Goal: Given an input pokemon, how can we classify what it’s main type is?

* Taking out unnecessary columns
  + Index, pokedex\_number, german\_name, japanese\_name, generation, height, weight, abilities\_number, ability\_1, ability\_2, ability\_hidden, catch\_rate, base\_friendship, base\_experience, growth\_rate, egg\_type\_1, egg\_type\_2, percentage\_male, egg\_cycles
  + df.drop(labels = [ ‘ ‘ , ]
* Check missing values - Yuet
  + If there aren't too many then we can manually fill it out
* Calculate the type\_weakness\_score - Mir
  + Add up all of the effectiveness values
  + Lower = better
  + Delete the old columns
* Two datasets - one with outlier and one without - Abby
  + Using total points to identify outliers
  + Remove total points afterwards.
* Normalization - Tanvir
  + Z-score

4/30/2021

* ~~Average the different forms of each pokemon entry~~
* Reinclude height, weight, egg\_cycles, percentage\_male, egg\_type\_1, egg\_type\_2, egg\_type\_number, growth\_rate, base\_experience, base\_experience, status, abilities\_number
* Get rid of against\_stat - DONE
* Normalization - Tanvir
* Test out correlation against all 18 pokemon types- (The point of this is to see what features we should keep, focus on the features that we’re not so sure about keeping).
  + , 'status', 'species', 'type\_number', 'type\_1', 'type\_2' - Yuet
  + , 'height\_m', 'weight\_kg', 'total\_points', 'hp\_scaled', 'attack\_scaled', - Mir
  + 'defense\_scaled', 'sp\_attack\_scaled', 'sp\_defense\_scaled'', speed\_scaled', 'catch\_rate', - Abby
  + 'base\_experience', 'growth\_rate', 'egg\_type\_number', 'egg\_type\_1', 'egg\_type\_2', 'egg\_cycles' - Ari
* Start looking at classifer -
* Genders - what do?

Questions for prof

* ~~Can a classifier predict 2 features at the same time?~~
  + ~~Should we duplicate the data?~~
* Should we combine the 2 columns?
* We have 18 class labels.
* ~~Ideas for classifier training?~~
* Using the correlation data, how to tell difference between each feature (like average?)
  + How can we use the correlation data to tell us which feature to keep and not keep?
* How do we find the correlation between different features if one of the features is categorical and not numerical?
* Dealing with 2 types
  + Idea: we duplicate all the pokemon with 2 types. The first copy of the pokemon has only type 1, while the second copy has only type 2
    - Duplication shows that both types are equally important
  + Idea: Just have the code check both Type\_1 and type\_2, the pokemon count as X type if it has that category in either columns.

Decision tree - Abby

Baiyes - Tanvir

Feature Selection - Mir

k-nearest neightbor - Ari

5/03

* Manually entered missing data for base\_experience
* Duplicated data

5/06 meeting w Prof

* Calculate how many in each type
  + 1. Use just type 1 to show density
  + 2. After duplication, get another distribution for each type
  + 3. Remove pokemons that have 2 types
    - Is there a type that every pokemon has 2 types?
  + Train model on 3 diff sets and see which performance is good
    - Bc performance is bad w duplication - gave 1 guy w same stats 2 diff labels - confuses classifier and cannot tell them apart bc features are the same
    - Compare performance - same model on 3 diff data sets
  + Data sets:

1. Only look at type 1 - just delete type\_2 (no duplication)
   * + - Will have 18 diff types
       - Assumption - type 1 is primary type and more important
       - Don't want to have 2 labels
2. Remove all pokemon that have 2 types
   * + - Will have 552 pokemon (1044-552)
       - Will this change # of types?
       - It will change distribution in each type
3. Duplicate data (what we already did)

* For each before training --> show class distribution
* In excel, can sort data by type\_1
* 18 diff types:
  + If one type has very few members, remove this type
  + Model cannot train well without enough information
  + How to set threshold ? (set bar to eliminate)
    - Focus on top 10 types\* - she recommends working with 10 types
    - Model doesn’t usually train well with sooo many types
    - If they have roughly equal amounts of pokemon - then drop limit a little
      * 1st set limit to having under 5 pokemon
        + If a type has less than 5 members - remove from data set
      * Use bar plot to make decision
* Binning Question:
  + How to decide on # bins?
  + Depends on what column
  + Ex. age - usually young, middle, old, or by 10s
  + No one correct answer
  + Depends on distribution if numerical value
* Feature selection

1. Use common sense ex. ‘Japanse\_name’
2. Run feature selection to see if related or not

* Before:
  + Binning
  + make sure columns / features are same type - all numerical or all categorical
* 2 more copies: \*this is before feature selection
  + 1. All numerical - 1 hot encoding
  + 2. Convert everything to categorical values
  + (this makes 6 copies total i think)
    - Run decision tree 6 diff times
    - See which way is best - pick that one
    - Compare, see if anything changes



Tasks:

* Update Proposal

**ORDER OF OPERATION**

1. Drop features we obviously don’t need like names.
2. Create 3 cuts of the dataset. #1 has uses only type\_1, drop type\_2. #2 delete all pokemon with type\_2. #3 Do our duping thing.
3. Create 2 variations of the 3 sets, #A uses all numerical so convert categorical to binary via one hot encoding. #B uses all categorical so convert numerical to categorical bins.
4. Figure out the datapoint distribution per type for each of the 3 main dataset. Kick out all pokemons with types that are below a threshold. We will figure out this threshold after we examine the distribution. At most ten types of classes.
5. Do feature selection for the main cuts of datasets, try 2 feature selection methods + different K values. We’ll decide on what models to test how good they are.

* Models to try:
* DSTree Entropy criterion, DSTree Gini criterion, ten-fold cross-validation
* GaussianNB, MultinomialNB, BernoulliNB
* KNeighborsClassifier, RadiusNeighborsClassifier

1. After feature selection, drop the columns that were not used to produce the best accuracy.
2. **POSSIBLY WHAT WE MIGHT DO:** We will run all the models we can think with the feature selection for each of the three main cuts of the dataset.
3. We will pick the TOP FIVE performing models out of ALL of them and divide up the fine tuning among us.

**FINDINGS**

* The only outlier in terms of stats and height Eternatus’ Eternamax form with almost double the total points as the second-highest pokemon and with a whopping height of 100 meters

Suggestions

* Trim decision tree