**Modeling questions:**

Explain what an OLS Regression is

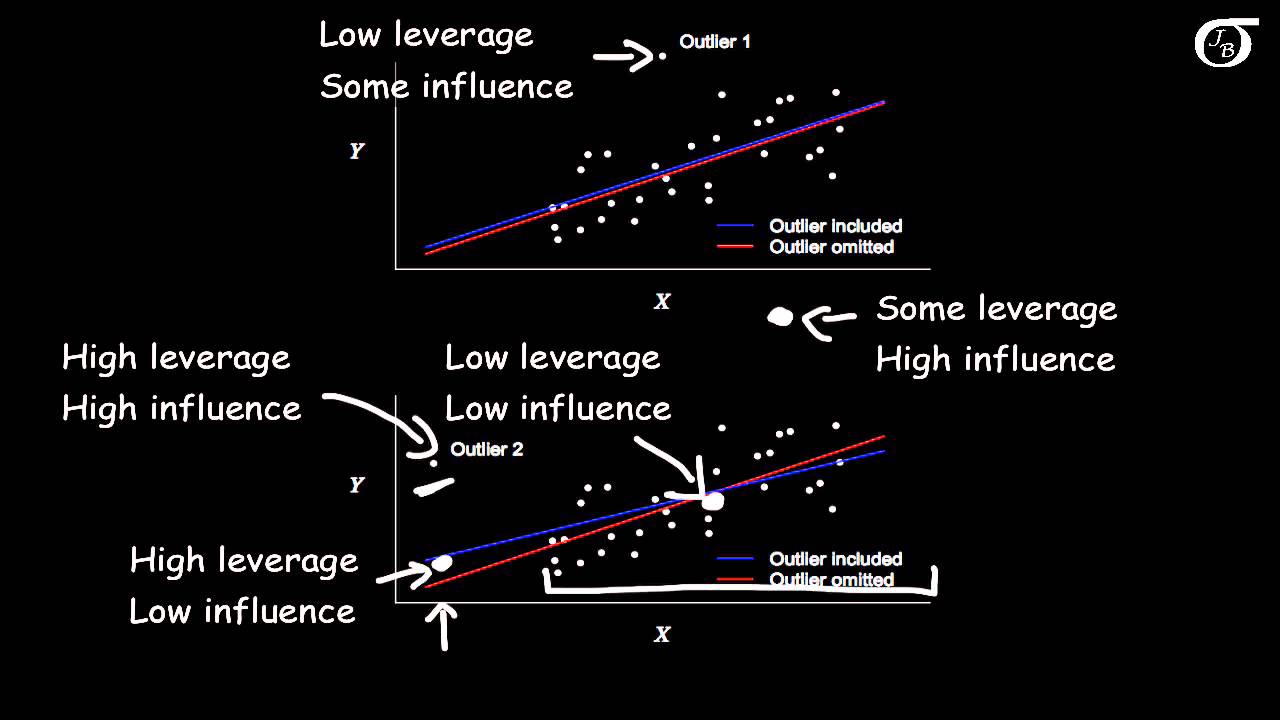
* OLS Regression stands for ordinary least square regression and it is a linear regression that takes the form of outcome = model + errori.
* The goal is to minimize the sum of errori2 (technically we want the sum of errori = 0 but the negative values makes it so that there are many possible lines to achieve this. Taking the square moves all the errors to positive and reveals one true minimum.

What are the OLS/Markov Assumptions?

* I was taught linear assumption, random sampling, no multicollinearity, Zero conditional mean (exogeneity), and homoskedasticity.
* **Linear assumption:** Assume our data is linear in the parameters we’ve chosen.
* **Random Sample assumption:** Independently and Identically distributed (IID) meaning our data comes from the same distribution but each data point is independent and is not affected by other data points
  + Independent error terms assumption (no autocorrelations) – Plot your error terms vs any kind of time measurement, or order of test, etc. You shouldn’t see any form of function or trend between the points, should be a block of random points.
  + Homoskedasticity assumption - can help you tell if your error terms are not part of the same distribution.
  + Normality of errors (not a big deal if you have a large sample) – Look at Q-Q Plot
* **Homoskedasticity assumption:** Your model has constant error variance. As you increase your fitted values, your residuals should not increase or decrease.
  + Plot your residuals vs. fitted values graph and see if the variance of the residuals funnel in or funnel out.
  + Breusch-Pagan Test.
* **No Multicollinearity assumption:** Each of your features should not have a perfectly linear relationship with each other. You interpret linear regressions as “holding everything else constant, how does x1 affect y?” This makes no logical sense of if x1 also affects x2.
  + Plot a scatterplot matrix and look at each of your feature’s relationship to each other.
* **Zero Conditional Mean/ Exogeneity assumption:** At each given x1, we want the average error to be 0. Look at the residuals vs. fitted values graph and you should see a straight line through zero.
  + This assumption tells you that you don’t have any omitted variables.
    - Causes you to over or under predict your coefficients
  + Also, that the relationship between x and y is 1 directional. (No Simultaneity).

What is Cook’s Distance?

* A method to check how much influence each data point has.
* If an outlier has a Cook’s Distance > 1, it means it has high leverage over your model. You should check out that outlier to see if it’s a valid data point.
* Distance – based on error of the prediction for the observation. Large distance means you have an outlier, but if the distance is large and the leverage is low, then the point won’t have much influence.
* Leverage – How far away is this data point from the average of all the data points? Think of a see-saw the further out you sit, the more leverage you have. You can have high leverage and low influence if you’re in line with the other points.





What is a Decision Tree? How does it work?

* Decision trees are a type of machine learning model used for both classification and regression by making sequential decisions looking at one feature at a time.
* For all features and all thresholds, calculate the information gain. Choose the one that offers the highest information gain and make that you splits. Then for each split do the same thing to determine the next split.

What is splitting the decision tree?

* To split the training data into two subsets based on a variable. Basically growing branches.

What is pruning the decision tree?

* Process to reduce the size of the tree by making a branch node into a leaf node and then throwing away all the leaf nodes underneath the original branch. This is a way to reduce the complexity of your model and help reduce overfitting.

What is Entropy?

* Measure of chaos or the measure of homogeneity of a sample set. If all data have the same label (fully homogeneous), entropy is at 0. When the data is split evenly between 2 labels, we have maximum entropy.
* b = number of possibly labels

A picture containing text, watch

Description automatically generated

What is Information Gain?

* Entropy of your original data set minus the average entropy of your two splits. You want your resulting splits to have as little entropy as possible.

What is Gini Index?

* A slightly simpler version of entropy that does not calculate logarithms which makes it faster to calculate but a little weaker than entropy. 1 – p1^2 – p2^2
* Proxy for how often a randomly chosen data point would be incorrectly labeled.

Pros/Cons of Decision Trees:

* Pros:
  + It’s inherently very interpretable. You can literally draw a tree after training.
  + It’s a nonlinear model. Not all systems are linear.
  + It can naturally handle categorical features, not just numeric ones.
  + Outliers do not really have an effect on your resulting classifier because each feature is split up into ranges based on information gain. Outliers therefore do not bias the decision-making process by the virtue of this process.
  + Prediction speed is pretty fast
  + Do not really need to pre-process the data as much as other models
* Cons:
  + Slow training time
  + Prone to overfitting if your tree doesn’t use pruning or has too many nodes.
  + Also prone to being unstable where some change in the data can cause a large change in the structure of the decision tree

What is a Random Forest? How does it work?

* It’s a type of ensemble learning model where a bunch of weak models are aggregated to form a more powerful model
* Can be used for both regression and classification.
* We grow multiple decision trees on bootstrapped training samples of the data. In a classifier, each decision tree will generate a prediction and in the end we tally the votes from each tree to get a majority prediction. In a regression, each decision tree will generate an output and then we take the average of these outputs as the final output.

What is Bootstrap Aggregating or Bagging?

* You train multiple decision trees by taking random samples of the data with replacement. This means one decision tree might have a certain data point occurring multiple times in its sample.
* In some cases (not sure if it’s always the case), there is random sampling done to decide which of the features to keep in each tree.
* You end up with multiple trees trained on different samples and during prediction, the results of every tree will be aggregated to give you a final result. Averaged for regression, tallied for classification.
* By training each tree separately on different samples, each tree might have high variance, but the overall random forest will have lower variance without increasing the bias. This helps avoid overfitting.
* Simply put, Bagging is when you aggregate the predictions of several base models trained on bootstrapped subsets of your training data in order to decrease the impact of overfitting.

What is out of bag error?

* It’s a way of estimating the test error of a bagged model.
* For example, you take row 1 of the data and you predict the results in only the decision trees that did not train with row 1. You then aggregate the results and see if you predicted row one correctly. You do this with every row of the data and see how many of the results were predicted correctly and incorrectly.

Pros/Cons of Random Forest

* Pros:
  + It is pretty interpretable but not as interpretable as a decision tree
  + It’s a nonlinear model.
  + It can naturally handle categorical features, not just numeric
  + Outliers do not impact your resulting model much.
  + Do not need to preprocess the data as much as other models
  + Works with both regression and classification
  + Can handle data with many features and actually provides the most impactful features
  + No longer unstable the way a single decision tree can be.
* Cons:
  + Slow training time
  + Still has potential to overfit but not nearly as much as a single Decision Tree.

What is Boosting?

* While Bagging trains many simple models in parallel, Boosting trains these models sequentially.
* Boosting is basically training weak models sequentially where each model tries to correct the previous model. For example, in AdaBoost, the weak learners are decision trees with a single split. In the first tree, all observations are weighted equally. The next tress will put more weight on the observations that were incorrectly predicted by the previous trees. In AdaBoost, the decision stumps that make better predictions have more say in the final result than other decision stumps.

KNN

* Uses entire data set as training set (no split for testing set)
* When an outcome is required for a new data instance, the KNN algorithm goes through the entire data set to find the k-nearest instances to the new instance, or the k number of instances most similar to the new record, and then outputs the mean of the outcomes (for a regression problem) or the mode (most frequent class) for a classification problem. The value of k is user-specified.
* The similarity between instances is calculated using measures such as Euclidean distance and Hamming distance.