Supervised: basics

1. **Preparing the dataset**
    - Get ready to use the same dataset as for the previous class
    - Create a new dataset containing only numerical columns: budget, popularity, revenue, runtime, vote_average, vote_count. Clean them by removing rows with NaN
    - From the release_date, extract 2 columns: year, and month. You can use to_datetime. Add those columns to the numerical dataset
    - Transform the first column, ‘adult’, into a boolean variable, whose value can be 0 or 1. You can use pandas get_dummies or do it manually (or with sklearn OneHotEncoder, less convenient)
    - Let’s say that we want to predict the column *popularity,* a score given by the platform the data was extracted from. Remove this column from the table, and keep it in a separate list. The order of elements in the list allows to match them with variables.
    - Split your dataset into a train and test set. You can do it manually or use sklearn train_test_split function. Keep for instance 1/3 as a test set.
2. **First prediction: linear**
    - Train a linear regression with sklearn. You can use the LinearRegression class, and method fit.
    - Compute the score we have seen, using corresponding functions in sklearn and the predict method of the linearRegression class. Do it first by using the train set, and then using the test set. Compare the difference.
    - To get a more intuitive idea of the performance, plot the relation between the target variable and the prediction (e.g., seaborn scatterplot, x=target variable, y=your prediction). With a perfect prediction, you should observe a diagonal line.
    - Check the coefficients coef_ and the intercept intercept_. Discuss with your peers about their interpretation. What about their magnitude? Sign?
3. **Decision Tree**
    - Train a decision tree (DecisionTreeRegressor) with sklearn, using default parameters, using unnormalized data with zero (normalization is useless with tree, and makes interpretation harder)

- Compute the scores, comparing between evaluation training and testing sets, with the linear approach.
- The default parameters make the tree overfit. Play with the parameters max_depth, min_samples_leaf, max_leaf_nodes, to try to limit the overfit (you should be able to improve over the linear regression)
- Train a tree small enough to be visualized, and plot it. You can use the built-in tool following the documents https://scikit-learn.org/stable/modules/tree.html#tree. The graphviz method usually gives the nicest results.
- Try to interpret the tree. Discuss with your peer.