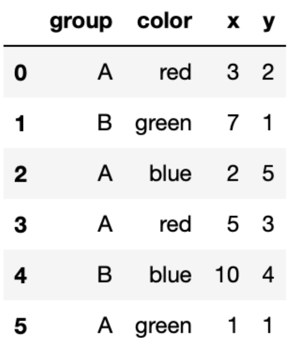
DSC 80 Discussion 8 Worksheet

READ THIS if you're completing this worksheet to submit to Gradescope for discussion credit:

- You don't need to fill in your answers directly on this worksheet, but all **7** questions are numbered, so if you're writing your answers elsewhere, make sure it's clear what's answering what.
- Either way, your responses should include **at least a couple sentences of explanation for every question** in order to receive credit.
- Submit your work to Gradescope by **Tuesday, May 28, by 11:59 PM**. There will be no extensions or late submissions accepted for this assignment. (It's optional, anyhow.)
- If you're having trouble, you can watch the Friday discussion podcast, although we might not get to every question during that time. If you really aren't sure, try your best – as long as you legitimately attempt every question, you'll get credit – or describe what you're having trouble with.

1 FA22 Final Problem 7



	group	color	x	y
0	A	red	3	2
1	B	green	7	1
2	A	blue	2	5
3	A	red	5	3
4	B	blue	10	4
5	A	green	1	1

Consider the dataframe to the left. Suppose you wish to use this data in a linear regression model. To do so, the `color` column must be encoded numerically.

Problem 1.1. True or False: a meaningful way to numerically encode the `color` column is to replace each string by its index in the alphabetic ordering of the colors. That is, to replace `blue` by 1, `green` by 2, and `red` by 3.

[ ] True
[ ] False

Problem 1.2. `scikit-learn`'s `OneHotEncoder` module has a keyword called `drop=first`, which the documentation says will ”drop the first category in each feature.” What's the purpose of this keyword, and will using it lead to a worse linear classifier?

2 FA22 Final Problem 8

Problem 2.1. Suppose you split a data set into a training set and a test set. You train a classifier on the training set and test it on the test set. **True or False**: the training accuracy must be higher than the test accuracy.

`[ ] True [ ] False`

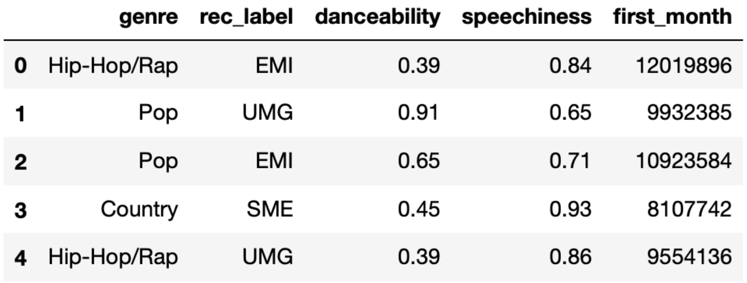
Problem 2.2. Suppose you train a model, but achieve much lower training and test accuracies than you expect. When you look at the data and make predictions yourself, you are easily able to achieve higher train and test accuracies. What should be done to improve the performance of the model?

Note: You haven't learned about decision trees yet (basically, just imagine a flow-chart), but for this question, all you need to know is that increasing `max_depth` *increases the complexity of your model.*

[ ] Decrease the `max_depth` hyperparameter; the model is "overfitting".
[ ] Increase the `max_depth` hyperparameter; the model is "underfitting".

3 SP22 Final Problem 10

The DataFrame `new_releases` contains the following information for songs that were recently released. The first few rows are shown below.



	genre	rec_label	danceability	speechiness	first_month
0	Hip-Hop/Rap	EMI	0.39	0.84	12019896
1	Pop	UMG	0.91	0.65	9932385
2	Pop	EMI	0.65	0.71	10923584
3	Country	SME	0.45	0.93	8107742
4	Hip-Hop/Rap	UMG	0.39	0.86	9554136

- `genre`: one of the following five possibilities: `Hip-Hop/Rap`, `Pop`, `Country`, `Alternative`, or `International`
- `rec_label`: the label that released the song (one of the following 4: `EMI`, `SME`, `UMG`, or `WMG`)

- `danceability`: how easy the song is to dance to, according to the Spotify API (between 0 and 1)
- `speechiness`: what proportion of the song is made up of spoken words, according to the Spotify API
- `first_month`: the number of total streams the song had on Spotify in the first month it was released

To start, we conduct a train-test split, splitting `new_releases` into `X_train`, `X_test`, `y_train`, and `y_test`. We first fit a linear model to the training data that only uses `danceability`, and call this model `lr_one`.

Problem 3.1. True or False: If `lr_one.score(X_train, y_train)` is much lower than `lr_one.score(X_test, y_test)`, it is likely that `lr_one` overfit to the training data.

[ ] True [ ] False

```
>>> X_train.shape[0]
50
>>> np.sum((y_train - lr_one.predict(X_train)) ** 2)
500000 # five hundred thousand
```

Problem 3.2. Given this output, what is `lr_one`'s training RMSE? Give your answer as an integer.

Now, suppose we fit one more linear model (with an intercept term) to the training data:

- Model 2 (`lr_no_drop`): Uses `danceability` and `speechiness` as-is, and one-hot encodes `genre` and `rec_label`, using `OneHotEncoder()`. (Note the lack of the `drop_first=True` keyword.)

Suppose we are given the following coefficients in Model 2:

- The coefficient on `genre_Pop` is 2000.
- The coefficient on `genre_Country` is 1000.
- The coefficient on `danceability` is $10^6 = 1,000,000$

Problem 3.3. Daisy and Billy are two artists signed to the same `rec_label` who each just released a new song with the same `speechiness`. Daisy is a `Pop` artist while Billy is a `Country` artist.

Model 2 predicted that Daisy's song and Billy's song will have the same `first_month` streams. What is the absolute difference between Daisy's song's `danceability` and Billy's song's `danceability`? Give your answer as a simplified fraction.