

CLASSIFICATION

Dr. Brian Mc Ginley



CLASSIFICATION

- Now some different problems:
 - With regression, we assumed we were taking input and producing a quantity: what number (continuous value) do we think y will be (close to) for a particular input?
 - Many ML problems have a more limited set of outputs. Often the question is more like: which group is this input (mostly likely) part of?
- These are classification problems.
- Our outputs will be restricted to the possible classes for our problem. i.e. for n samples with K categories, we predict a vector of n values chosen from C_1, C_2, \dots, C_k .
- Our inputs will stay the same: an $n \times m$ matrix of numbers (for n samples and m features).



CLASSIFICATION

- Some problems that can be seen as classification:
 - Is this email spam? (2 categories)
 - Is this credit card charge fraud? (2 categories)
 - Is this biological sample malignant or benign? (2 categories)
 - What (English) word did the user say into the microphone? ($\approx 470k$ categories)
 - What date will Easter fall on next year? ($\ll 365$ categories) Or is that regression?
 - How many people are in this picture? Who are the people?
 - Given an RGB colour value, what colour word describes it?
 - Given weather observations, can you guess what city are we looking at?



MODELLING

- There are many ways to model classification.
- All do the same basic job:
 1. Train on known data
 2. Make predictions about correct category for other unseen inputs.

```
model = SomeClassifierModel(...)\nmodel.fit(X_train, y_train)\ny_predicted = model.predict(X_test)\nsomehowcompareresults(y_test, y_predicted)
```

- (Often use validation to pick the best form of the Classifier, previously used

```
model = LinearRegression(...)
```

- it all takes the same basic idea in code)



EVALUATING CLASSIFICATION PERFORMANCE

- For evaluating Classification algorithms often people just see how accurate the predictions are on the test set

```
y_pred = model.predict(X_test)
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

- Will give a score (< 1) which is the fraction of how many predictions were correct.
- This stores the y predictions separately (which may be useful later) but the model has a method itself that can be used if you just want the accuracy score

```
model.score(X_test, y_test)
```

- This one line does the predicting and comparing of predictions in the same step. We will come back to this later, about alternatives.



BINARY CLASSIFICATION

- Definition:
- A Binary classification problem has:
 - Negative Class
 - Positive Class

$$y \in [0, 1] \text{ or } y \in [-1, 1]$$

- We'll consider Multi-class problems later.
- Some models can directly address multi-class (Naive Bayes, k -nearest neighbours), while some are inherently binary but can be “extended” (SVM, Logistic Regression)



CATEGORICAL RESPONSE VARIABLE

- Methods used when the response variable (\hat{y}) has a distinct number of possible outcomes:
 - a customer buys or does not buy a product,
 - a patient lives or dies,
 - or a candidate accepts or declines a job offer.
- In general, we call the two outcomes of the response variable "success" and "failure" and represent them by 1 and 0.
 - The mean is then the proportion of 1s.

