Explanation for quiz questions

Question 1

Suppose we have a feature with all the values between 0 and 1 except few outliers larger than 1. What can help us to decrease outliers' influence on non-tree models?

Correct answers:

- Apply rank transform to the features. Yes, because after applying rank distance between all adjacent objects in a sorted array is 1, outliers now will be very close to other samples.
- Apply np.log1p(x) transform to the data. This transformation is non-linear and will move outliers relatively closer to other samples.
- Apply np.sqrt(x) transform to the data. This transformation is non-linear and will move outliers relatively closer to other samples.
- Winsorization. The main purpose of winsorization is to remove outliers by clipping feature's values.

Incorrect answers:

- StandardScaler. No, despite feature will be scaled, relative distances between outliers and other values still will be huge.
- MinMaxScaler. No, despite feature will be scaled, relative distances between outliers and other values still will be huge.

Question 2

Suppose we fit a tree-based model. In which cases label encoding can be better to use than one-hot encoding?

Correct answers:

- When categorical feature is ordinal. Correct! Label encoding can lead to better quality if it preserves correct order of values. In this case a split made by a tree will divide the feature to values 'lower' and 'higher' that the value chosen for this split.
- When we can come up with label encoder, that assigns close labels to similar (in terms of target) categories. Correct! First, in this case tree will achieve the same quality with less amount of splits, and second, this encoding will help to treat rare categories.
- When the number of categorical features in the dataset is huge. One-hot encoding a categorical feature with huge number of values can lead to (1) high memory consumption and (2) the case when non-categorical features are rarely used by model. You can deal with the 1st case if you employ sparse matrices. The 2nd case can occur if you build a tree using only a subset of features. For example, if you have 9 numeric features and 1 categorical with 100 unique values and you one-hot-encoded that categorical feature, you will get 109 features. If a tree is built with only a subset of features, initial 9 numeric features will rarely be used. In this case, you can increase the parameter controlling size of this subset. In xgboost it is called *colsample_bytree,* in sklearn's Random Forest *max_features.*

Incorrect answers: None

Question 3

Suppose we fit a tree-based model on several categorical features. In which cases applying one-hot encoding can be better to use than label-encoding?

Correct answers:

- If target dependence on the label encoded feature is very non-linear, i.e. values that are close to each other in the label encode feature correspond to target values that aren't close. Correct! If this feature is important, a tree would try to make a lot of splits and select each feature' value in a category on its own. But because tree is build in a greedy way, it can be hard to select one important value in label encoded vector. This won't be the problem if you use OHE.

Incorrect answers:

- When the feature have only two unique values. Incorrect. In this case both one-hot encoding and label encoding will produce similar columns.

Question 4

Suppose we have a categorical feature and a linear model. We need to somehow encode this feature. Which of the following statements are true?

Correct answers:

- Depending on the dataset either of label encoder or one-hot encoder could be better. Correct! Although one-hot-encoding is usually gives better results in this case, we can come up with examples when one-hot-encoded feature will not lead to a better performance of a linear model.

Incorrect answers:

- Label encoding is always better than one-hot encoding. Incorrect. Usually the dependence between the feature and the target is non-linear. In this case a linear model will not be able to utilize Label Encoded feature efficiently.
- One-hot encoding is always better than label encoding. Incorrect. Consider the toy example when the label encoded feature and the target are equal. In this case a linear model on this feature will have the perfect quality.