Introduction to Machine Learning

M. Fuat Kına

Outline

- What is ML?
- Types of ML models (supervised, unsupervised)
- Supervised learning
- Classification (k-NN)
- Regression (regularization)
- Hyperparameter tuning
- Preprocessing
- Pipeline
- Unsupervised learning (k-means clustering)

Credits for:

https://jakevdp.github.io/PythonDataScienceHandbook/

https://app.datacamp.com/learn/courses/machine-learning-with-scikit-learn

https://app.datacamp.com/learn/courses/supervised-learning-with-scikit-learn

https://app.datacamp.com/learn/courses/unsupervised-learning-in-python

What is Machine Learning?

- “Machine learning technology teaches computers how to perform tasks by learning from data instead of being explicitly programmed.”
- Giving computers the ability to learn to make decisions
  - whether an email is spam or not spam given its content and sender
  - cluster Wikipedia entries into different categories based on the words they contain
- From Statistics to Machine Learning
  - from fitting the model to predicting the world through nowcasting or forecasting…
- When there are labels present, we call it supervised learning. When there are no labels present, we call it unsupervised learning.

Unsupervised and Supervised Learning

Unsupervised, unlabeled

- Uncovering hidden patterns and structures from unlabeled data
- Clustering

Supervised, labeled

- The goal is to learn from data for which the right output is known, so that we can make predictions on new data for which we don't know the output
- automate a time-consuming or expensive manual task
- to make predictions about the future
- need labeled data (historical data, crowdsourced labeling, etc.)

Supervised learning

- Using predictor variables or features and a target variable, to build a model that is able to predict the target variable
  - Continuous target variable
    - *Regression*
  - Binary target variable
    - *Classification*
- *Feature* or independent variable - predictor
- *Target* or dependent variable - response - outcome
- Python: Scikit-learn as **sklearn**

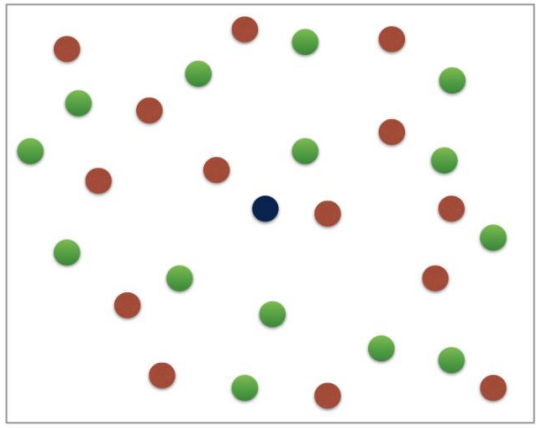
Which of these is a classification problem? Provided below are 4 example applications of machine learning. Which of them is a supervised classification problem?

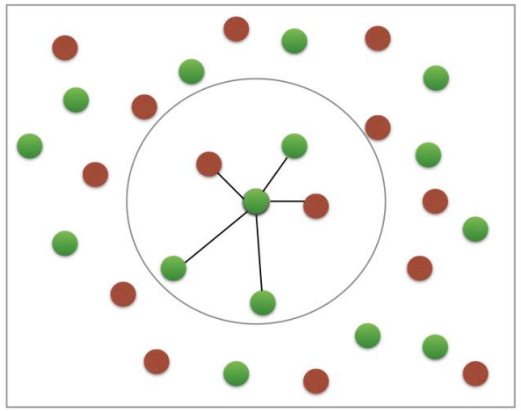
A. Using labeled financial data to predict whether the value of a stock will go up or go down next week.

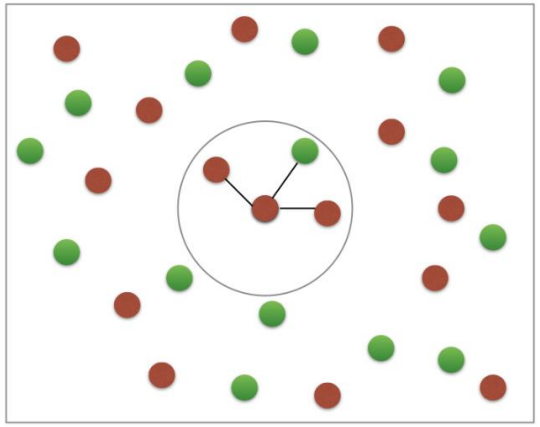
B. Using labeled housing price data to predict the price of a new house based on various features.

C. Using unlabeled data to cluster the students of an online education company into different categories based on their learning styles.

D. Using labeled financial data to predict what the value of a stock will be next week.







Classification

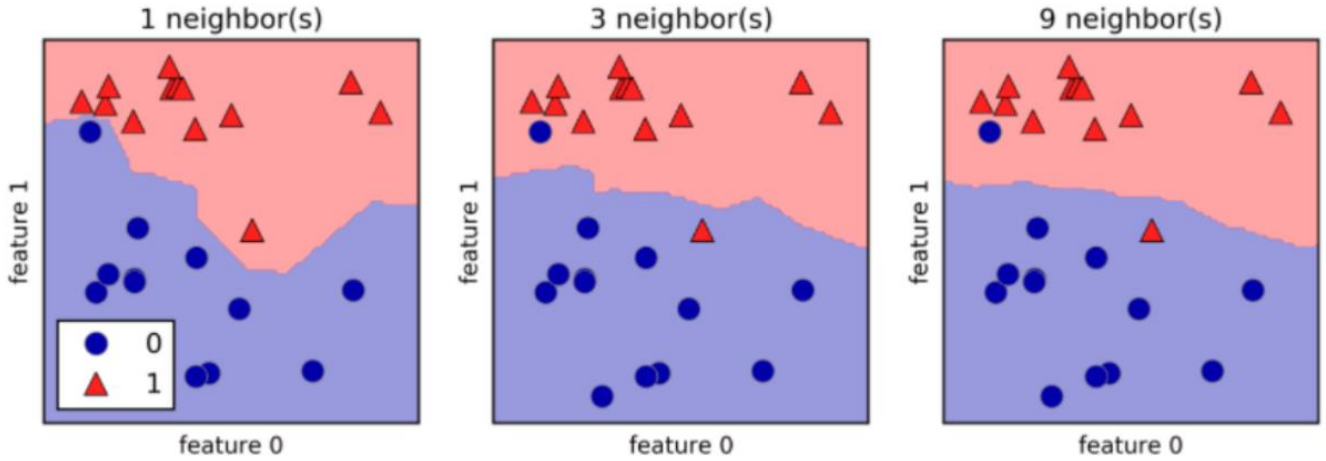
- k-Nearest Neighbors
  - Looking at k closest labeled data points, and taking a majority vote
  - Draws decision boundaries

Classification: sklearn fit and predict

- Two steps of Machine Learning: learning and predicting
- Training a model on the data = .**fit**()
- To predict labels of a new data = .**predict**()

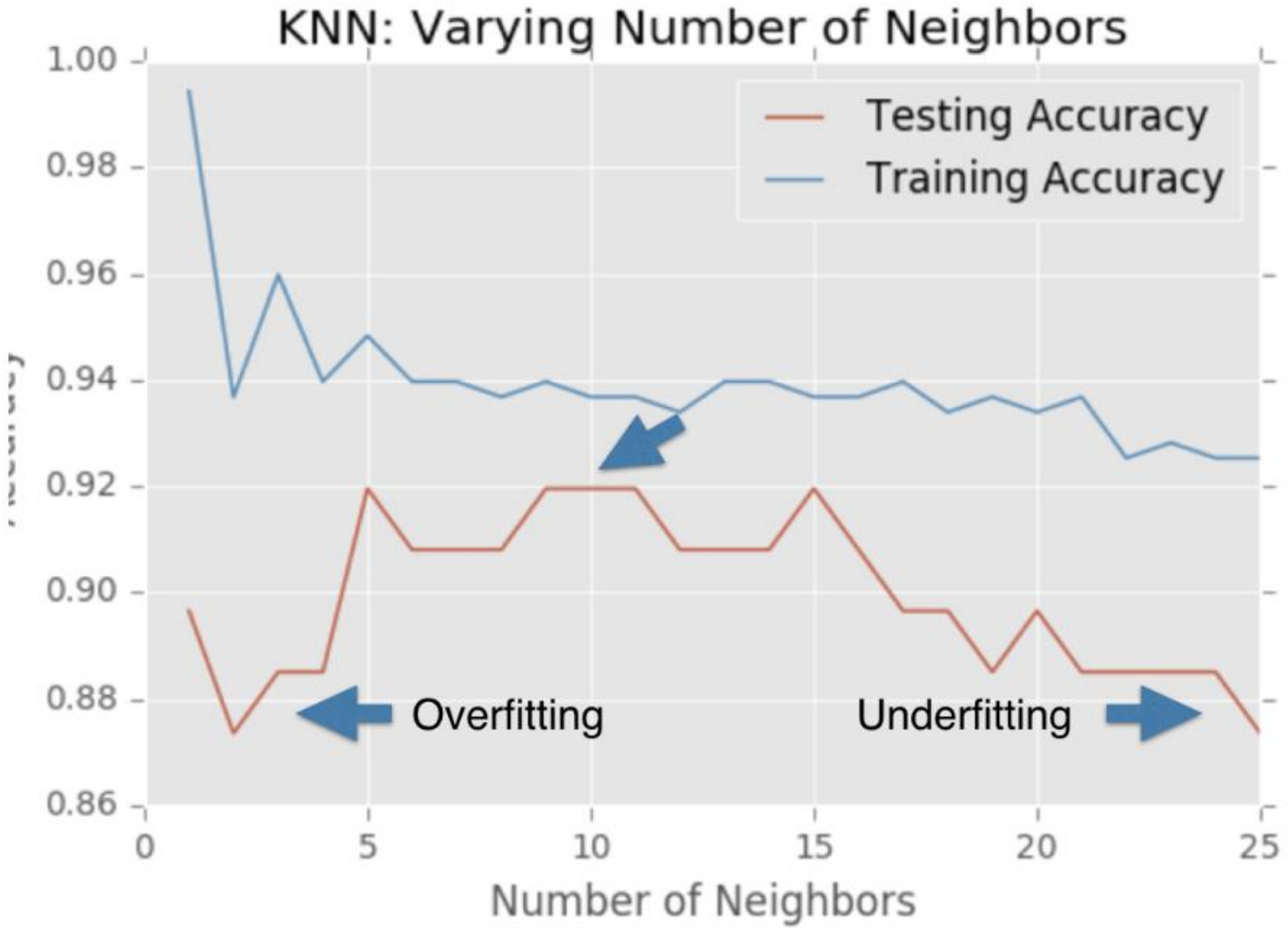
How to measure performance?

- Accuracy = fraction of correct predictions
- For which data should I check the accuracy, fitted data or predicted data?
- **Train/test split**
  - Fit the classifier on the training set
  - Make prediction on the test set



Model complexity

- Larger k = smoother decision boundaries = less complex model
- Smaller k = more complex model = can lead to overfitting


KNN: Varying Number of Neighbors
Testing Accuracy
Training Accuracy
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
Overfitting
Underfitting
0
5
10
15
20
25
Number of Neighbors

Regression

- Continuous target variable

Which of the following is a regression problem?

1. An e-commerce company using labeled customer data to predict whether or not a customer will purchase a particular item.
2. A healthcare company using data about cancer tumors (such as their geometric measurements) to predict whether a new tumor is benign or malignant.
3. A restaurant using review data to ascribe positive or negative sentiment to a given review.
4. A bike share company using time and weather data to predict the number of bikes being rented at any given hour.

Basics of a regression

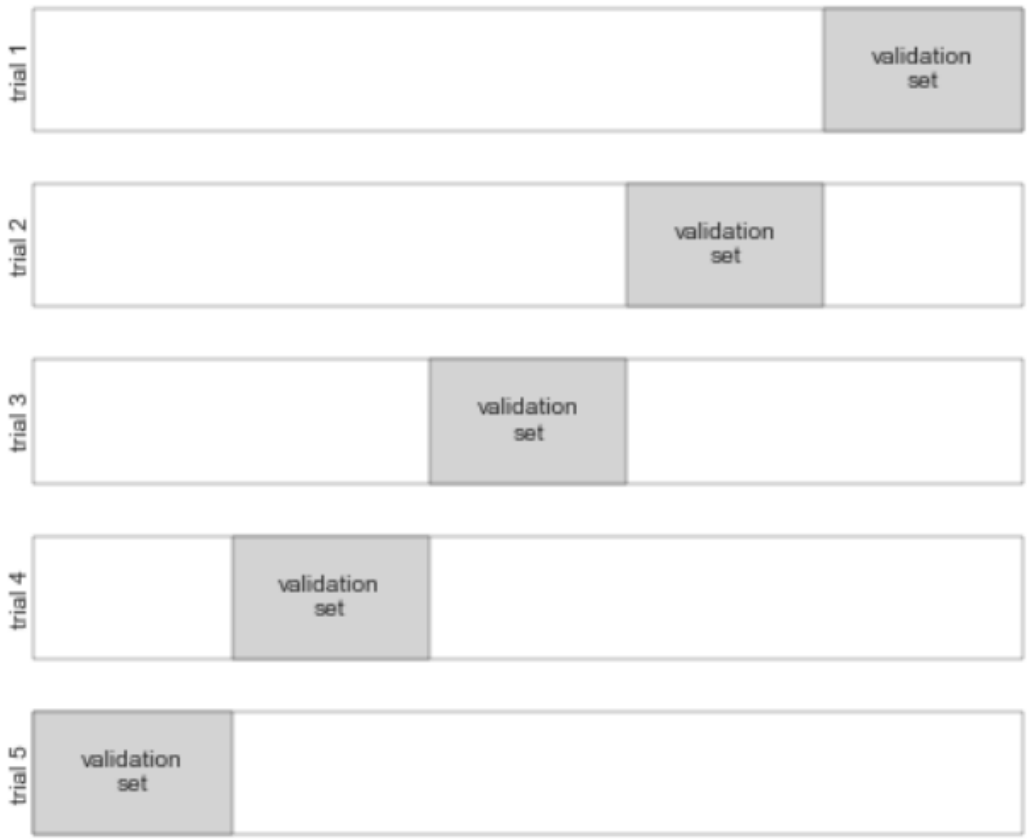
- y=ax+b
  - Define a line that minimizes the error/loss function
  - OLS -> tries to minimize sum of squares of residuals

- Let’s run the traditional OLS model with **sklearn**

What is really new within Machine Learning models?

- Cross-validation
- Regularizers
  - Lasso
  - Ridge
- Fine-tuning the model
  - ROC curve
  - AUC computation
- Preprocessing and pipelines

Cross-validation



Regularizers

- OLS has a perfect overfitting problem
- Regularization: Penalizing the coefficient

Regularizers

- Ridge:
  - Loss function = OLS loss function + ridge penalizer
  - Alpha controls model complexity
  - Alpha is zero for OLS (overfitting)
  - Very high alpha can lead to underfitting

$$\alpha * \sum_{i=1}^{n} {a_i}^2$$

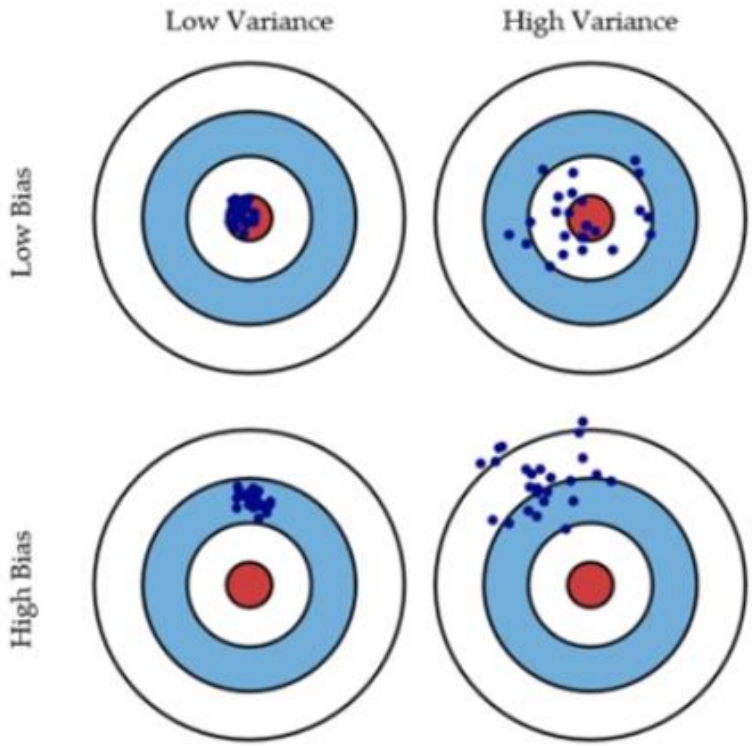
Regularizers

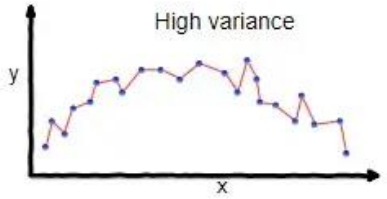
- Lasso:
  - Loss function = OLS loss function + lasso penalizer
  - Alpha controls model complexity
  - Alpha is zero for OLS (overfitting)
  - Very high alpha can lead to underfitting

  - Lasso helps to detect the most important feature, or relatively less important features

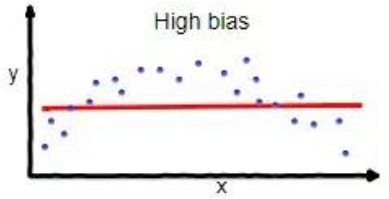
- Let’s see cross-validation and regularizers in python

$$\alpha * \sum_{i=1}^{n} |a_i|$$

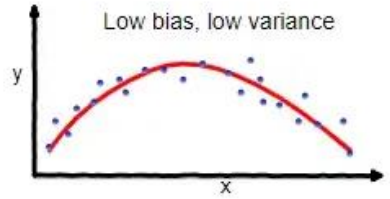




overfitting



underfitting



Good balance

Regularizers

Fine-tuning the model

- How good is your model?
  - Accuracy?
    - Spam classification: 1% is spam, %99 is real
    - If ALL e-mails are classified as real, the model becomes 99% accurate!
  - Called class imbalance
  - We need more nuanced metrics

Fine-tuning the model

- Confusion matrix:

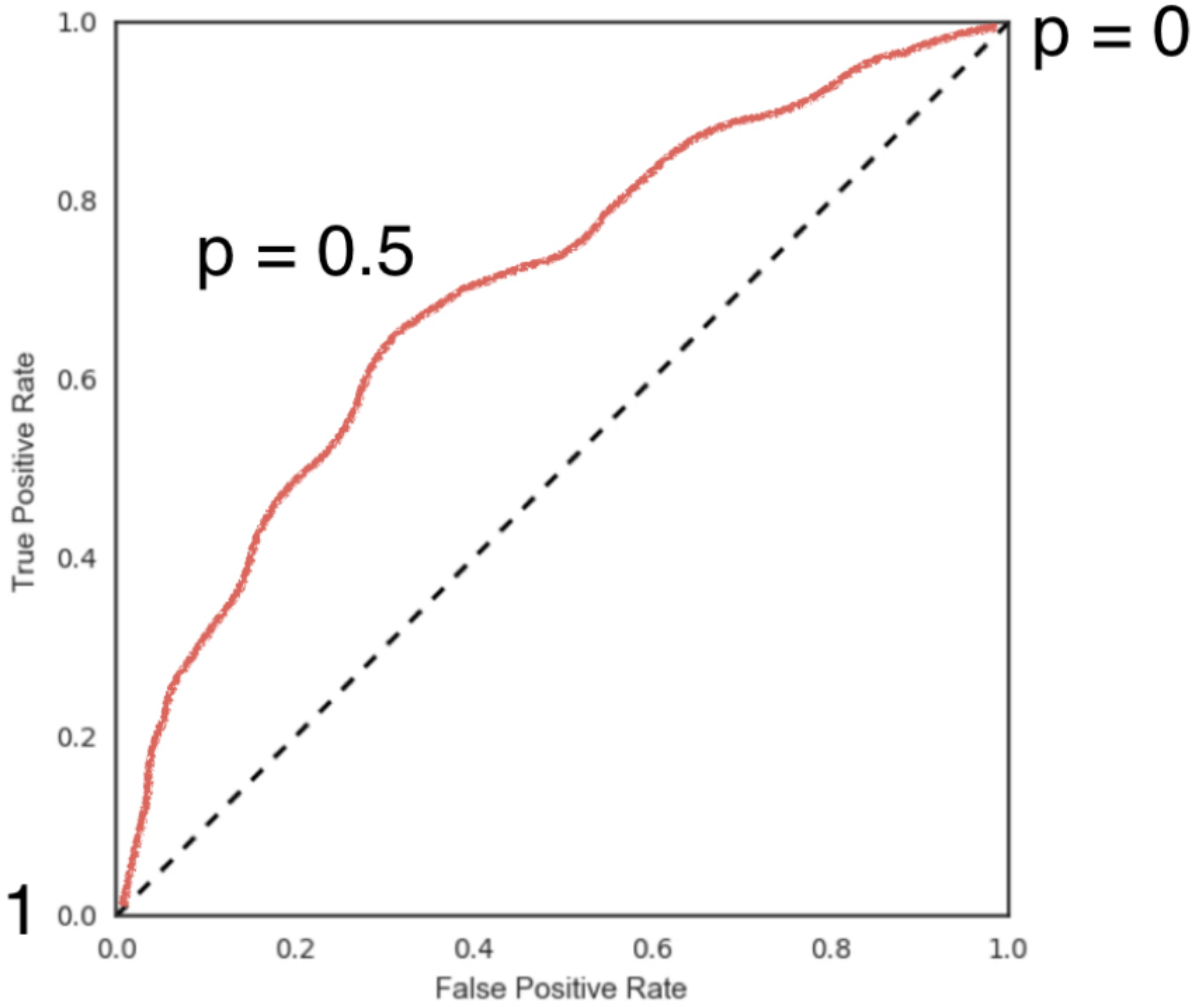
	Predicted: Spam Email	Predicted: Real Email
Actual: Spam Email	True Positive	False Negative
Actual: Real Email	False Positive	True Negative

- Accuracy = (tp + tn) / (tp + tn + fp + fn)
- Precision = tp / (tp + fp)
- Recall = tp / (tp + fn)

- F1score = 2 * (precision*recall) / (precision + recall)

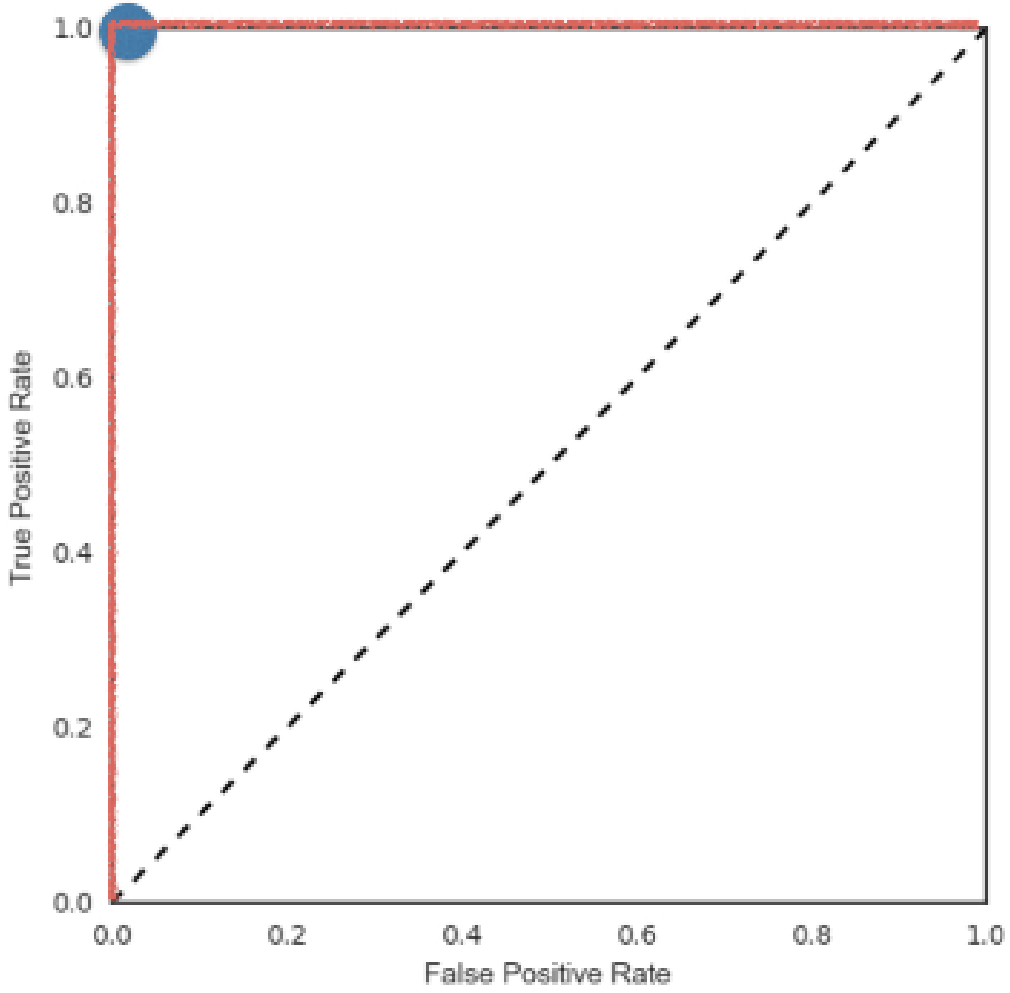
Logistic Regression and the ROC curve

- Logistic regression does similar with classification, predicting probabilities (e.g., label 1 if p>0.5)
- The ROC curve (varying treshold):



The ROC curve and AUC

- Larger the area under the ROC curve = better model (AUC)
- Let’s see confusion matrix, ROC curve and AUC computation in python



Hyperparameters tuning

- Ridge and Lasso regression / choosing alpha
- k-Nearest neighbors / choosing n_neighbors
- Parameters like **k** and **alpha** are hyperparameters
- How to choose the correct hyperparameter?
- Try a bunch of hyperparameter values, fit all of them separately, see how each performs, choose the best performing one
  - Grid search cross-validation
  - Randomized search cross-validation

- Let’s do practice!

Preprocessing and Pipeline

- Real world data is not prepared for you to run the best model
- Convert categorical features to dummies
    - Scikit-learn: OneHotEncoder()
    - Pandas: get_dummies(), pd.cut()
- Handle the missing data
    - If drops the rows/observations: data loss
    - Then, impute!
        - Using the mean of non-missing entries
        - Linear imputation if you have a time dimension
        - Scikit-learn: Imputer

```
from sklearn.preprocessing import Imputer
imp = Imputer(missing_values='NaN', strategy='mean', axis=0)
imp.fit(X)
X = imp.tranform(X)
```

- Pipeline: Do all at once!
    - Impute, Normalize, Run, Fit, Predict, Test…

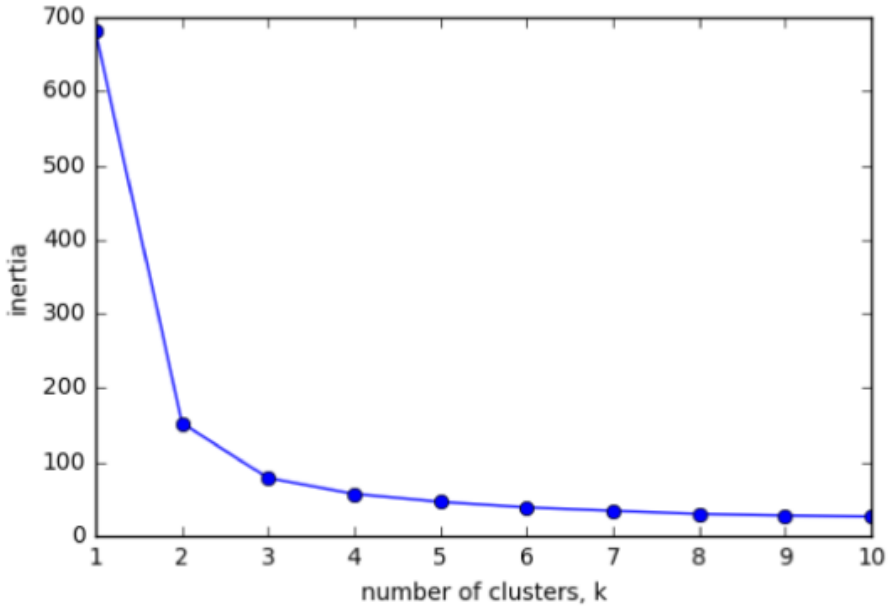
Unsupervised learning

- While supervised learning finds patterns for a prediction task, unsupervised learning finds patterns in data without a specific prediction task in mind.
- Remember Iris dataset!
- And let’s consider we only have features (without the target), run an unsupervised model, the model figures out natural clusters already available in the data.
- k-means clustering
  - Resembles kNN classification, but here there is no label.
  - Model predicts labels all by itself.

Clustering

- Measuring the quality of our clusters: Inertia!
- A good clustering has tight clusters (which means more clusters and low inertia), but not too many clusters
- Then, choose the elbow (as often suggested)

- Let’s run a pipeline on the k-means clustering



Some other machine learning models

- Naive Bayes Classification
- **Support Vector Machines**
- **Decision Trees and Random Forests**
- **Principal Component Analysis**
- Manifold Learning
- Gaussian Mixture Models
- Kernel Density Estimation
- …
- Neural networks
- …