1. Making Predictions Using Supervised Models
2. How Do We Know if a Model is Doing Well or Not?
3. Fitting a balanced model
4. Feature selection
5. Training / Cross-Validation/ Test
6. Making Predictions Using Supervised Models

* **Supervised Learning:** Using data to try to predict a specific outcome
* **Regression:** Predicting a continuous variable
  + **Continuous variable:** number you can get by measuring (e.g. # rushing yards)
* **Classification:** Predicting a categorical variable
  + **Categorical variable:** variable that comes in fixed sets (e.g. win vs. lose)

Simple Examples (based on y=mx+b formula for a line)

* **Regression:** [Passing Yards in 2022] = **m**\*[Passing Yards in 2021] + **b**
  + Need to learn best value for **m** and **b**
* **Classification:** [Win / Lose this week] = **m** \* [Team Rushing Yards Last Week] + **b**
  + You can encode win/loss into numbers by setting win as 1 and loss as 0
  + You also can set a cutoff to get data into 0/1 form
    - For example, any value above 0.5 gets set to 1 and any value below is 0
  + Now you can use your data to learn the best **m** and **b** value

Chart, scatter chart

Description automatically generated

More complicated models use more variables

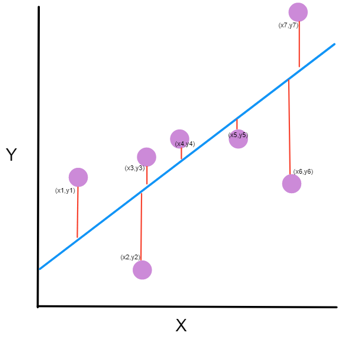
**Example 1:** [Passing Yards in 2022] = **m1**\*[P-yards in 2021] + **m2**\*[# P-TDs in 2021] + **b**

**Example 2:** [Hall of Fame (0/1)] = **m1**\*[years played] + **m2**\*[number of super bowls won] + **b**

**Teaching Task:**

* Teach the class what supervised learning, regression, and classification mean
* **Come with a different sports-related example of a regression and classification task to teach everyone the difference between them**

2. How do we know if a model is doing well or not?



When predicting an exact number, you evaluate the model by considering the **mean absolute error**.

Week 1: Predicted (20), Actual (30), Absolute error (|20-30| = 10)

Week 2: Predicted (26), Actual (20), Absolute error (|26-20| = 6)

Mean Absolute Error = 8 [because (10 + 6) / 2) ]

Some models try to predict an exact number (e.g. how many points will Titans score this week), while others try to predict categorical variable (e.g. Titans win or loss).

Table

Description automatically generated

When predicting a categorical variable, you consider whether predictions agree with true outcomes

**Accuracy:** What percentage of possible predictions were correct?

Example: (42 + 40) / (42 + 40 + 5 + 3) = 91%

**Teaching Task:**

* Explain how to evaluate a regression task using mean absolute error
* Explain how to evaluate a classification task using accuracy
* Explain why accuracy can be misleading **and come up with another sports example where just using accuracy would fail?**
* Introduce precision and recall as other ways to evaluate a model

Sometimes accuracy can be misleading. For example, let’s try to predict whether an NFL player will make the hall of fame using some statistics from their career. Only 1% of NFL players eventually make the hall of fame, so if you just created a model that predicts “No” every time, it would be 99% accurate. Therefore, other metrics are also used to evaluate classification tasks.

**Precision:** What percentage of players that the model predicted would make the hall of fame actually did?

**Recall:** What percentage of players who made the hall of fame did the model predict correctly?

3. The Goldilox Problem: Finding a Model that’s “Just Right”

In the story Goldilocks & the 3 Bears, Goldilocks tries 3 different bowls of porridge. The first one is too cold, the second is too hot, but the third is “just right.” When trying to train a model, it is very important that it’s just right and not too bad or too good.

Chart, scatter chart

Description automatically generated

Underfitting (the porridge is too cold)

When a model is underfitting, it does a poor job of learning from your data and is bad at predicting the outcome you care about.

**Example**: [Receiving yards in game] = 2\*[jersey number] – 20

Jersey number is not strongly related to how many yards a receiver will catch., so there are other better variables that can be used.

Overfitting (the porridge is too hot)

Weirdly enough, you also don’t want your model to be too accurate. If it fits too closely to your data, it probably will not work well on data it hasn’t seen.

**Example**: [Points Scored in Game] = 10 + 20 (if name is Luka Donicic) or -5 (if name is Ben Simmons), etc.

*You can add tons of variables to make model more accurate, but if you make it too specific to your data, it likely won’t generalize to new data.*

Diagram

Description automatically generated

Choosing a balanced model (the porridge is “just right”)

To avoid underfitting, understand the data & choose good variables

To avoid overfitting, there are two main techniques…

1. **Regularization**: penalize models that use too many variables
   1. Example: For every variable, you penalize accuracy by 5%
      1. Model A: 90% accuracy with 6 variables = score of 60
      2. Model B: 80% accuracy with 3 variables = score of 65
   2. Model B would be the choice because it has a higher score
2. **Feature Selection**: try to use as few variables as possible, as a long as the model is still accurate enough
   1. **Example**:choose model with the least variables that still gets 80% accuracy

Chart, scatter chart

Description automatically generated

**Teaching Task:**

* Explain overfitting and underfitting
* **Come up with a sports example for each**
* Teach the class about regularization and feature selection

A picture containing chart

Description automatically generated4. What Variables Should We Include in a Model?

**Recursive Feature Addition:**

* Train a model with all possible variables to get their importance (how strong the weight is in the model).
* Train model using best feature and then add one at a time in order of importance and keep the variable if it improves accuracy based on previous step

**Correlation analysis**:

* Want to find variables correlated with outcome (when variable goes up, outcome also goes up)
* Some variables may also correlate with each other, and in those cases you want to choose one of them that correlates best with outcome to lower the total number of variables in model

There are often many variables that can possibly be included in a model. For example, if we are trying to predict how many fantasy points a running back will score this season, we can consider “fantasy points scored last year”, “rushing yards last year”, “how easy/hard are their opponents this year”, etc

In general, **we want to use the simplest model that works “well”**. It is difficult to define well, but you could theoretically say, “I want the simplest model that’s 90% accurate”

Text

Description automatically generated

**Teaching Task:**

* Explain the concept of feature selection (i.e. choosing a what features to include)
* Describe correlation analysis and **come up with another sports example**
* Describe recursive feature addition

5. How do we train a model and evaluate performance

Not only do we want our models to be accurate on the data we have, but we also want them to work well on other data we haven’t seen before. If an algorithm can only predict fantasy points scored in 2021, but doesn’t work for 2022, then it isn’t very valuable.

To test this, we take some data that we have and leave it out of our training to pretend like it is new data. Then we can use this “left out” data for evaluation.

For typical machine learning problems, we split our data into 3 groups

**Training Set:** Data used to train the model and learn model parameters (e.g. m & b in y=mx+b)

**Cross Validation Set:** Data used to evaluate many different models and choose the best one

**Test Set:** Data used to evaluate the performance (e.g. accuracy) of our best model from CV

**Teaching task:**

* Explain the importance of models working on new data.
* Explain the definitions of training set, cross-validation set, and test set.
* Talk about 70/15/15 split and **come up with a sports example where data splitting could be important (like the example above)**.

Choosing how to split the data is actually very important and challenging. Very often, people will choose 70% of the data for training, 15% for cross validation and 15% for testing.

The more data used for training, the more that your algorithm can learn. Unfortunately, that leaves less data for cross-validation/testing, so you may believe your evaluation less.

*Example*: When trying to predict expected points that an NFL team will score, one could naturally say the training set will be all data from 2000-2015, the CV set will be data from 2016-2018, and the test set will be data from 2019-2021. Unfortunately, this would not work very well because the NFL has changed so much since 2000 to 2021 (there’s much more passing and scoring today). Therefore, it is important to make sure each group (train, CV, and test) include data from every year, every team, etc.