Scikit-learn

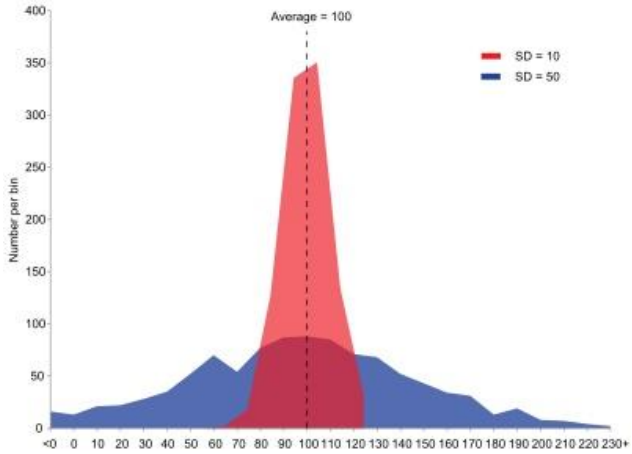
and some more stuff

Agenda

- Exploratory Data Analysis
- Correlations
- Labels
- Train-test split
  - why
  - how
- Validation

What is variance?

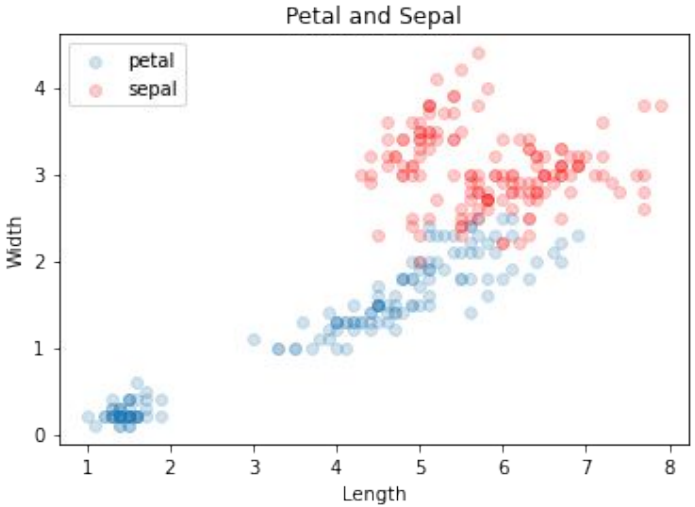
- The spread between numbers in a data set.
- Variance measures how far each number in the set is from the mean and thus from every other number in the set.



What is bias?

Exploratory Data Analysis - examples

If dataset is very large - take a subset (see splits) for faster manipulations.



Exercise

Iris dataset - load into dataframe

For all plots: Annotate with appropriate title and labels.

A) Scatterplot that shows the petal length and width and sepal length and width all in one plot. Let petals be blue dots and sepals be red.

B) Scatterplot of sepal length and width, color by target.

C) Two scatterplots, Petal and Sepal, colored by target.

Correlation and Covariance

Measure the **relationship** and **dependency** between two variables.

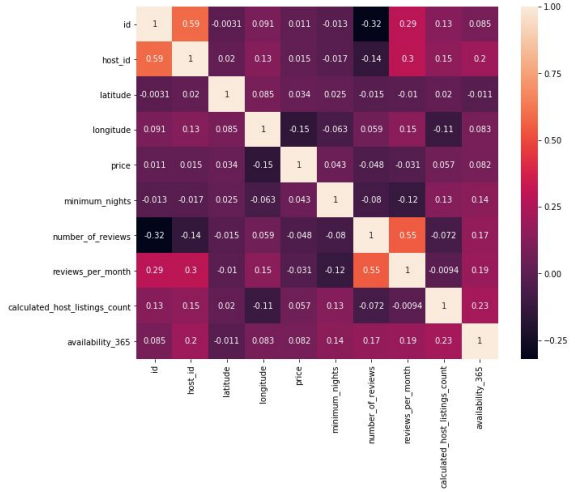
Covariance - direction, non-standardised

Correlation - direction and strength, standardised: values are between -1 (negative association) and +1 (positive association).

The closer to zero the weaker the correlation.

Negative association - one variable increases as the other decreases and vice versa

Positive association - move together, whether increase or decrease.



Correlation helps us investigate and establish relationships between variables.

```
from sklearn.preprocessing import
StandardScaler
iris_std = StandardScaler().fit_transform(iris)
```

RESTORY...

Scikit-learn - Labels

Labels are qualitative data.

Usually word/s (as strings).

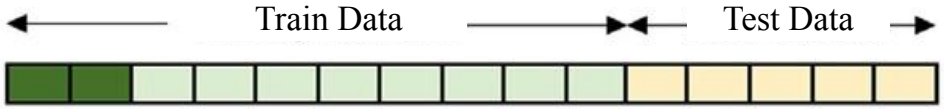
The **LabelEncoder** in Scikit-learn will convert each unique string value into a number, making out data more flexible for various algorithms.

Values between 0 and n_classes-1.

```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
data = ["paris", "paris", "tokyo", "amsterdam"]
le.fit(data)
LabelEncoder()
list(le.classes_)

>>> ['amsterdam', 'paris', 'tokyo']
```

Scikit-learn -Train-test split



- Why - To truly measure how correct/ effective our model is.
- Data - split train/test, test size usually 20%
  Can be random but better to retain the variance of the population (which hopefully what the data reflects) - stratified (if important attribute is known).
- Do this early on so that no bias can “enter”, which one might get once seen the data.
- Train the model - “have the answer”
- Test the model - predictions! Must remove the “answer” so that it’s not used.
- Not very appropriate when the dataset is small.

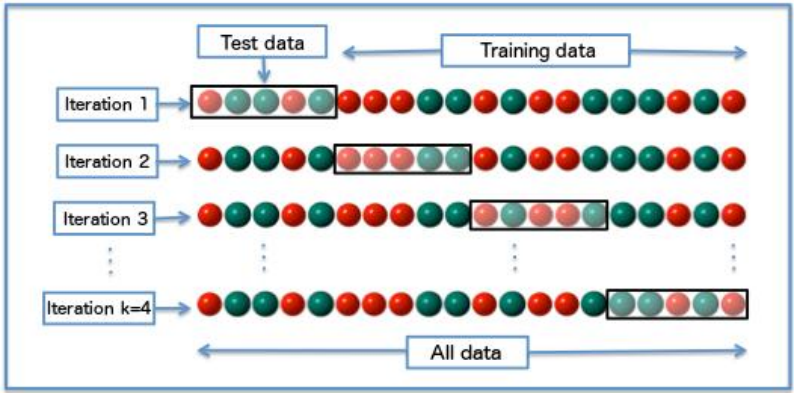
```
from sklearn.model_selection import train_test_split
train, test = train_test_split()
```

Scikit-learn - Validation

train - validate - train - validate … DONE

Ensures a

Split: **k** is the magic number
usually 5 or 10, but the important thing is that it is split **evenly**.



"If the test set is locked away, but you still want to measure performance on unseen data as a way of selecting a good hypothesis, then divide the available data (without the test set) into a training set and a validation set." - AI: A Modern Approach

… there is a bias-variance trade-off associated with the choice of k in k-fold cross-validation. Typically, … one performs k-fold cross-validation using k = 5 or k = 10, as these values have been shown empirically to ***yield test error rate*** *estimates that suffer neither from excessively high* ***bias*** *nor from very high* ***variance****.*

Data - splits

Train/Test Split: Taken to one extreme, k may be set to 2 (not 1) such that a single train/test split is created to evaluate the model.

Stratified: The splitting of data into folds may be governed by criteria such as ensuring that each fold has the same proportion of observations with a given categorical value, such as the class outcome value. This is called stratified cross-validation.

LOOCV: Taken to another extreme, k may be set to the total number of observations in the dataset such that each observation is given a chance to be the held out of the dataset. This is called leave-one-out cross-validation, or LOOCV for short.

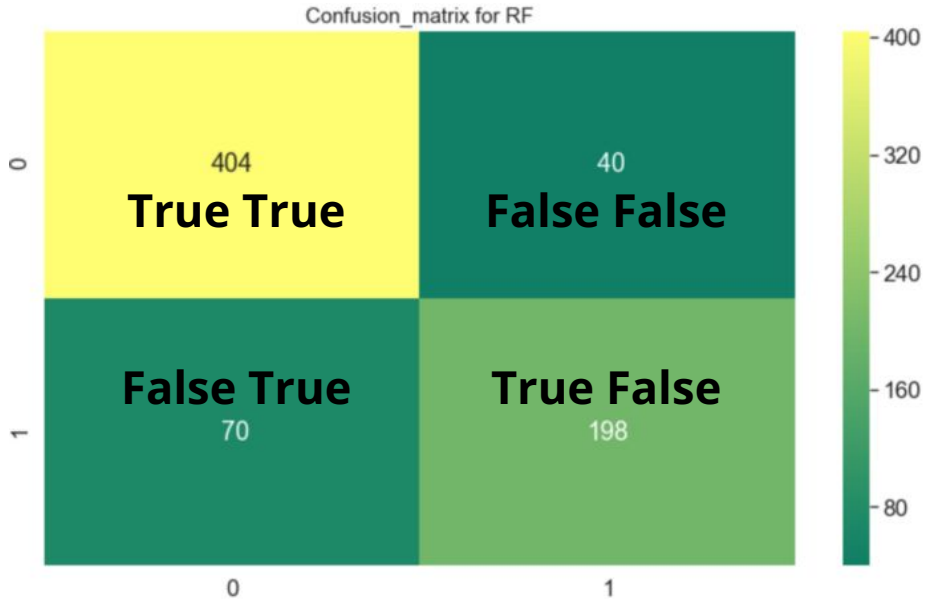
Repeated: This is where the k-fold cross-validation procedure is repeated n times, where importantly, the data sample is shuffled prior to each repetition, which results in a different split of the sample.

Nested: This is where k-fold cross-validation is performed within each fold of cross-validation, often to perform *hyperparameter* tuning during model evaluation. This is called nested cross-validation or double cross-validation.

True or False

Predicted value vs actual value

	Prediction	Actual
True True	True	True
False True	True	False
False False	False	True
True False	False	False



Exercise - Titanic

Missing data - how to fix and fix

correlation before and after fix

- % survived and not survived per Pclass
- % survived sex m vs f out of all and within class
- % survived for age (binning)
- % survived per fare (highest, lowest - binning)

sibsp # of siblings/spouses aboard

parch # of parents/children aboard

ports **C**=Cherbourg, **Q**=Quieenstown, **S**=Southampton

Sammanfattning

- What is variance? What is Bias?
- Exploratory Data Analysis
- Correlations
- Labels
- Train-test split
  - why
  - how
- Validation

Länkar

- Plot examples
- Iris plots
- Covariance vs. Correlation
- K-fold validation
- Difference between testing and validation
- Scikit-learn LabelEncoder

- **Överkurs**:
  - Scikit-learn train test split for evaluation of model
  - Scikit-learn ML cheat-sheet