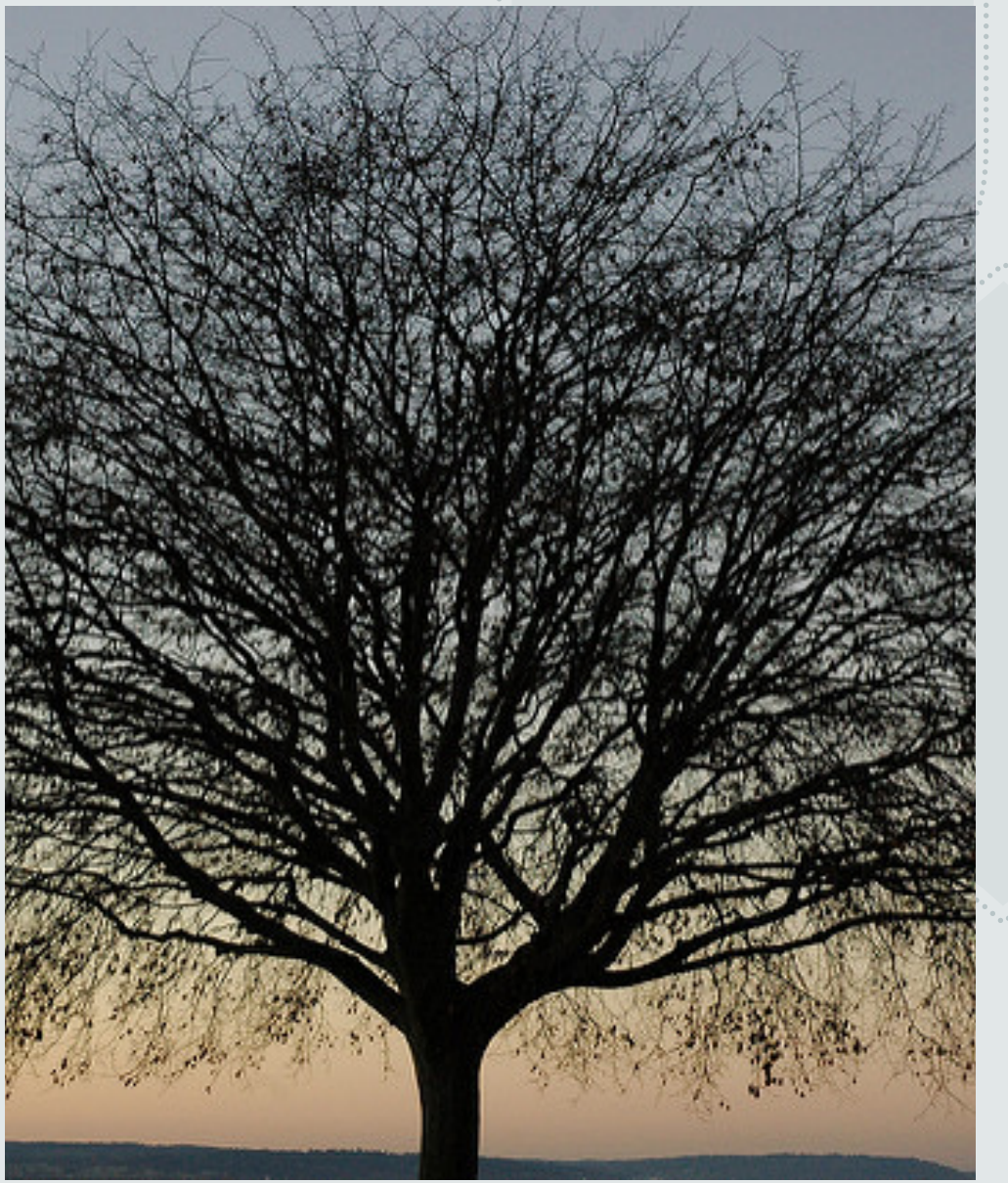
Nodes – testar värdet till en variabel

Edges – Resultatet av testen och anslutning till nästa node eller leaf

Leaf – ”terminal” slutliga noder som predikterar klassen

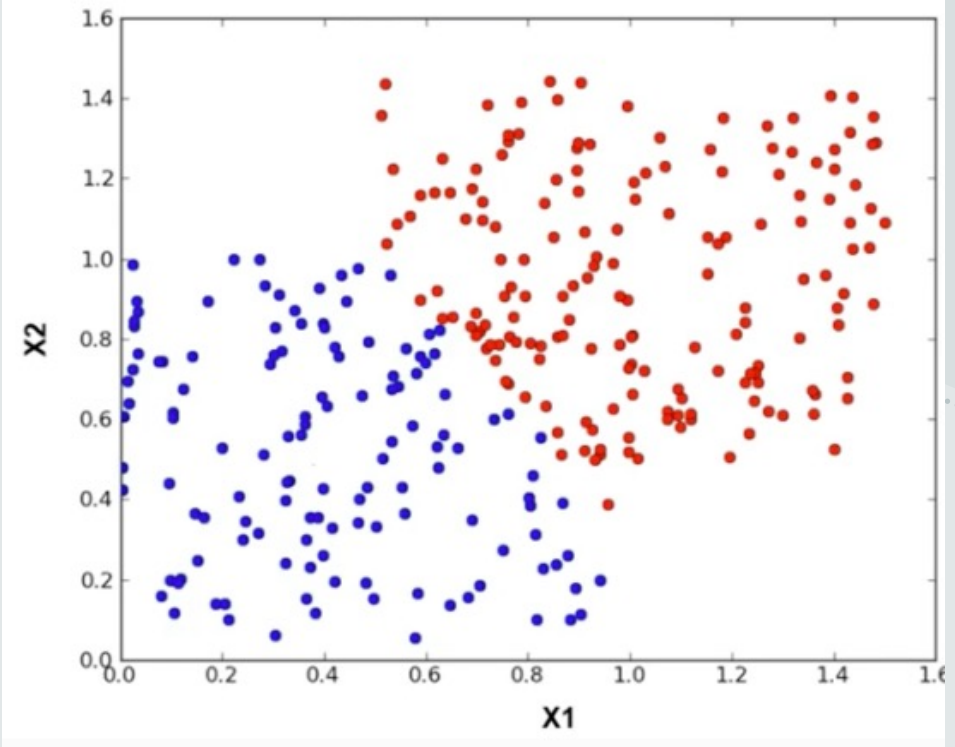
Classification tree

- Byggd upp av *rule inference*. Iterativ process som delar upp datan och sedan delar upp den vidare på var och en av grenarna.
- Använder en beslutsalgoritm. Vid trädroten splittras datan efter den featuren som ger störst vinst i information (reducerar osäkerhet mest när man klassificerar)
- Genom en iterativ process fortsätter datan splittras
- Sätter en gräns för hur djupt trädet kan bli för att undvika overfitting



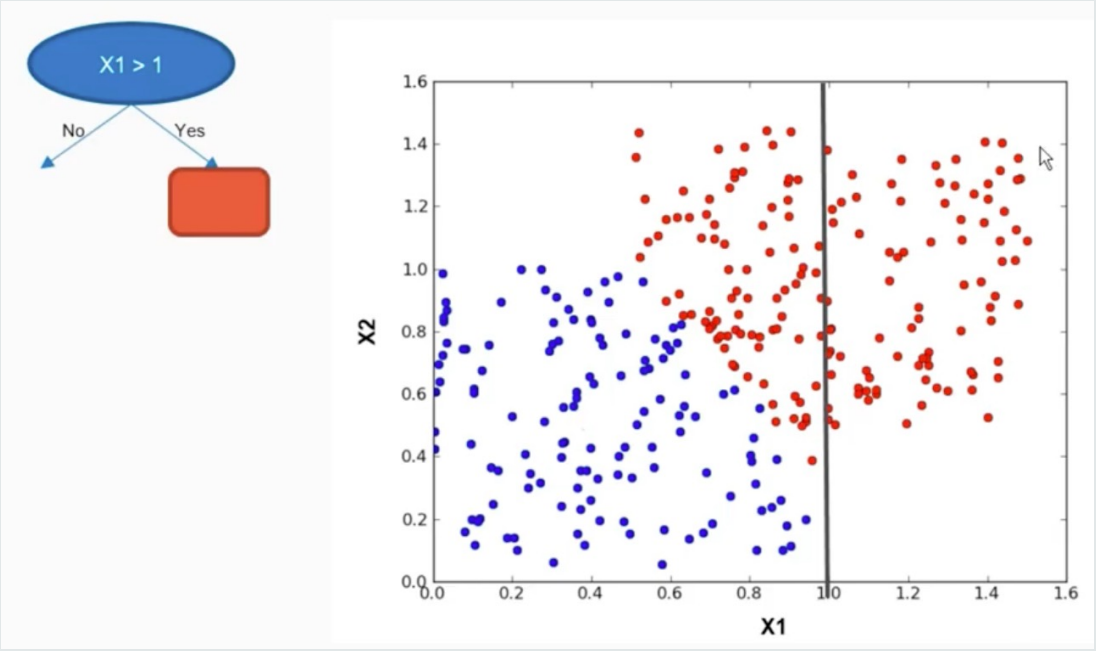
Rule inference

- I detta exemplet är det två egenskaper (features) x_1 & x_2
- Två kategorier: Blå och röd
- Generella regeln är att algoritmen försöker dela in egenskaperna i lådor/delar sådana att delarna är så ”rena” som möjligt
- Med ”ren” menar vi att varje del innehåller nästan enbart observationer från samma kategori

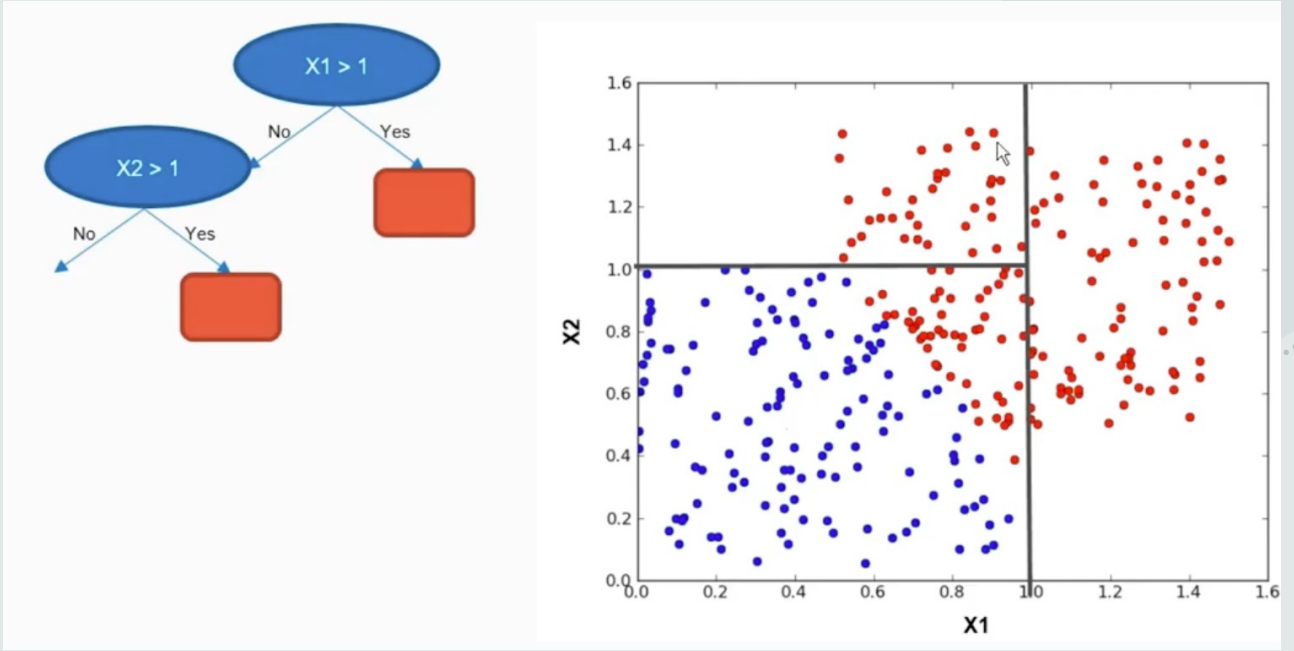


Rule inference

Regel 1

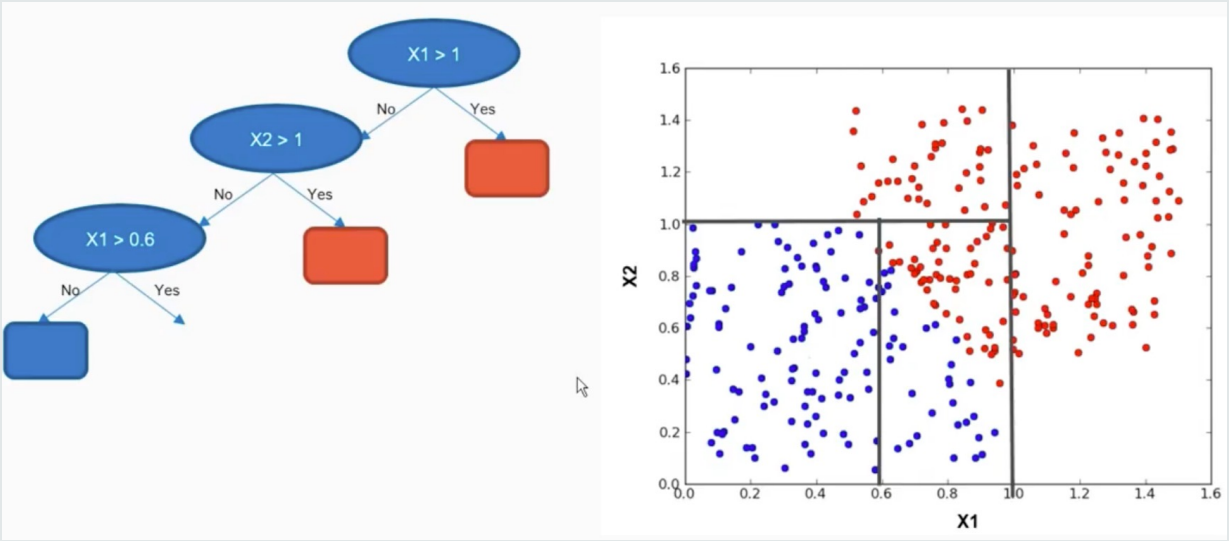


Regel 2

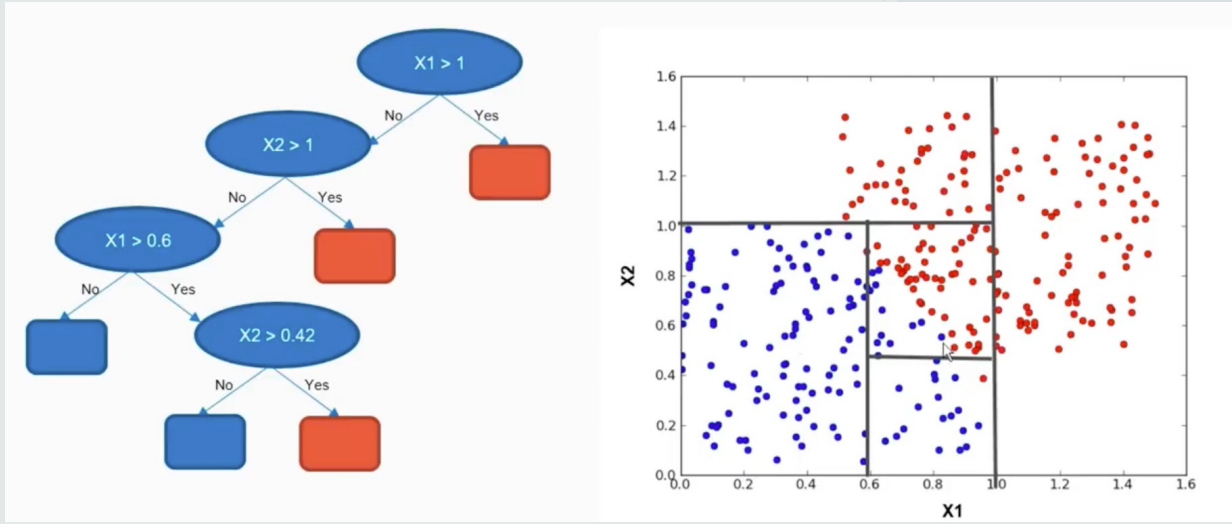


Rule inference

Regel 3



Regel 4

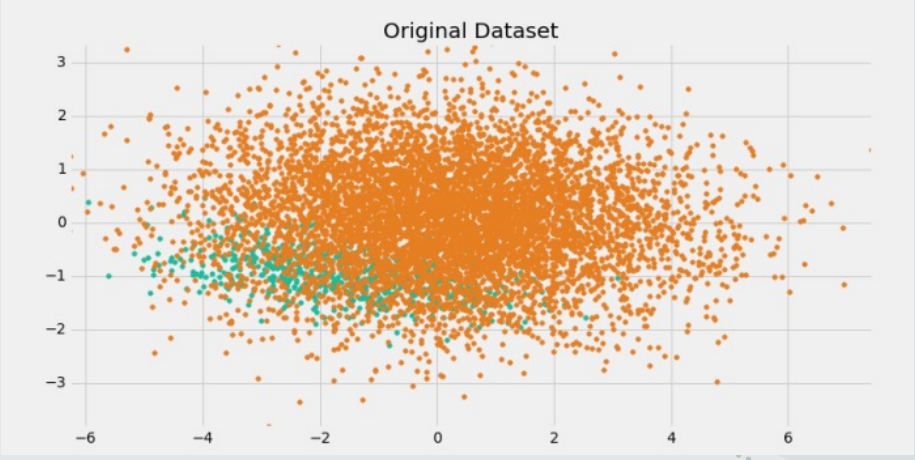


DecisionTreeClassifier Estimator in scikit-learn

- Importera DecisionTreeClassifier från träd metoden
- Importera parameter:
  - max_features: The number of features to concider when loking for the best split
  - max_depth: The maximum depth of the tree, normal 4-12) (risk for overfit if to deep
  - min_sample_split: The minimum number of samples required to split an internal node (can be used to counteract overfitting)
  - min_sample_leaf: The minimum number of samples required to be at a leaf node (can be used to counteract overfitting)

Imbalanced data

- Klassificerings innebär att man förutsäger en ”class label” etikett för en given observation.
- Ett obalanserat klassificeringsproblem är där det finns mycket färre datapunkter i en av klasserna. Det kan variera från en liten till en allvarlig obalans. *Ex det finns ett datapunkt i minoritetsklassen för hundratals, tusentals eller miljoner exempel i majoritetsklassen*
- Imbalanced data utgör en utmaning för prediktiv modellering eftersom de flesta maskininlärningsalgoritmerna som används för klassificering utformades med antagandet om lika många datapunkter i varje klass.
- Detta resulterar i modeller som har dålig prediktiv prestanda, särskilt för minoritetsklassen. Detta är ett problem eftersom minoritetsklassen vanligtvis är viktigare och därför är problemet känsligare för klassificeringsfel för minoritetsklassen än majoritetsklassen.



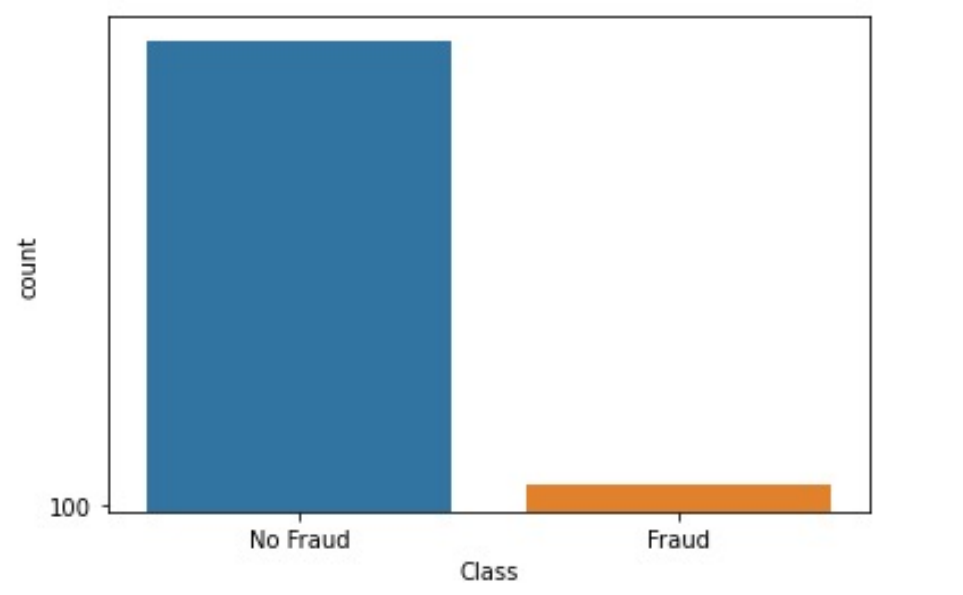
Stratified Train-Test Splits – Imbalanced data

- Gäller endast klassificering.
- Vissa dataset med **kategorisk target** har ett obalanserad (imbalanced) antal värden i target klassen.
- *Ex det finns 1000 datapunkter med registrerad cancer (1) och 10.000 där det inte är cancer (0).*
- Önskvärt att dela upp datamängden i träning- och testset på ett sätt som **bevarar samma proportioner av fall** av varje klass som observerats i den ursprungliga datamängden.
- Detta kallas stratified train-test splits.
- Vi kan uppnå detta genom att ställa in "stratify" -argumentet till y-komponenten i originalet
  Python: i train_test_split() modulen sätt stratify=y

Undersampling and oversampling - Imbalanced data

Random resampling:

- **Oversampling** – Duplicating samples from the minority class
- **Undersampling** – Deleting samples from the majority class.



”The class imbalance problem”

1. Anta att en accuracy metric är rätt val för att mäta fel när det inte är det
2. Anta att test distributionen är det samma som träningsdistributionen när det inte är så
3. Anta vi har tillräckligt från minoritetsklassen när det inte är så

Undviker vi dessa tre misstagen har vi inte problem med obalanserade klasser

Naive bayes – Probabilistic classifiers based on applying Bayes' theorem

Var modellen kommer ifrån

Några karakteristik av Naive Bayes classifiers

Två typer av Naive Bayes klassificeringar man kan använda i scikit-learn

Intuition bakom The Naive Bayes Models

- Naive Bayes är en familj med sannolikhets klassificeringar baserad på att använda Bayes' teorem med den "naiva" antaganden av att alla features är oberoende
- $P(A|B)$ – posterior. Sannolikheten för en händelse A händer given att händelse B har hänt
- $P(B|A)$ – likelihood. Sannolikheten B händer om A har hänt
- $P(A)$ – prior. Sannolikheten till hypotesen
- $P(B)$ – evidence. Sannolikheten för att observera evidence. Ofta ignorerat

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Karakteristik om Naive Bayes Models

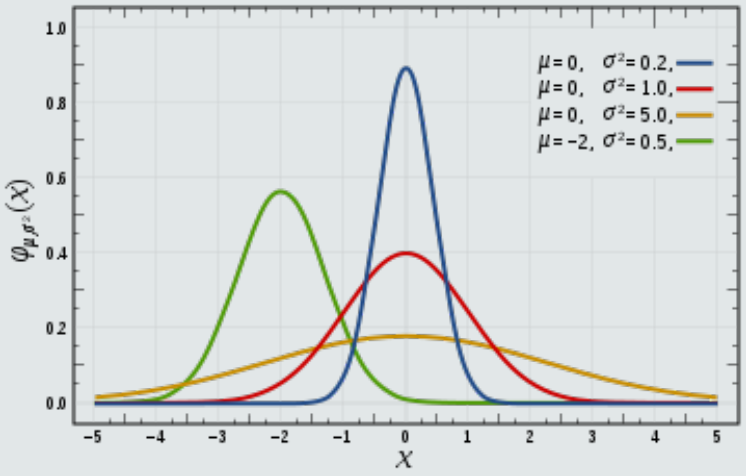
De är högt skabara och snabba att träna och tillåter en bra mängd features

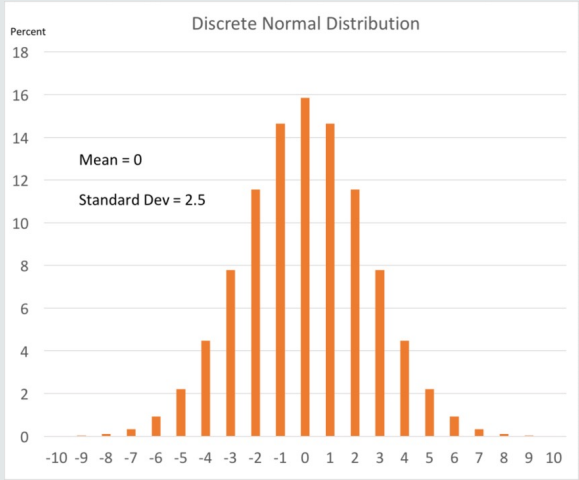
Naive Bayes Models producerar väldigt goda prediktioner även när man jämför med mer komplexa modeller

De har få hyperparameter i modellen vilket är en fördel

Gaussian ND
Multinomial Naive Bayes

- **Gaussian NB** är en modell från Bayes familjen där likelihooden till input features antas vara normalfördelade (Gaussian). Användbar när features är *kontinuerliga*
- **Multinomial NB** passar bra när klassificeringen är med *diskrete features*





The Gaussian Naïve Bayes in scikit-learn

- Använder GaussianNB från scikit.naive_bayes
- Relativt lätt att använda för simple modellering, inte manga hyperparametre att använda
-
- Methods:
- predict_proba: for predicting probabilities
- predict: for predicting classes

The Gaussian och Multinominal Naive Bayes in scikit-learn

Använder GaussianNB och MultinomialNB från scikit.naive_bayes

Relativt lätt att använda för simple modellering, inte manga hyperparametre att använda

Multinomail: alpha är en smoothing parameter (use default value = 1)

Methods:

predict_proba: for predicting probabilities

predict: for predicting classes

Threshold Value och metrics

- Modeller baserad på sannolikheter predikterar att det är den positive eller negativa klassen (1 eller 0) vid att använda beräknade sannolikhet och en **threshold value**
- Det är minsta sannolikhet som klassificerar en observation I den *positiva klassen*
- Default är 0.5 (50%)
- Den kan ändras och detta kan användas för att justera precision och recall metrics i klassificeringen

Predicting Credit Card Default

- Default (positiv klass) är att man inte betalar krediträkningen
- Varje rad är en kund

Predicting Bankruptcy

- The dataset is about bankruptcy prediction of Polish companies
- Basing on the collected data five classification cases were distinguished, that depends on the forecasting period:
  - 1stYear
  - 2ndYear
  - 3rdYear
  - 4thYear
  - 5thYear

Övning

Vad har vi gjort idag?

- Classification trees
- Rule inference
- Imbalanced data
- Naive Bayes

Nästa lektion

- Data preprocessing
- Hantera NaN värden
- Feature scaling: normalisering, standardisering, robust scaler
- Outliers – upptäcka och hantera