Models by Type

- **Classification**
  - Logistic Regression (non-linear) – outcome is either 0 or 1 (S-curve)
    - $P = \frac{e^{a+bx}}{1+e^{a+bx}}$
    - Logistic function is an example of a sigmoid function
  - Support Vector Machine (linear)
    - Drawing a line to separate classes of data points that maximizes the distance of the line to the closest points (support vectors) of each class (maximum margin)
    - Orientation on the support vectors (instead of stereotype examples of each class) sometimes leads to SVM having superior performance
  - Kernel Support Vector Machine (non-linear; such as Gaussian, Sigmoid, polynomial)
  - K-nearest Neighbors (non-linear) – classification of data point based on common class of the majority of its k nearest neighbors
    - 1. Define K (=number of neighbors taken into consideration for classification)
    - 2. Assign data point to the class of the majority of its K nearest neighbors
  - Decision Tree / Random Forest, XGBoost and other Ensemble Techniques (non-linear)
    - Decision tree is a greedy algorithm
  - Naïve bayes (Bayesian) – classifying based on the probabilities of presumably independent variables (non-linear)
    - Performs well with small amounts of data, is not that accurate
    - Classifying a new data point based on its features, given the classifications in priorly observed data based on individual (independent) probabilities of the features classifying prior data points
    - "A Naive Bayes Classifier is one where you have several things that describe what you are looking at, like it's color, size, whether it has eyes or not… and you assume that they have nothing to do with each other."
  - Decision Tree / Random Forest / Classification and Regression Trees (CART)
    - If/then logic, dependency between variables
    - Random forest = ensemble learner, several decision trees which results are aggregated into one based on majority vote from the trees
    - XGBoost (for extreme gradient boosting) is a performant framework for decision trees
      - It is an ensemble technique, that means it combines several algorithms sequentially to reduce errors based on gradient descent
      - Runs faster by optimal allocation of computing power

                - Strong in using regularization against overfitting
            - Reduce Overfitting (only pattern, no noise)
            - Pruning – removing sections of decision tree that add little insights in order to prevent overfitting
        - Artificial Neural Network (ANN)
            - See PPT about ANN
- **Regression**: Relation between 2 variables, or a dependent variable and a group of independent variables
    - Least Squares / Simple / Multiple Linear Regression (linear)
    - Support Vector Regression (SVR, linear and non-linear)
        - Linear Regression for higher dimensions
        - Choice of a kernel (like Gaussian) and regularization against noise/overfitting
        - Whereas SVM tries to minimize the error between prediction and real value, SVR makes sure no error is above a threshold
    - Polynomial Regression (non-linear)
    - Decision Tree / Random Forest (non-linear)
        - Splitting data based on features that result in the largest information gain (=largest purity of nodes)
            - Information gain can be computed based on 3 measures
            - Entropy – based on log function. Measures probability that data point belongs to certain class (100% - purity, 50/50% - impurity). Function (konkaver Bogen) is highest at 0.5 and thus is minimized (towards 0 or 1)
            - Gini Index: minimizing probability of wrong classification (is 0.5 for perfect binary classifier)
            - Classification error
        - Pruning the tree means setting a limit for the maximum depth (acts against overfitting)
        - Random Forest represents majority votes of several decision trees
            - Mode of the classes (classification) or mean prediction (regression)
    - Autoregressive Integrated Moving Average (non-linear)
    - Artificial Neural Network (ANN)
    - Tests
        - Heteroscedasti city: testing whether a model is capable of predicting values of dependent variable across the spectrum. If a sequence of variables is homoscedastic, all variables in the sequence have more or less the same variance
        - F-test: does regression fit data well?
    - Performance: Mean Squared Error (MSE)
- **Clustering**
    - SVM (Support vector machines) for classification/clustering

        - drawing a line between 2 groups of data points in a plot. Line should maximize the 2D difference (euclidean distance) to the datapoints
    - k-nearest neighbours (KNN) supervised clustering, similar to SVM but non-linear, performs better with a lot of data points. Majority vote of k neighbors of data points
        - Approach: assigning a new data point to given clusters
        - Take the neirest neighbors (usually n=5) of the data point according to Euclidean distance, assign data point to category that most neighbors belong to
    - K-means unsupervised clustering (KMC) – cluster into k clusters
        - Based on centroids – the graphical center point/average of the values in a cluster
        - Approach: Choose number of cluster, select random centroids, assign points to centroids, compute new centroid, reassign data points to new centroid (iteratively)
            - Number of clusters: based on minimizing WCSS (within cluster sum of squared distances) (elbow method)
            - K-Means++ avoids random centroid initialization trap (leading to wrong clusters)
    - Hierarchical (bottom-up clustering)
    - Gaussian Mixture Models (severall Gaussian distributions, with EM algorithm)
    - Self-Organizing Maps (SOMs, unsupervised deep learning)
        - Used for feature detection / dimensionality reduction
        - Nodes have fixed weights. Nodes closest to the ones that exactly predict outcome are grouped, similar process as in k-means clustering
- Recommender Systems & Association Rule Learning
    - Content-based (what a user has previously liked) vs. collaborative filtering (what other users have liked [item-based] or similarity to other users [user-based])
    - Apriori (recommender system)
    - Warp (Weighted Approximate-Rank Pairwise)
    - Singular Value Decomposition (SVD, similar to PCA)
    - Boltzmann Machines (unsupervised deep learning, based on PyTorch)
        - Recommendation system, non-directional neural network with visible and hidden nodes
    - Restricted Boltzmann Machines (RBM)
        - Based on visible nodes that show attributes weighted by hidden nodes
        - System tries to find global minimum by adjusting weights (contrastive divergence)
    - Deep Belief Network
        - Stacked RBMs with directionality from hidden to visible nodes
    - Deep Boltzmann Machines
        - Non-directional stacked RBMs

- Autoencoders (outputs the input after deconstructing (decoding) and encoding it again)
    - Use cases: recommendation system, dimensionality reduction, Google image search, fake faces
    - Stacked Autoencoder (two encoding layers, performs similar to Deep Belief Networks)
    - Deep Autoencoder (autoencoder based on stacked RBMs but directional)

- Sequential Models
    - (Uni- or bidirectional) Recurrent Neural Networks
        - For time series prediction, has a "memory"
        - Hidden layer gives an output, but also feeds back into itself based on recurrent weight
        - Bidirectional Recurrent Neural Network accesses both previous and subsequent sequences (for instance, previous and following word in a text sequence)
    - LSTM (Long Short-Term Memory Networks)
        - Solves problem of vanishing gradients. For text translation/ language modeling, image generation
        - Based on previously stored value and new value, "valves" decide whether or not to let a value pass through
    - Recursive Neural Networks
        - Every hidden layer is a tree structure
        - Creates tree representation of sequential input
    - GRU (Gated Recurrent Unit)
        - Similar to LSTM, more computationally efficient, by having only 2 gates, performs well on smaller datasets
    - Dynamic Memory Networks
        - State of the art for Q&A systems
        - They use an attention mechanism to filter for the relevance of data stored in the hidden state of the recurrent neural network
        - They use semantic (knowledge base based on GloVe) and episodic memory (based on 2 GRUs to create attention mechanism)
    - Sequence2Sequence (Seq2Seq)
        - Uses 2 LSTMs for encoding and decoding
        - Hidden state from LSTM encoder is fed into the decoder
        - Can be used to generate sequential data, like music
        - A selective attention mechanism ranks the relevance of LSTM output to be "memorized"
    - Differentiable Neural Computer (DNC)
        - ANN (or any network) with external memory storage/bank as matrix
            - Memory stores weights, which are differentiable (thus D NC)

                - Temporal links based on the usage of memory weights point towards different weights in the memory in order to "order" them
                - Network is feed-forward but system as a whole is recurrent, because read vector of previous memory state is fed back into the network
            - Controller reads and writes to this memory bank, similar to actual computer
                - Controller has attention mechanism to decide what to read and write to/from the memory bank
    - ARIMA (Autoregressive Integrated Moving Average) for showing seasonality/trends in data
        - Data shows a certain trend, seasonality (non-stationarity) for multivariate (more than 1 variable) time series
        - For unsupervised problems. Similar supervised problems - LSTM
- Representative Models
    - Convolutional Neural Networks (for image classification)
        - Process: Feature Detector creates convolution which is input to ReLu layer (for non-linearity). Max pooling then reduces data while preserving key infos. Flattening creates a vector out of the matrix of features. Vectors of images are fed into ANN (Output function: Softmax, Loss Function: Cross-entropy)
        - Add zero padding – preserving the image size after filtering mechanisms of the CNN
    - OpenCV
        - Image and object detection
    - Autoencoder (see recommender systems)
    - Denoising Autoencoder
        - Copies a deserved output by removing (purposefully added) noise from the input and learning a more dense representation
    - Contractive Autoencoder
        - Works with added regularization to adittionally penalize overfitting
    - Sparse Autoencoder
        - Increases complexity of representation by adding sparse vectors
    - Rule-based machine learning (unsupervised)
        - If/then logic based on combination of values of features classifying outcome
- Generative Models
    - Generative Adversarial Networks (GANs)
        - Composed of a generator and an (adversarial) discriminator which performs binary classification of the generated output by comparing it to the original data. The goal is for the generator to generate output which cannot be classified as "fake" by the discriminator

        - Originally, an ANN is used for the generator. Discriminator uses convolution to generate code from the generated images and then compares it to original
    - Variational Autoencoder (VAE)
        - Learns probability distribution of input data and generates new similar output
    - Natural Language Processing (based on variable classifier)
        - Cleaning from stop words and delimiters, retrieving only word stems
        - Bags of words = sparse (= lots of zeros) matrix with frequencies of all words
        - Google's word2vec model calculates the semantic distance of words in a vector space
- Reinforcement Learning / Associative Learning
    - How software agents take actions in a simulated or real environment in order to accumulate a predefined reward (compare game theory)
    - For sequential decision making, maps reward optimization of an agent in an environment
    - Model-based RL:
        - Transition function (predicting next state from current state based on action taken) and reward function (how much reward in any state)
        - Markov Decision Process: State (evaluated in state-value transition function) – Action (evaluated in action-value transition function, optimized based on policy) – Reward (based on value functions and derived Bellman equation which computes optimal state-value function from optimal action-value function) – New State etc. – cumulative reward
        - Bellman equation defines value of a state in terms of value of next state
        - Stationary (action depends only on last state) vs. deterministic (based on current state) policies vs. stochastic policy (random choice of actions and returns probabilities)
        - Gibbs sampling (for unknown common probability of two events)
            - MCMC (Monte Carlo Markov Chain) technique
                - Monte Carlo stands for repeated random sampling
                - A Markov Chain is a chain of evens where the probability of an event occurring depends only on the state attained from the previous event within the chain of events
            - Focused on posterior probability (a type of conditional probability)
            - Opposite is prior probability (probability before making references to any observations)
        - Flexible system, but slow
    - Model Free RL: Tries building up a lookup table (= policy) between action and cumulative future reward, where the right strategy for any given state can be derived from
        - Fast, but inflexible (if reward function changes, everything changes)

        - Temporal Difference Learning (TDL) predicts the expected value of a reward including discount factor (causes dopamine transmission in brain) at the end of a sequence of states
            - Kalman TD as most advanced (Kalman filter calculates join probability distribution of unknown variables)
        - Q Learning (most popular Model Free RL technique) creates a mapping from states to actions as a policy
            - Based on action-value function in Bellman equation (measures discounted future value of an action)
        - Deep Q Learning uses an ANN to approximate the Q function
    - Upper Confidence Bound (based on confidence interval, similar to boxplot)
    - Thompson Sampling (based on distributions)