Feature Selection

How to reduce features and combat overfitting

What will be covered today

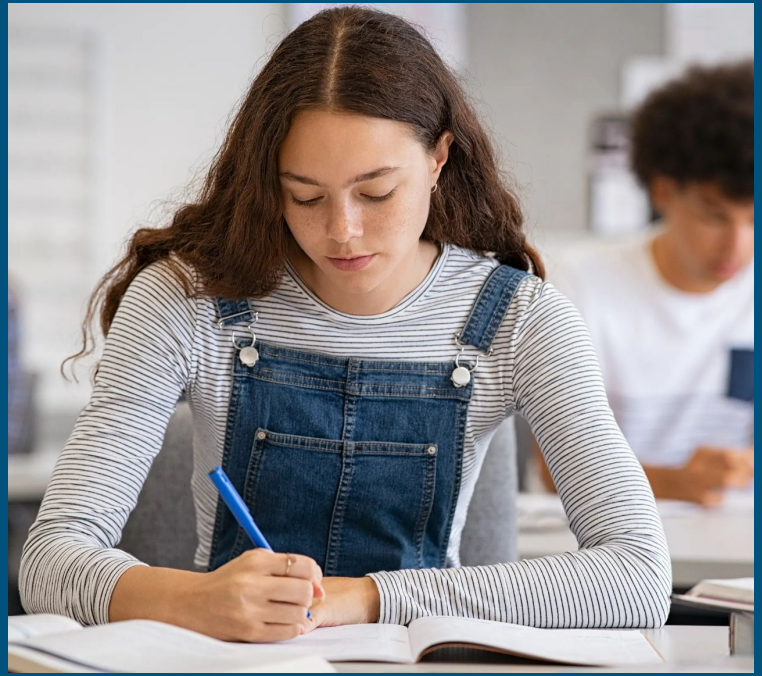
- Brief Discussion on Feature Engineering
- Feature Selection
  - What is it and how does it help us?
  - Removing features that don’t contribute
  - Checking correlation between all columns
  - Heatmaps (my favorite)
  - Keeping the features that have the highest correlations

Let’s think about features

- Features represent different aspects of our data
- Features can have information

Features

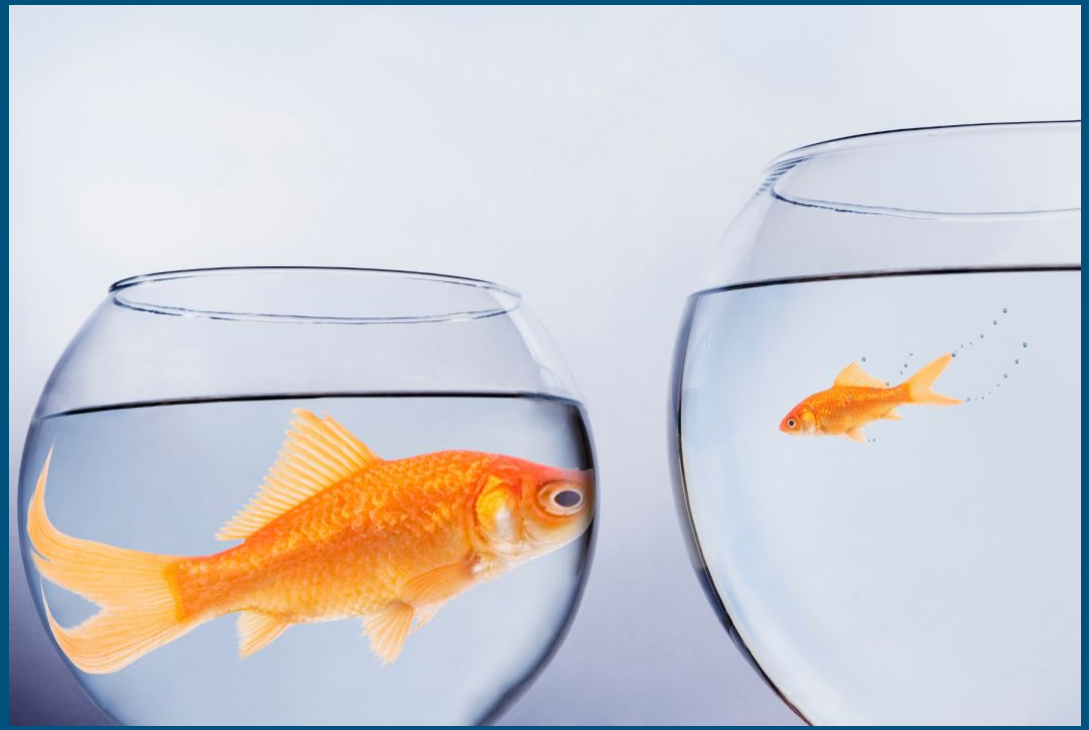
- Are all features equally important?
- Ex:
  - Predicting grades given
    - Major, year, amt of credit hrs, sex, height, hrs of study/week, id number



Features

- They don’t all given same amount/quality of info
- Is it always better to have more features?
  - Why or why not?

Bigger is not NECESSARILY Better

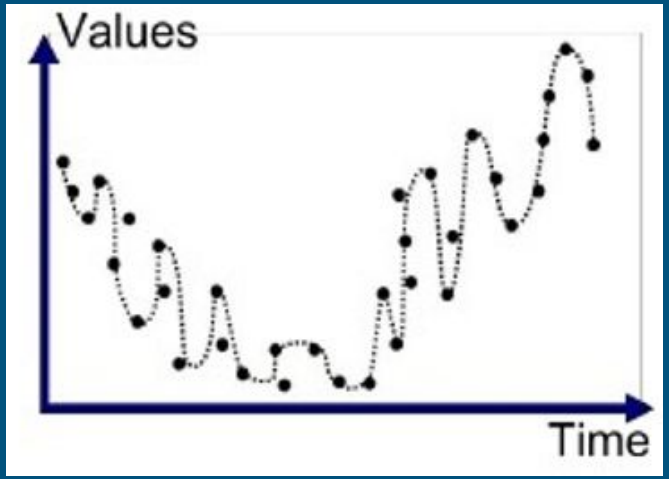


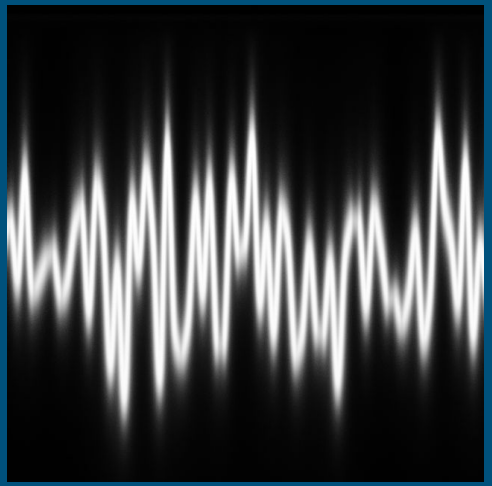
Bigger is not NECESSARILY Better

- Student grades
  - Predicting grades given
    - Major, year, amt of credit hrs, sex, height, hrs of study/week, id number

Too many features

- Leads to overfitting
- Introduces extra noise into the dataset





What about rows?

- Is it always better to have more rows?
  - Why or why not?

More rows are always better

How do we decide what features to keep?

1. Feature Engineering
2. Feature Selection

Feature Engineering

- You construct new features from the ones you have
- You can makes one that make more sense

Ex: City Happiness

- We had city feature
  - What was annoying about that?

Ex: City Happiness

- We had city feature
  - What was annoying about that?
    - We had to one hot encode it- got lots of columns
    - What if we wanted to make a prediction on a city that was not in the dataset? We couldn’t do it!

Feature Engineering Example

- What if we converted every city entry into
  - Col for latitude
  - Col for longitude
- How would this benefit us?

Feature Engineering Example

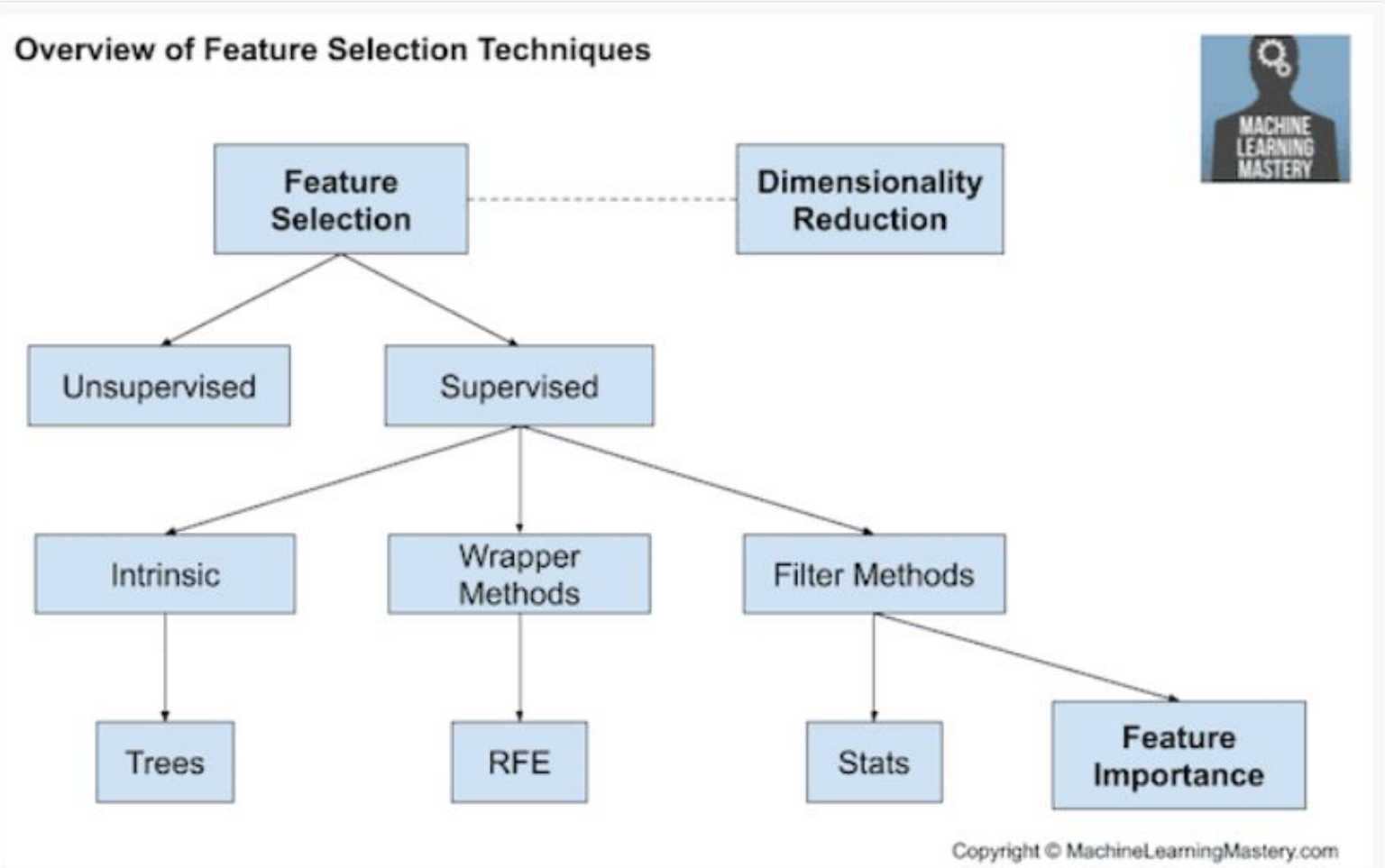
- How would this benefit us?
  - No one hot encoding needed!
- We might could interpolate happiness for other cities that were not in the dataset based on their latitude and longitude

Feature Selection

- We will focus on this for today
- Feature Selection- where you look at the features you have and you just select the ones you want to keep
  - Can do this manually (domain expertise)
  - Can use statistics
- Let’s explore!

Feature Selection

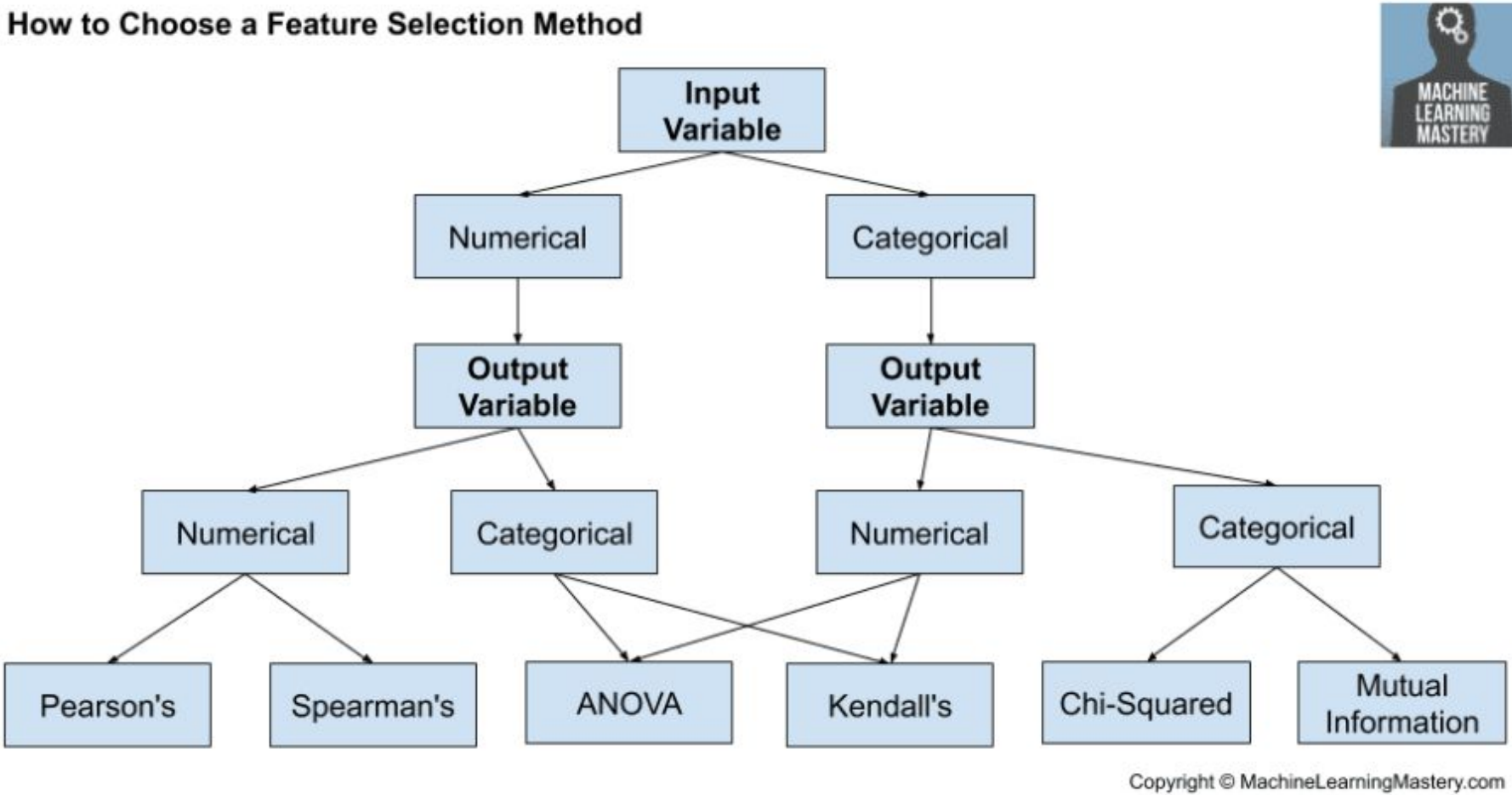
- Talk quickly about some definitions
  - Supervised vs unsupervised
    - Looks at target vs doesnt
  - Filter methods
    - Filters out features based off of stats
  - Wrapper methods
    - You try training models on different features and keep the best
  - Intrinsic
    - Some models will naturally start to ignore features as they train (trees and neural nets)- we can help this process with regularization



Overview of Feature Selection Techniques

Something to note...

- When doing stats, different stats are used for numerical features vs categorical.
- Dont worry about this right now, we will cover it, but be aware



How to Choose Feature Selection Methods For Machine Learning

Lets begin

- Open up sample code
  - Start with Taylor Swift example
    - Numerical features and numerical outputs
- Start off by talking about dataset
- Talk about what is in starter file

Let’s start by evaluating our models

- Let’s write down and record how our models do

Let’s look through our dataset

- And features look obviously weird or odd? Anything stand out?

Swiftiness Col

- It is a measure of how Swifty the song is
- But all songs are listed as 100
- Note: This was a fake column I put in , but you often get things like this

Columns with low variance

- If there is a col where almost all rows have the same value, how useful is that feature to a machine learning model?

Columns with low variance

- If there is a col where almost all rows have the same value, how useful is that feature to a machine learning model?
  - It’s NOT!
- It doesn’t give the model any info to discriminate between data points and the corresponding target

Ex: Predicting Grades

- If using student data….
- Ask if you are excited about spring break to each student
  - Will it easy for me to identify a pattern if everyone is looking forward to Spring Break?

So let’s get rid of that column

- We could do this manually, but it is not always obvious
- Enter, the Variance Threshold Test!
  - Look at code
- NOTE: When should this column be removed in the pipeline?

Variance Threshold

- Can compare the variance of each column to a threshold
- Should before training your models, can be done before or after splitting the data
  - As long as you remove the same column from both your training set and your testing set.

Recheck models and their performance

- How did these results perform compared to before?

What’s next? How do we know what other features to remove?

- What if we found the correlation between each feature and the target?

Lets do this!

- Should be done before splitting the data so we can compare the features to the target as well.
- Look at code
  - Everything is based on df.corr()

How to view correlation matrix?

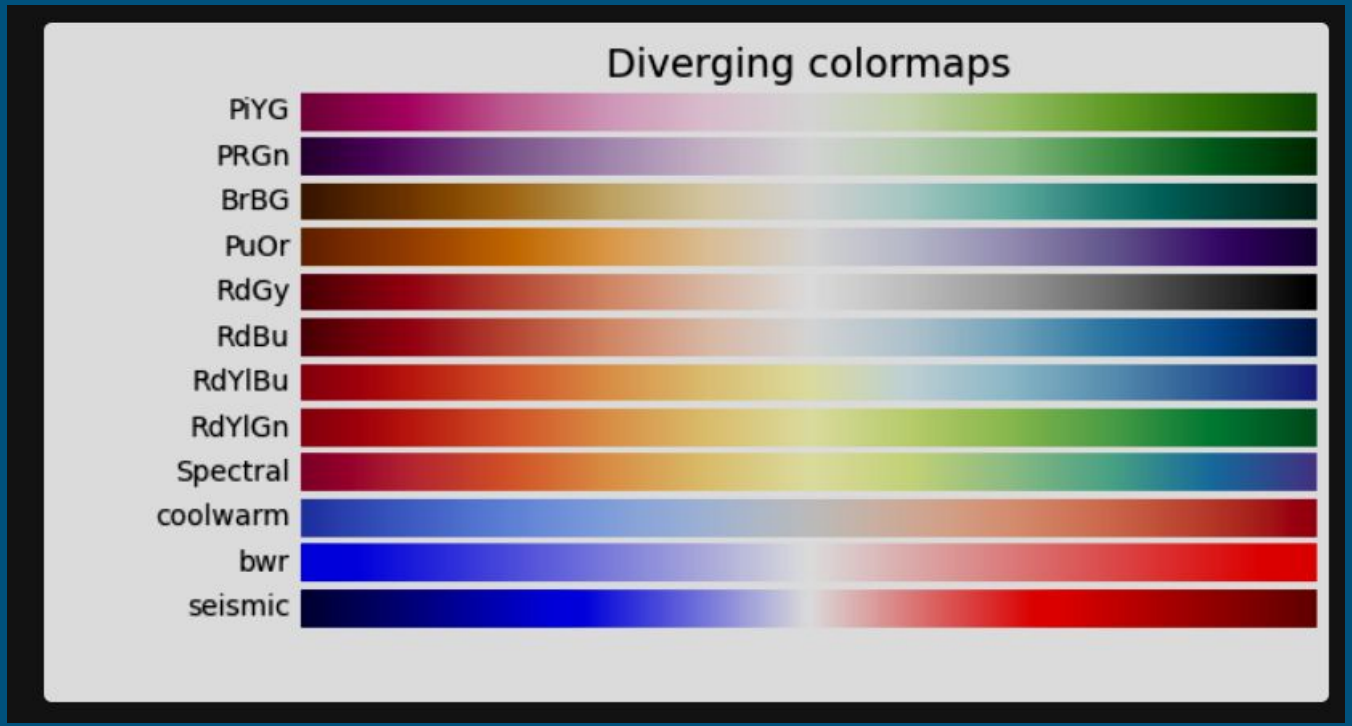
- Heatmaps!
  - What is that?

Heatmap


Correlation Coefficient
1.000
0.6000
0.2000
-0.2000
-0.6000
-1.000
Petal Width
Petal Length
Sepal Width
Sepal Length
0.818
-0.366
0.963
1
0.872
-0.428
1
0.963
-0.118
1
-0.428
-0.366
1
-0.118
0.872
0.818
Sepal Length
Sepal Width
Petal Length
Petal Width

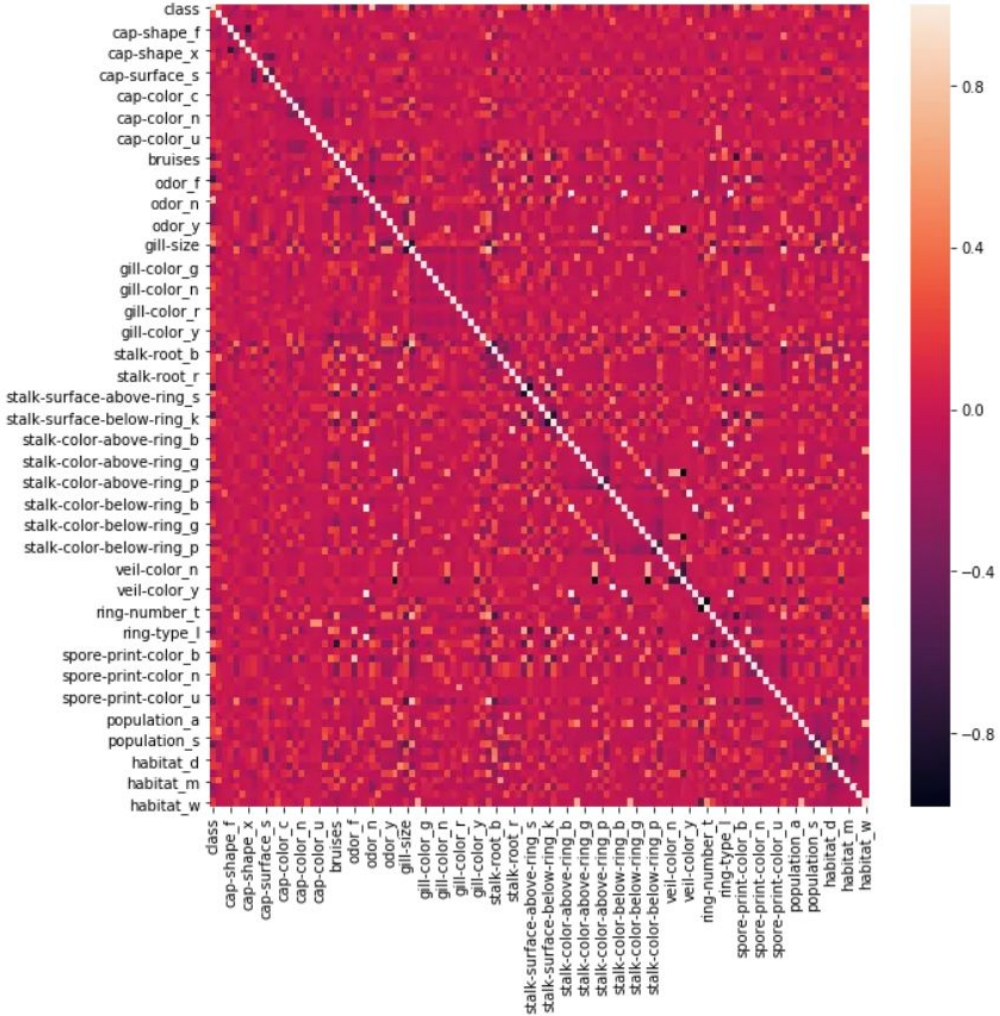
Go through process of making heat maps

Seaborn and Matplotlib color maps



Heatmap question

- What would happen if you did one hot encoding and did a heatmap of that?


class
cap-shape_f
cap-shape_x
cap-surface_s
cap-color_c
cap-color_n
cap-color_u
bruises
odor_f
odor_n
odor_y
gill-size
gill-color_g
gill-color_n
gill-color_r
gill-color_y
stalk-root_b
stalk-root_r
stalk-surface-above-ring_s
stalk-surface-below-ring_k
stalk-color-above-ring_b
stalk-color-above-ring_g
stalk-color-above-ring_p
stalk-color-below-ring_b
stalk-color-below-ring_g
stalk-color-below-ring_p
veil-color_n
veil-color_y
ring-number_t
ring-type_l
spore-print-color_b
spore-print-color_n
spore-print-color_u
population_a
population_s
habitat_d
habitat_m
habitat_w
0.8
0.4
0.0
−0.4
−0.8

Heatmap disadvantage

- Not great at comparing large amounts of columns
  - At least for the human eye

Questions

- What features seem most correlated with target var?
  - What does this mean?

Questions

- What features seem most correlated with target var?
  - What does this mean?
    - The higher the correlation, the more likely it is that a change in this feature will result in a change in the target

Questions

- What features are most correlated with each other?
  - Why do we care? What does this mean?

Questions

- What features are most correlated with each other?
  - Why do we care? What does this mean?
    - If some features are highly correlated, maybe we could remove some of them
    - That makes it where we are not including the same info multiple times
    - This reduces noise and bias

If features are highly correlated...

- We can manually remove them!

How do we keep the features that are most related to the target?

- Show code for SelectKBest with Pearson Coefficient

Check model’s performance again

- How did this do compared to before?

Checking correlation between numerical and categorical data

- numerical feature - numerical target
  - Pearson Correlation
- numerical feature - categorical target
  - ANOVA
- categorical feature - numerical target
  - ANOVA
- categorical feature - categorical target
  - Chi-squared

Numerical Feature - Numerical Target

```
# f_regression is Pearson Correlation
from sklearn.feature_selection import SelectKBest, f_regression

# score the different features and record it

fs = SelectKBest(score_func=f_regression, k=6)
fs.fit(xTrain, yTrain)
# Get columns to keep and create new dataframe with those only
indices = fs.get_support(indices=True)
xTrain = xTrain.iloc[:,indices]
xTest = xTest.iloc[:,indices]

xTrain.head()
```

Numerical Feature - Categorical Target

```
# f_classif is an ANOVA test
from sklearn.feature_selection import SelectKBest, f_classif

# score the different features and record it
fs = SelectKBest(score_func=f_classif, k=6)
fs.fit(xTrain, yTrain)
# Get columns to keep and create new dataframe with those only
indices = fs.get_support(indices=True)
xTrain = xTrain.iloc[:,indices]
xTest = xTest.iloc[:,indices]

xTrain.head()
```

Categorical Feature - Numerical Target

```
# f_classif is an ANOVA test
from sklearn.feature_selection import SelectKBest, f_classif

# score the different features and record it
fs = SelectKBest(score_func=f_classif, k=6)
fs.fit(xTrain, yTrain)
# Get columns to keep and create new dataframe with those only
indices = fs.get_support(indices=True)
xTrain = xTrain.iloc[:,indices]
xTest = xTest.iloc[:,indices]

xTrain.head()
```

Categorical Feature - Categorical Target

```
1  # chi2 is chi squared test
2  from sklearn.feature_selection import SelectKBest, chi2
3
4  # score the different features and record it
5  fs = SelectKBest(score_func=chi2, k=6)
6  fs.fit(xTrain, yTrain)
7  # Get columns to keep and create new dataframe with those only
8  indices = fs.get_support(indices=True)
9  xTrain = xTrain.iloc[:,indices]
10 xTest = xTest.iloc[:,indices]
11
12 xTrain.head()
```

The methods we talked about so far are called Filter Methods

- They filter out features based off of stats!

Lets now talk about Wrapper Methods for Feature Selection

- We’ll focus on one example
- Recursive Feature Elimination
  - RFE

How does it work?

How does it work?

1. You give the RFE object a machine learning model and it will train on all the features.
2. It ranks the features based on how important they were in making the model
3. It drops the least important feature
4. Repeats until it ends with the specified number of features

Wrapper methods vs Filter methods

- Training repeatedly pros and cons over using stats
  - What do you think?

Wrapper methods vs Filter methods

- Pros
  - Avoids having repeating information from features that share correlation with target
  - Can have higher confidence that it will lead to better performance because it is empirical

Wrapper methods vs Filter methods

- Cons
  - Takes more time then doing statistical tests
  - It takes orders of magnitude more time to do feature selection then it would to train your model

Code example

- Lets try some things and see how it goes!