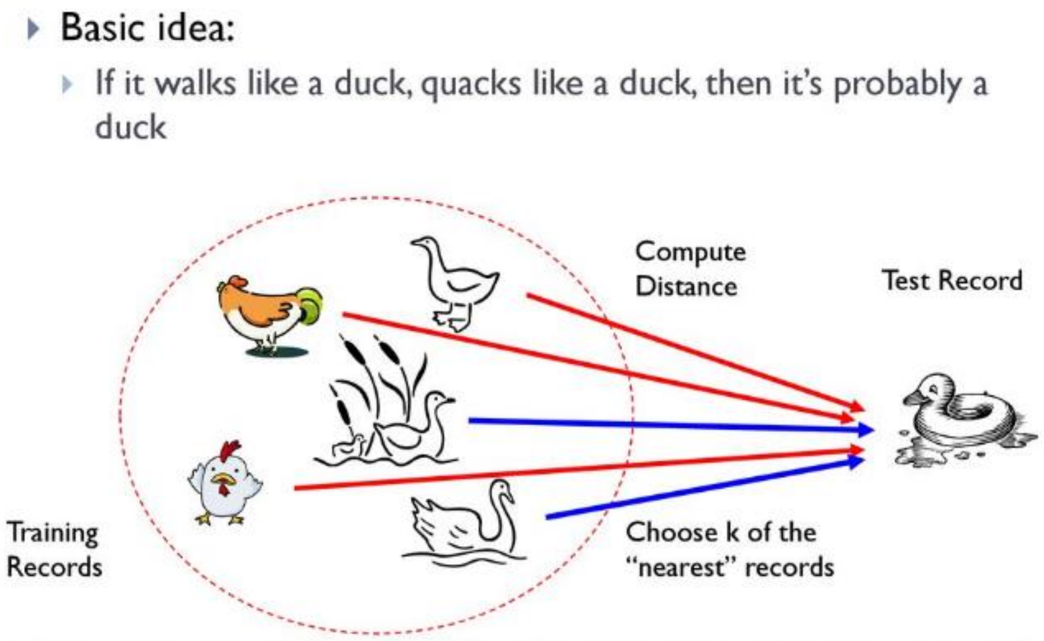
K Nearest Neighbors

- K Nearest Neighbors is a classification and regression algorithm that makes a prediction based upon the closest data points
- It involves comparing features for the new instance with the features from the instances we have already collected.
- We can quantify how similar two objects distance vectors are using either the Euclidean distance or the City Block (Manhattan) distance as metrics



- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck

Distance Metrics

- **Euclidean Distance**:

  Square-root of the sum of the squared differences between each characteristic

  - Example Euclidean distance between two objects (1 & 2) on 3 different characteristics (X, Y, and Z)
  - Distance = square-root $[(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2]$

- **City Block/Manhattan Distance**:

  Sum of the absolute differences between each characteristic in two vectors

  - Example city block distance between two objects (1 & 2) on 3 different characteristics (X, Y, and Z)
  - Distance = $|X_2 - X_1| + |Y_2 - Y_1| + |Z_2 - Z_1|$

Euclidean Distance

- Looks at **squared differences between each characteristic**
- **Sensitive to scaling** (small scale = small differences; big scale = big differences) = Need all characteristics to have similar
- **Not scale invariant** = converting cm to mm will produce different results
- **Gives equal emphasis to characteristics**: Absolute difference is squared between each characteristic
- Better for **lower dimensional data**: Because all characteristics are considered
- **Emphasizes large differences** between characteristics: Because the differences are squared. Susceptible to noise and random variation.

City Block/Manhattan Distance

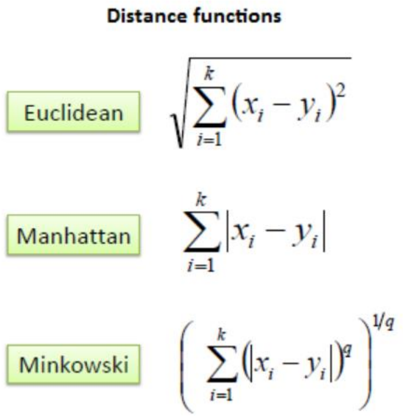
- Looks at **absolute differences between each characteristic**
- **Sensitive to scaling**: all characteristics need to have similar means and standard deviations.
- **Not scale invariant:** Converting cm to mm will produce different results
- **Gives equal emphasis to characteristics**: Absolute difference is calculated between each characteristic - Matches and Non-matches both matter for similarity
- Better for **higher dimensional data than Euclidean**
- **Describes number of mismatches** if vector values are coded as 1s and 0s
- **Dampens large differences** between characteristics: Because the differences are not squared, less susceptible to noise
- **Fast** to compute

Running the Algorithm

- **Prepare the dataset**
  - Each row is a different person/event
  - Each column is a variable (e.g., attributes of the outcome)
  - Only use variables that you think are highly predictive
  - Have one column be designated as the predictor variable
- **Standardize all variables** (gives variables equal importance)
- **Provide dataset to KNN algorithm**
- **Choose a value for K**
  - Too low = not enough generalizability
  - Too high = neighbors are no longer predictive
  - Usually chosen via cross-validation

K Nearest Neighbors (contd.)

- Algorithm Steps
  - ✓ For a given point, we calculate the distance to all the other points.
  - ✓ We then pick up the k closest points.
  - ✓ Calculate the probability of each class label given those points
  - ✓ The original point is classified with the label with the largest probability



Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

- Benefits
  - ✓ Training is very fast
  - ✓ Handles complex relationships between variables an outcomes.
  - ✓ Easy to calculate/explain

- Limitations
  - ✓ Slow to make predictions
  - ✓ Does not handle many predictors well
  - ✓ Must standardize variables first
  - ✓ Must have large datasets

Applications of KNN

Image Classification:

- Finding similar handwritten digits
- Finding similar menu items

Collaborative Filtering:

- Making automatic predictions about the interests of a user by collecting preferences or taste information from many users
- Detecting spams and frauds
- Recommendation Systems
- Determining health risks

Evaluating Prediction Errors

Discuss the different measures of quantifying the performance of regression and classification predictions

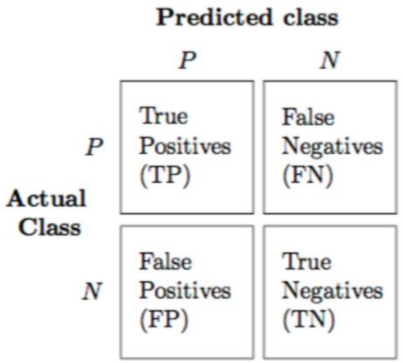
Goal- Low error and high generalizability

Regression Models

- R^2 : Describes the proportion of variation in the outcome that can be explained by the model.
  It is equal to the square of the correlation between the predictions and the actual value and ranges from 0% to 100%
- Mean Absolute Error (MAE)
- Root Mean Absolute Error (RMAE)

Classification Models

- Confusion Matrix: Visual representation of performance of classification algorithms.
- Mathews Correlation Coefficient
- F1 Scores



		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Cross Validation

- Discussing why generalizability is a problem (Overfitting, changes in time, changes in sample characteristics)
- Evaluating generalizability of a model using K-Fold Cross Validation

K-Fold Cross Validation Steps-

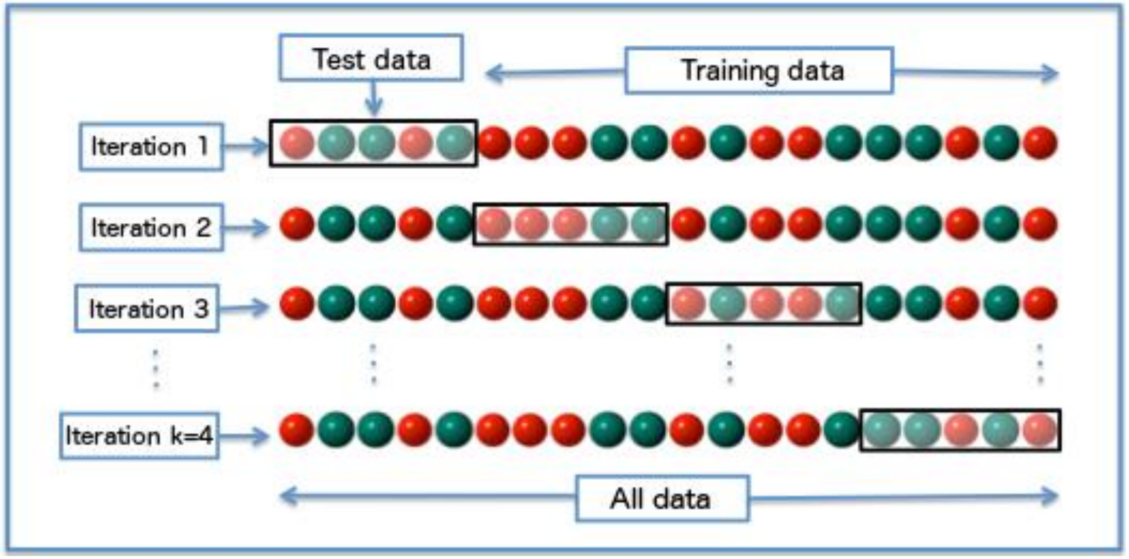
- ✓ We have a dataset with predictors and outcomes for N observations
- ✓ We choose a K value where K is the number of times we want to split the data and test our model
- ✓ Split the dataset so we have N/K cases left out
- ✓ Train the model on the remaining cases (N-N/K)
- ✓ Evaluate the training model by comparing predicted outcomes to the true outcomes
- ✓ Repeat the process for the remaining cases
- ✓ Combine (Average/Median) the evaluations to evaluate overall performance.

Cross Validation (contd.)

We generally use larger values for K since:

✓ We get more variability in the models performance estimate.

✓ Better representation of the models performance on new data



Limitations of K-Fold Cross Validation-

- Cannot address changes in relationships over time
- Cannot address unrepresentative samples
- Does not work when outcomes are skewed or imbalanced (Also introduce stratified K-Fold cross validation)

Feature Engineering

Methods

- Recategorize
- Infer feature from another variable
- Dimension Reduction
  - PCA
  - Cluster Analysis
- Apply Descriptive Statistic
  - Central tendency
  - Variability
  - Extremes
  - Trends

Common Mistakes and Limitations

- Using the outcome variable
- Not using descriptive statistics properly
- Too much overlap with other predictor variables
- Multicollinearity

Note- Cross validation is the best way to determine whether to keep or shelf a featured variable. If the cross validated error improves with the feature added, we should keep it.

K-Means Clustering

- K-means is a form of cluster analysis used to detect the sub-groups in a collection of objects/observations

- Shows which objects/ observations have similar patterns of characteristics/ responses

- K-means tries to divide *n* people into a set of *k* distinct clusters that are described by the *means* of the characteristics of people in those clusters

- Whereas PCA reduces the number of predictor/outcome variables into smaller numbers, K-means can reduce the number of people into smaller groups

K-Means assumes that clusters are centered at means for certain variables

- Each cluster is defined by a **centroid**
    - **Centroids** are a point representing a specific value on every characteristic
    - A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.
- **The objective of K-means is to find the position of *k* centroids**

Lloyd’s Algorithm

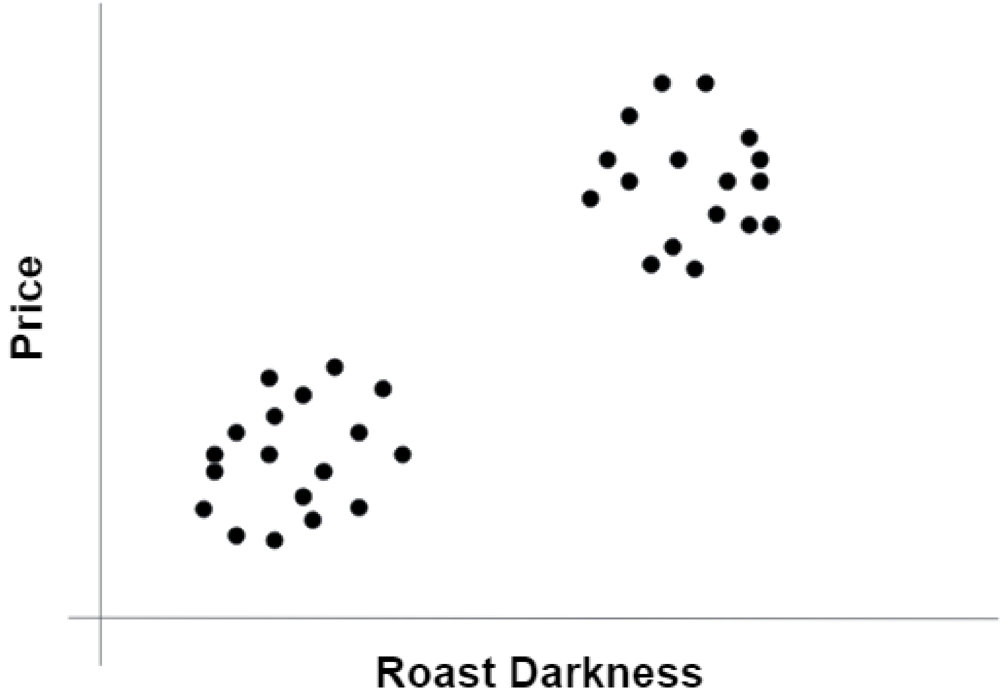
- **How to find the position of the centroids** ?

Start by picking a **random** position for each of the *k* centroids

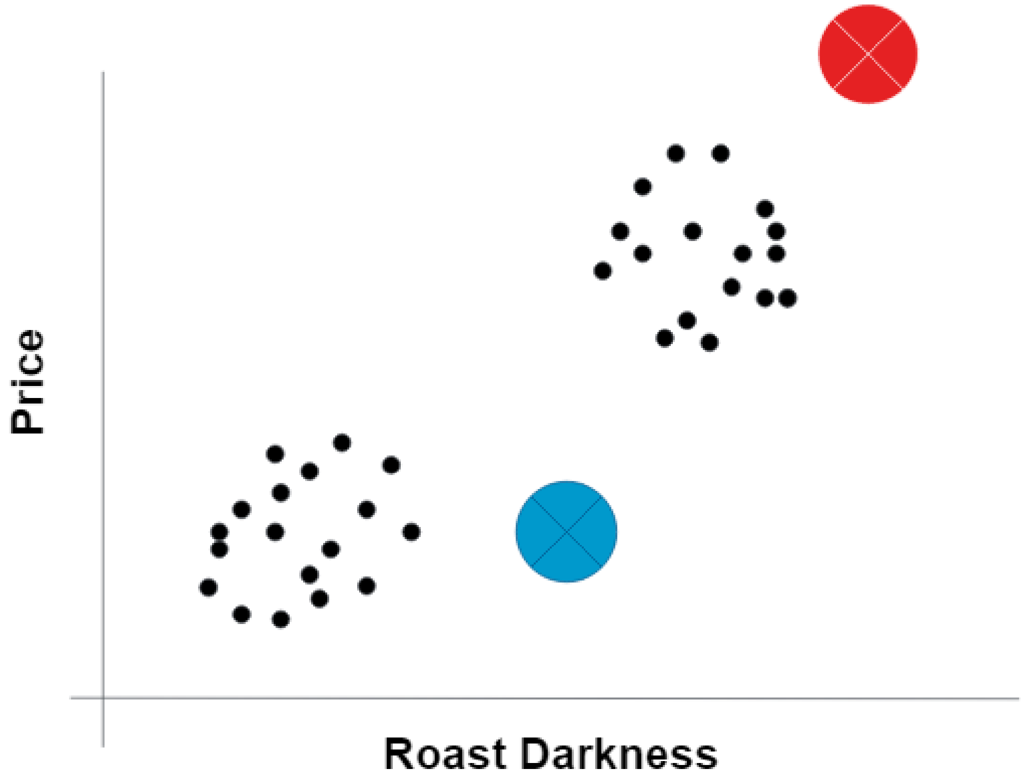
- **Assign** the points to the closest cluster (squared Euclidean Distance)
- **Move the centroids** to the middle of their assigned points

- **Repeat** assigning and moving **until cluster assignment doesn’t change**

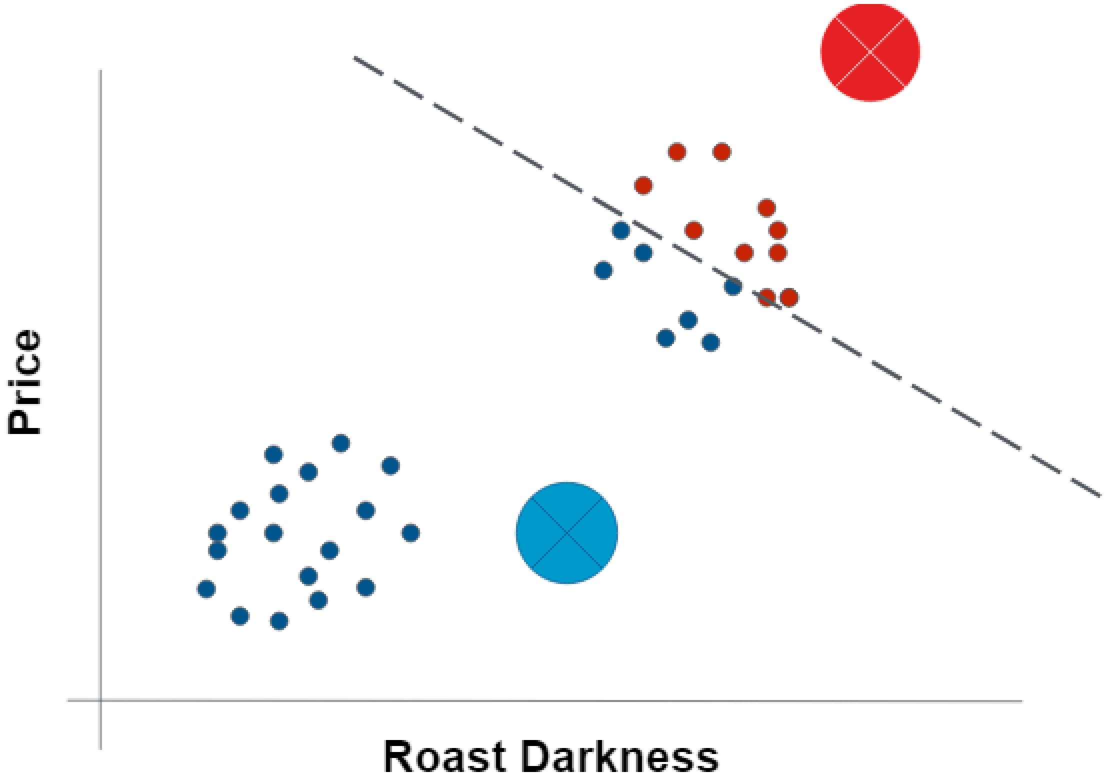
Collection of points:
Coffee Price vs Roast Darkness



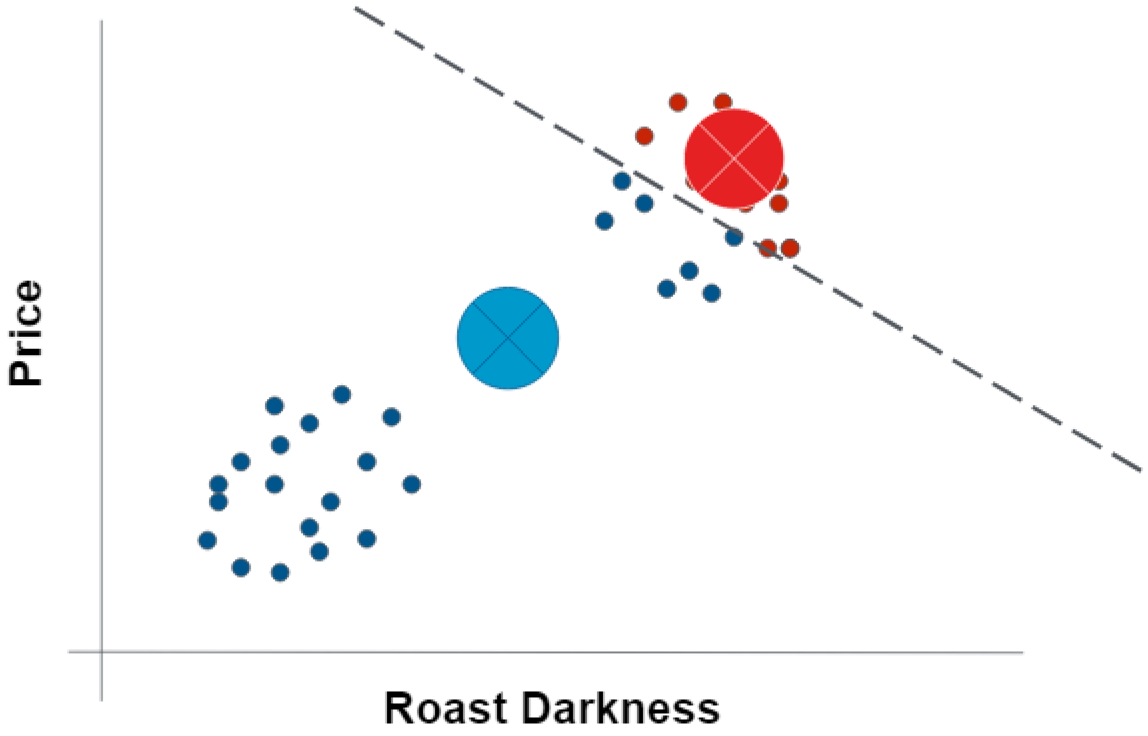
Step 2- Randomly place centroids



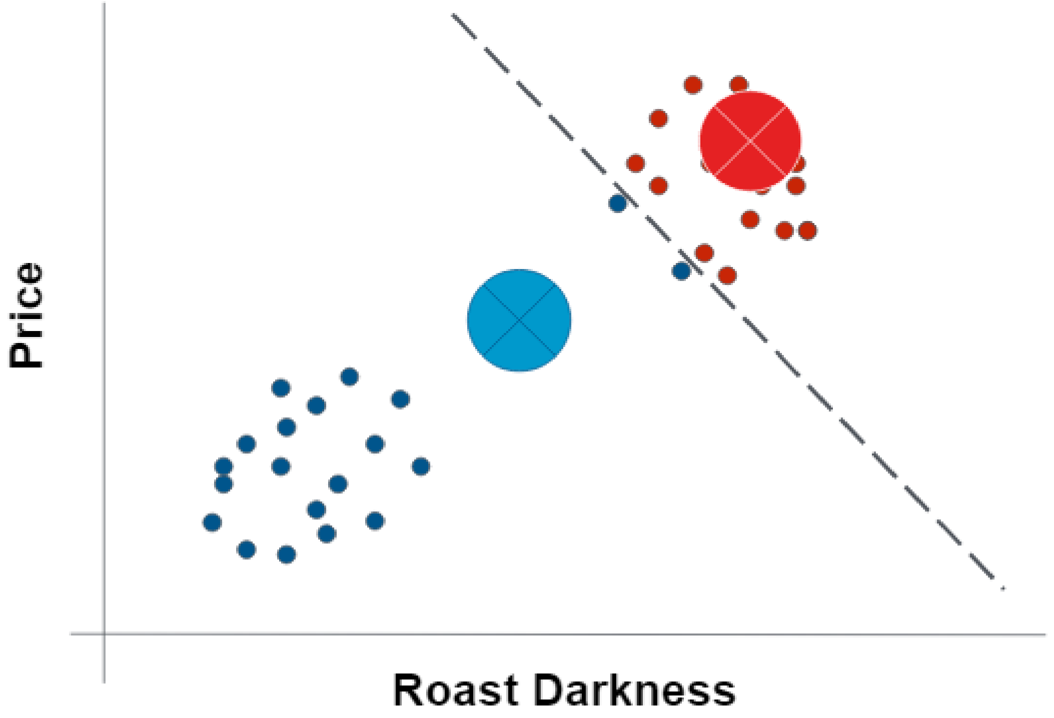
Step 3- Assign points to closest centroids



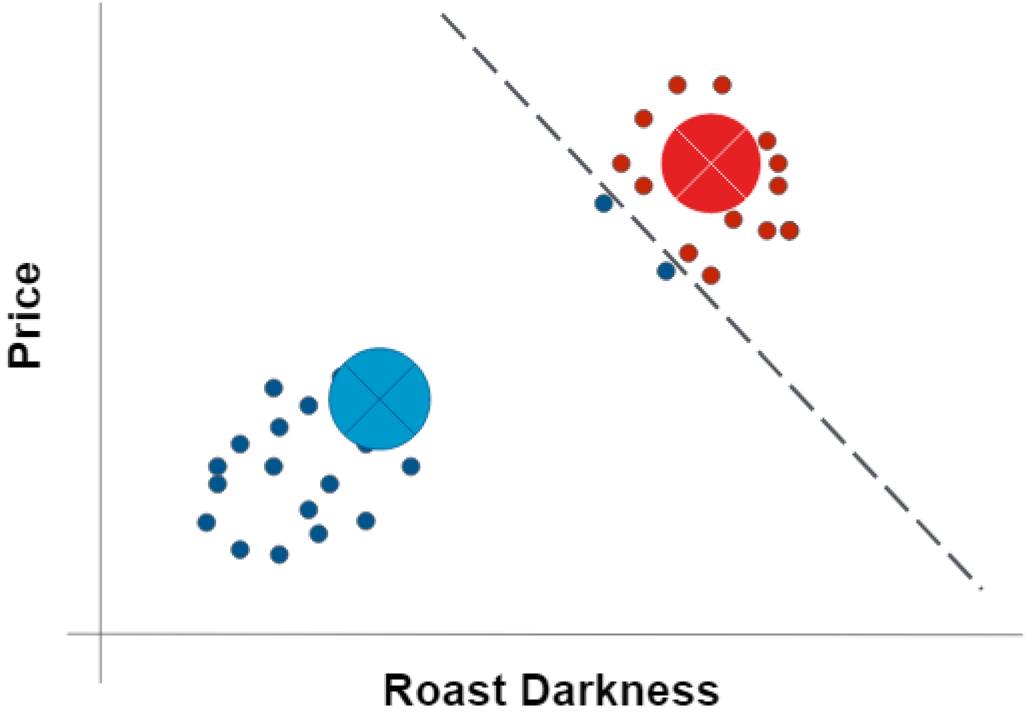
Step 4- Move centroids to the mean of assigned points



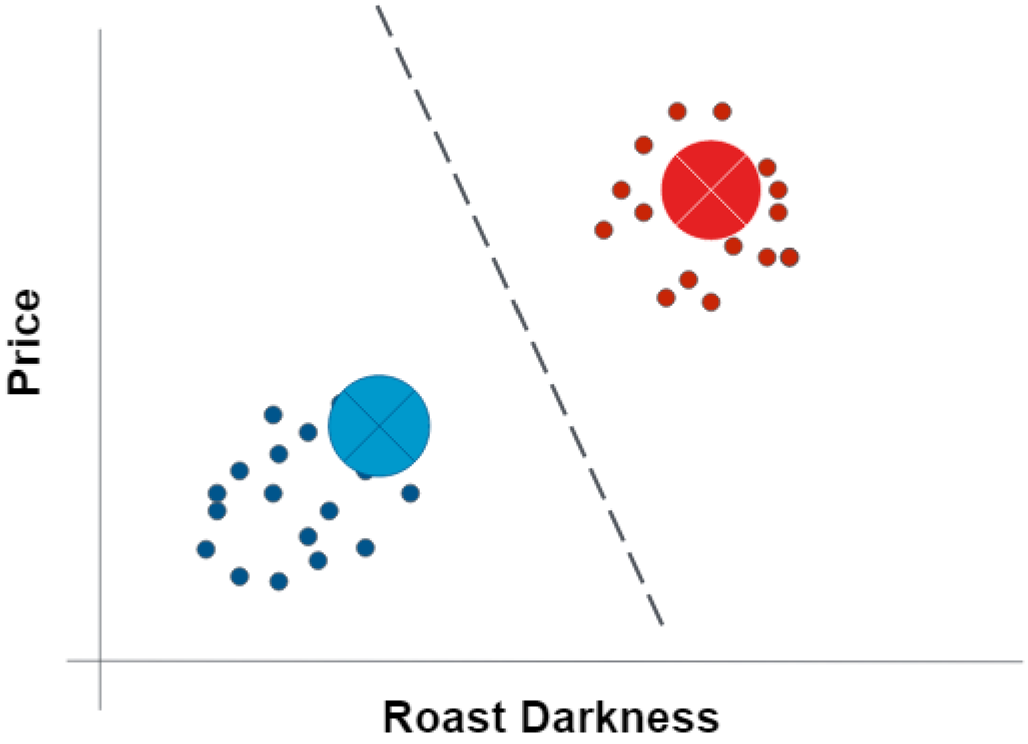
Step 5- Assign points to closest centroid



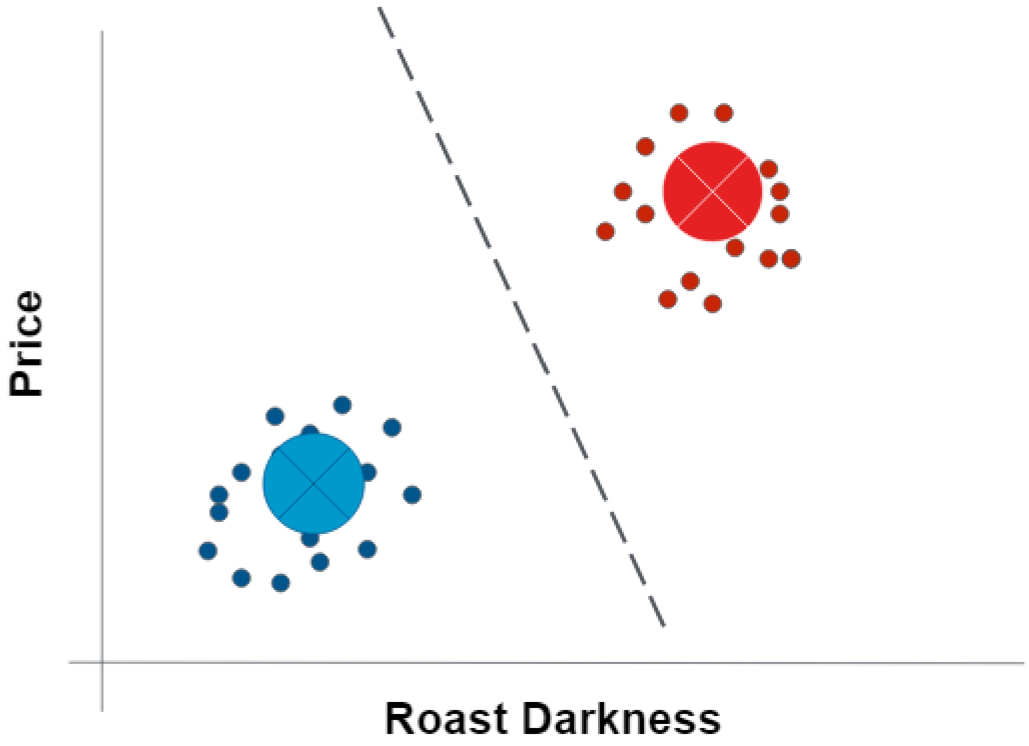
Step 6- Move centroids to the mean of assigned points



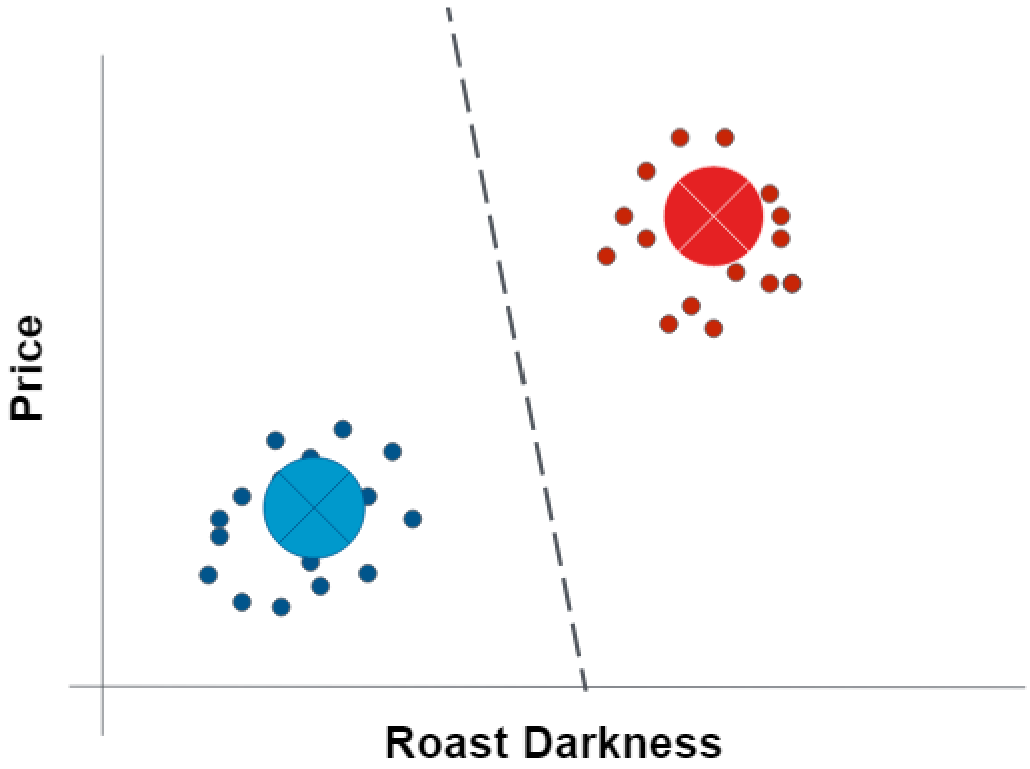
Step 7- Assign points to closest centroid



Step 8- Move centroids to the mean of assigned points



Step 9- Assign points to closest centroid



<u>NOTE</u> - STOP WHEN NO CHANGE OCCURS

Algorithm Steps

- **Load** rectangular formatted data that is standardized
- **Choose** *k*: number of clusters to fit to data
- **Run** K-means algorithm with random centroid starting positions
- **Examine and Interpret results**

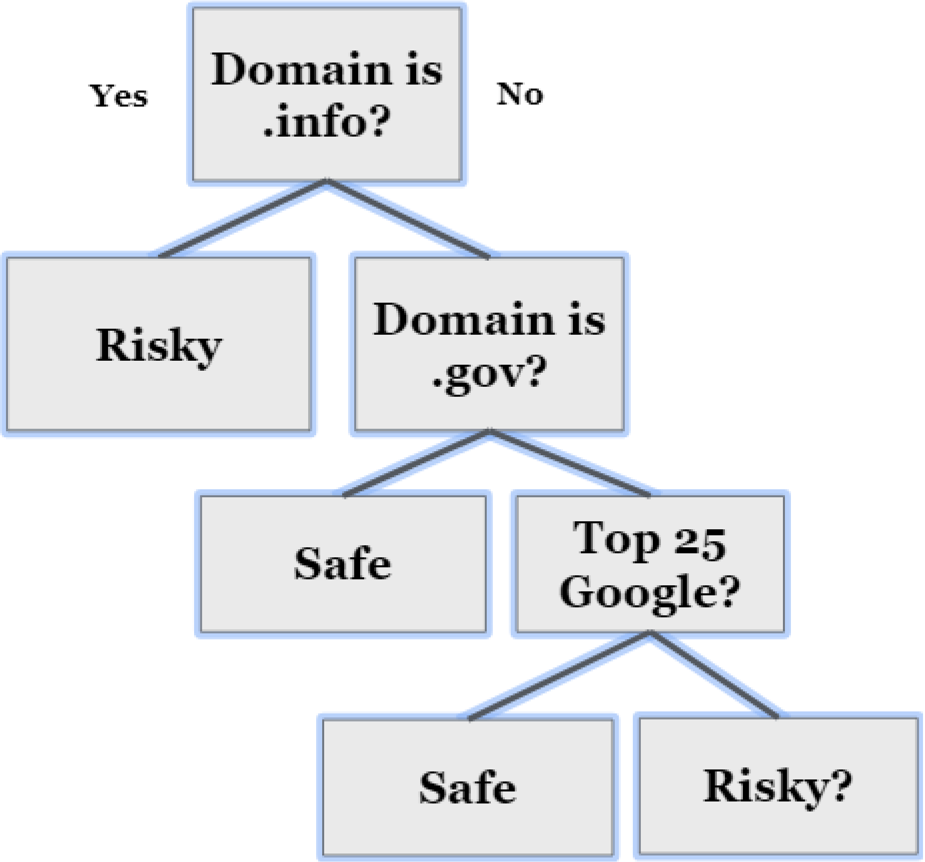
- Make sure to standardize the variables if they are very different scales
- Data must be ratio/interval scaled

Number of clusters = square-root(number of items / 2)

- If you are clustering 100 people: Expect roughly 7 clusters
- If you are clustering 50 people: Expect roughly 5 clusters

Decision Trees

- Decision trees are flexible and powerful method of finding how an outcome relates to the values of many predictors.
- Decision trees make use of many different variables to determine which group a case belong to based on the values of its variables
- Goal- To ask a series of questions that separate our
- **Example**: Classifying whether a website is "safe" or "risky"
  - Can know if a website is risky based on its domain and Google rank
  - If ".info" domain, risky



Decision Trees(contd.)

Algorithm Steps-

- ✓ Start with a dataset which includes outcome categories and variables associated with it.
- ✓ Split on every possible value of the outcome variable
- ✓ Split points into two groups based on whether they are above or below that value
- ✓ Examine the **purity** (separation between groups) from splitting at that point
- ✓ Choose the variable and value for that variable that gives us the best change in purity measure
- ✓ Repeat splitting process and keep explaining the exceptions till we reach a stopping point.

Benefits-

- ✓ Relationship between outcome and predictors are clear
- ✓ Easy to understand what variables are important in making predictions
- ✓ Can examine non linear relationships between predictors and outcomes
- ✓ Quick to calculate
- ✓ Works for both classifications and regression problems.

Limitations-

- ✓ Complex to find the best tree since we have many hyperparameters to search through
- ✓ Tree models can be prone to overfitting