**Machine Learning – Expected Travel Time**

K-Nearest-Neighbor:

* Trainingsphase: speichern der Vektoren und labeln der Trainingsdaten
* Klassifikation: k ist eine definierte Konstante, Vektor wird klassifiziert indem getestet wird, welche Klasse in dem vorgegebenen Abstand (k) öfter vorhanden ist
* Schwierigkeit: k richtig wählen
* Lösbar durch Gewichtung der Klassifizierung

Support Vector Regression

* Ausgangssituation: Mengen, bei denen bekannt ist, welcher Klasse sie angehören, jedes Objekt ist Vektor
* Ziel: Hyperebene, die Trainingsobjekte in zwei Klassen teilt
* Regeln: 1. Ebene, die Klassen am besten trennt

2. Ebene, die den größten Abstand zu beiden Klassen hat

* Algorithmus kann mit „Ausreißern“ umgehen

Decision Trees

* Fragen werden an den Datensatz gestellt (z.B. „Ist heute ein Tag unter der Woche?“)
* Aus der Antwort wird der nächste Schritt abgeleitet
* Am Ende ist klar welche Daten einbezogen werden müssen

Lineare Regression

* y = a + b \*x (y ist Output, x ist Input)
* Ziel: a und b durch Algorithmus finden
* Algorithmus findet also lineare Funktion, die gegebene Daten am besten beschreibt

Logistic Regression

* Wahrscheinlichkeiten, für bestimmte Punkte, werden auf Grundlage der gegebenen Datensätze berechnet
* Werte zwischen 0 und 1