Building Classification models with scikit-learn

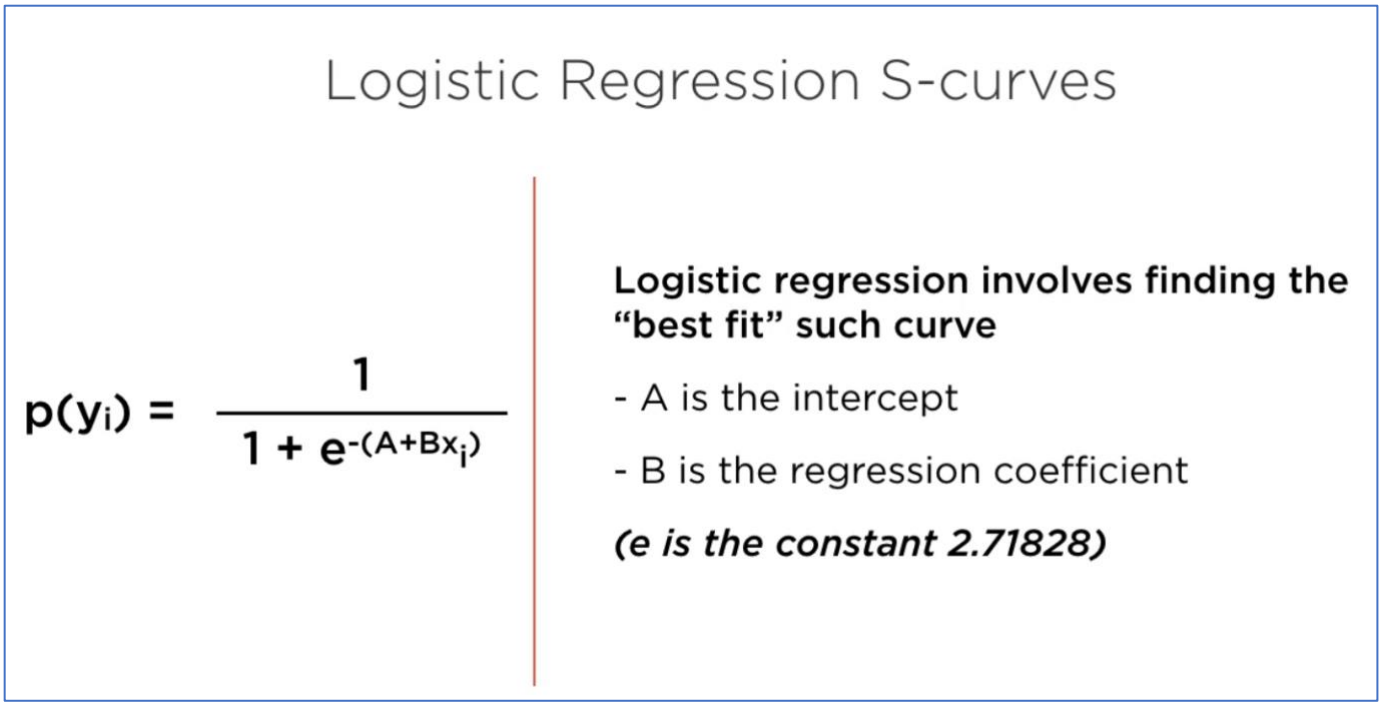
- By Janani Ravi

Classification or Classifier:

- Building a ML classification model using scikit-learn
  - Binary classifier: objective is to build a machine learning based classifier model to differentiate between cat or dog.
  - Training: ML's classifier needs to be trained, we need to feed in a large corpus of data that has been correctly classified and this is the corpus that the classification model will use to learn from data.
  - Classification model: The data is fed into a classification algorithm to get a classification model or a classifier. It can be one of many algorithms like, naive based, support vector machines, decision trees.
    - we need feedback into the model to improve its parameters, this feedback is the loss function or the cost function.
    - The idea behind training the model is to minimize the loss function and this minimization of the loss function improves our model parameters and makes it a more robust model for prediction.
  - Prediction: Once we have trained model, we'll use it to predict new instances that the model has never encountered before.

Logistic Regression for classification:

- We have training data, and we need to set up S curve on this data.
- The fundamental intuition underlying logistic regression helps find how probabilities are changed by actions.
- We have a bunch of training data points, logistic regression tries to fit an S curve on the data where the equation of the S curve is given below, when we train our logistic regression, we try to find is the best fit S curve through all the points.
- The whole objective of training your logistic regression classification model is to find this best fit S curve which involves finding the best values of A&B in this formula.



Cross Entropy : Loss function

- We need to minimize the loss function while training the model.
- Minimizing this loss of the model improves the model parameters and gives us a robust model.
- Linear regression -> the cost function -> mean square error
- Logistic regression classifier -> cost function -> cross entropy
- The output of the classification model is a probability score and cross entropy measures how well the estimated probabilities of the model match the actual labels to which the data points belong.
- Minimizes the cross entropy of actual labels versus the labels predicted by our model for better fir of S curve.

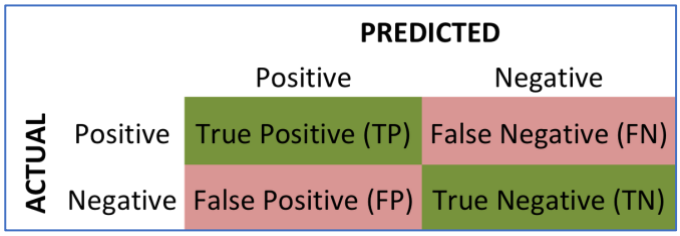
Accuracy, Precision, Recall

Accuracy:

- Compare the predicted labels versus the actual labels, higher the number of matches, higher the accuracy.
- But accuracy is not the correct metrics if the value is high, chances are it is predicting wrong even for the negative results. It might have showed no cancer for an actual cancer patient since the % of such patient is very less.
- This happens when:
    - Certain labels in the data set are common or rarer than others, such a data set is set to be skewed.
    - For example, in our medical records data set only maybe one in every hundred individuals has this rare type of cancer when you're working with skewed data accuracy tends to be a poor evaluation metric that's when you need to consider other metrics and for that we set up a confusion matrix.

Confusion Matrix:

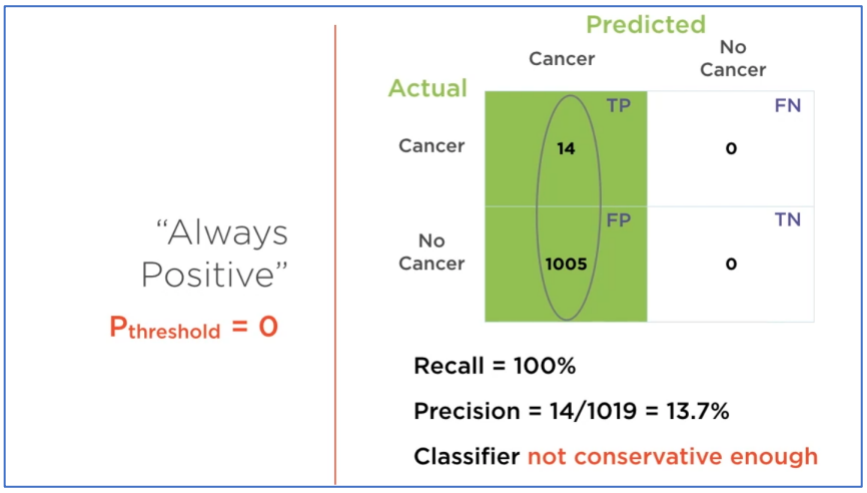
- A confusion matrix is simply a grid tabulating the predicted labels from our model versus the actual labels.
- Calculating accuracy, precision and recall using Confusion matrix:

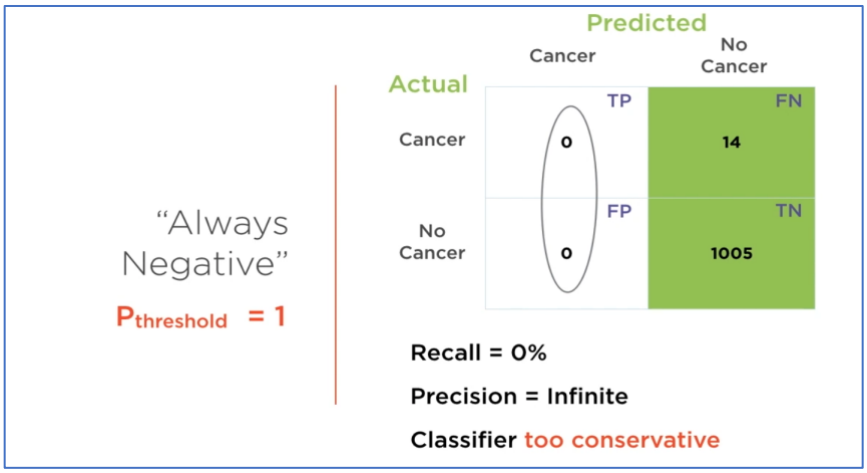


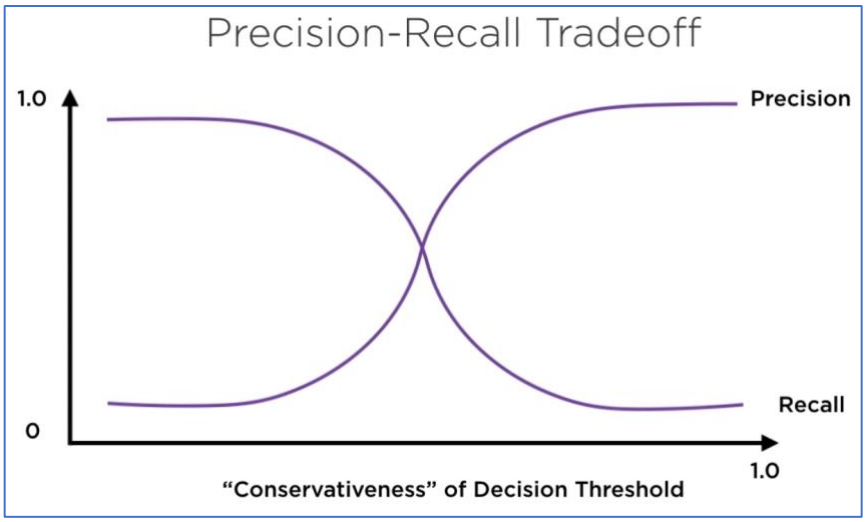
- $$\text{Accuracy} = \frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$$
- Accuracy takes a note of all the cells where the actual label is equal to the predicted labels.
- $$\text{Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}$$
  Precision can be thought of as the accuracy of your model when it flags cancer on prediction. So, we consider that column in prediction.
- $$\text{Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}}$$
  Recall considers the two cells on the top, recall is the accuracy of your model when cancer is present, that Is row with cancer by actual dataset.

Evaluating classifier:

we can apply these metrics to evaluate our classifier to find the best model.
In logistic regression classification, it is necessary to choose the right threshold to classify the data correct.



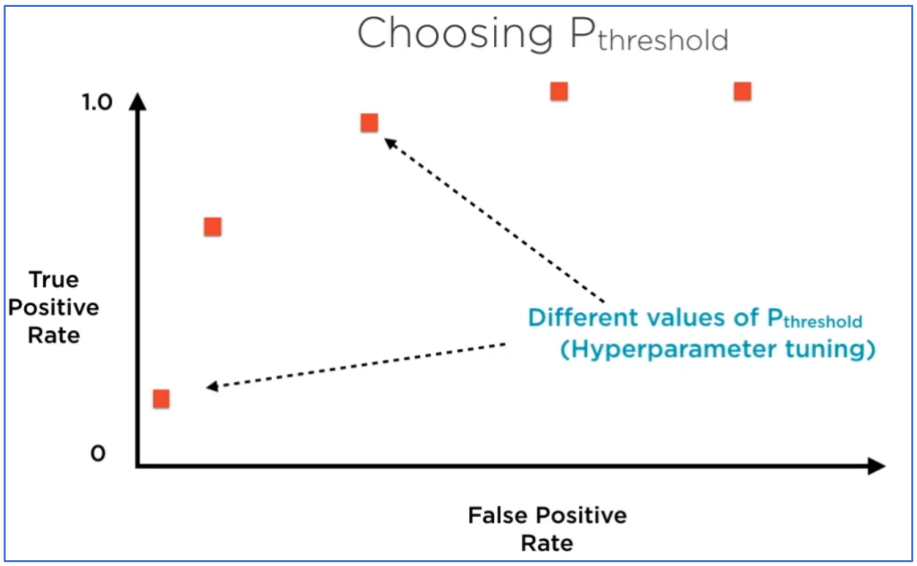


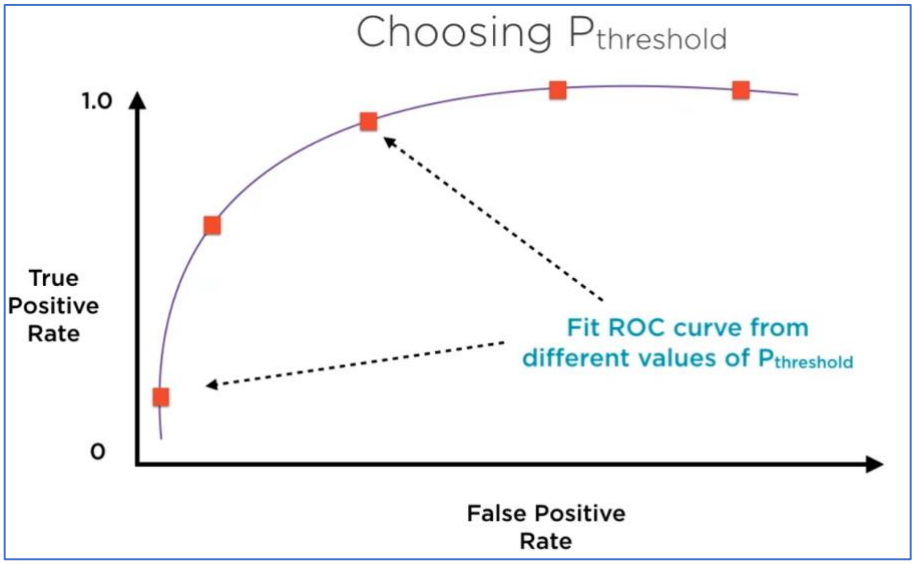


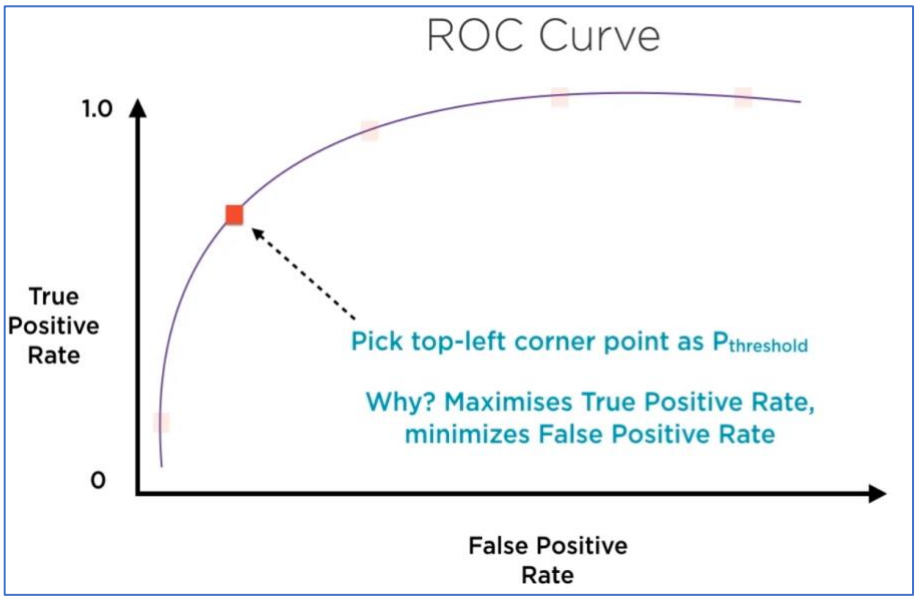
ROC Receiver operating characteristic - curve

For a good classification model, we want:

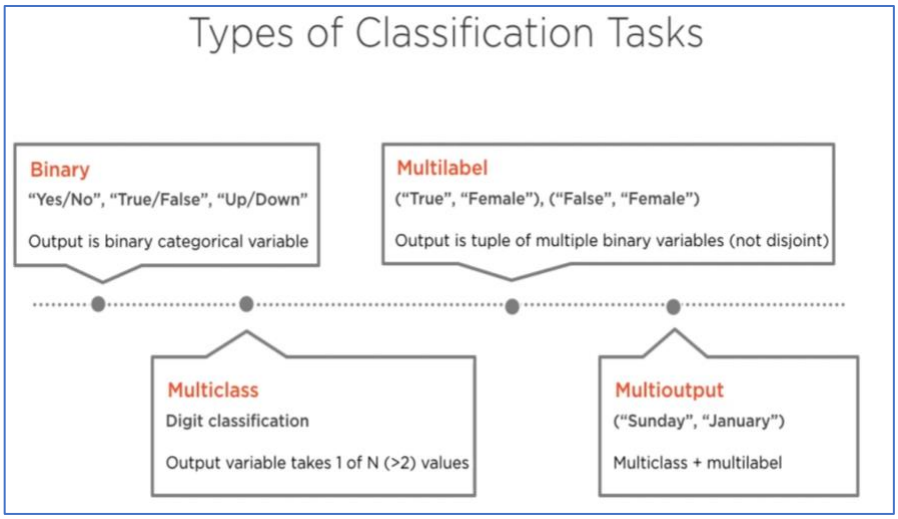
- True Positive to be as high as possible.
- False positive to be as low as possible.


Choosing $P_{threshold}$
1.0
True Positive Rate
0
Different values of $P_{threshold}$ (Hyperparameter tuning)
False Positive Rate


Choosing $P_{threshold}$
1.0
True Positive Rate
0
Fit ROC curve from different values of $P_{threshold}$
False Positive Rate


ROC Curve
1.0
True Positive Rate
0
Pick top-left corner point as $P_{threshold}$
Why? Maximises True Positive Rate, minimizes False Positive Rate
False Positive Rate

Types of Classification:



Different kinds of classifier models are optimized for some algorithms are inherently,
multi label in nature such as the naive based algorithm
binary classification, logistic regression, and support vector machines
inherently binary classifiers can be generalized for multi label classification.
two specific techniques to generalize binary classification models to multiclass classification :

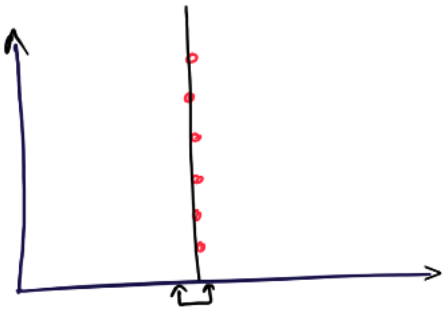
1. one vs. all technique
2. one vs. one

Linear Discriminant Analysis and Quadratic Discriminant Analysis classifier:

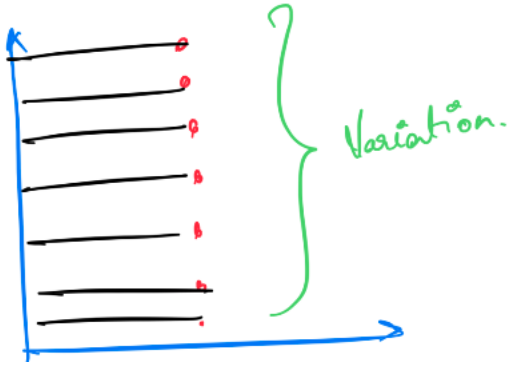
The intuition behind an important dimensionality reduction technique called principal components given the set of data points our objective is to find the best directions to represent the data.

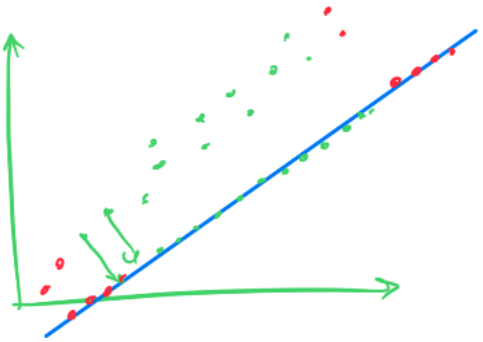
PCA: Principal component analysis

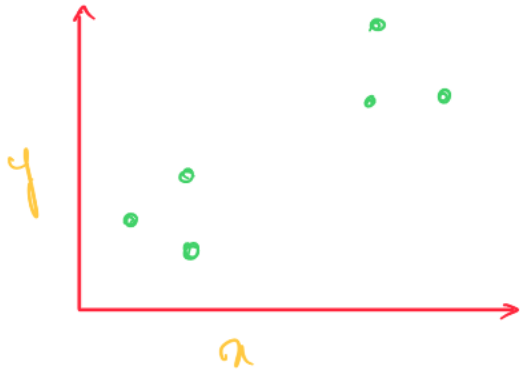
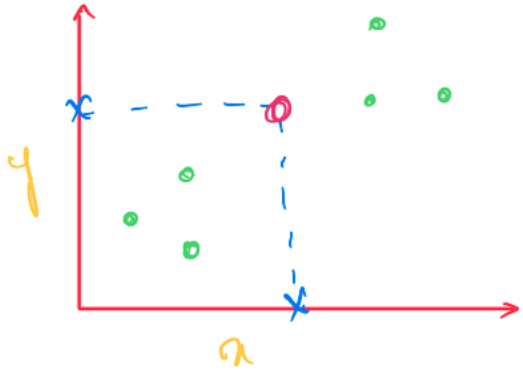
- It is a popular technique for analyzing large datasets containing a high number of dimensions/features er observation, increasing the interpretability of data while preserving the maximum amount of information, and enabling the visualization of multidimensional data.
- The process of extracting the most important information.
- Leads to compression.
- It's a trade of between faster computation and less memory consumption versus information loss.

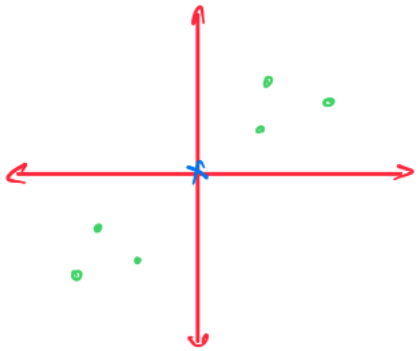


- in the above graph we can see the data is represented in 2D, 2 variables/2 features are used. We can try to reduce the dimension without losing the information.
    - Doing the above all the values are at same point on X axis, there is no variability.

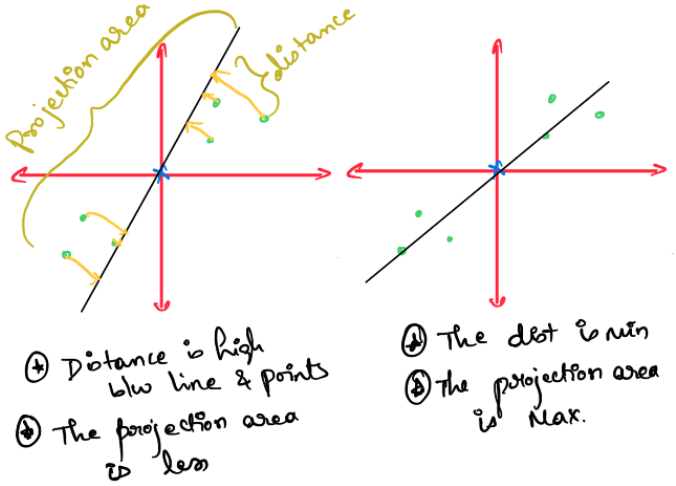
- 

  - This increases the variation, and all the points are available on Y axis.
  - Another example,
  - 

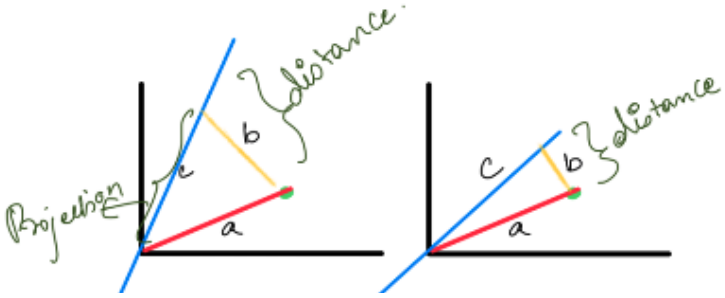
- Step-by-step PCA
  - Let's say the data has 2 features and we plot them on the graph.
  - 
  - Now take the average of each feature and plot it. From the average of both the features we get the center of the data.
  - 
  - Shift the data such that the center is on the origin (0,0) on the graph.



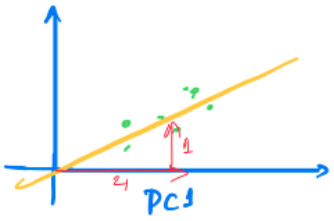
- Fit a line to the data that goes through the origin.
  - The should either minimize the gap between line and data or
  - The line maximizes the area of projection on the line.



  - Mathematically,
    - The distance between the points and origin remains same no matter which line we choose.
    - Since they make right angle, If we apply Pythagorean theorem to the above line, the distance between points and line has relation as,

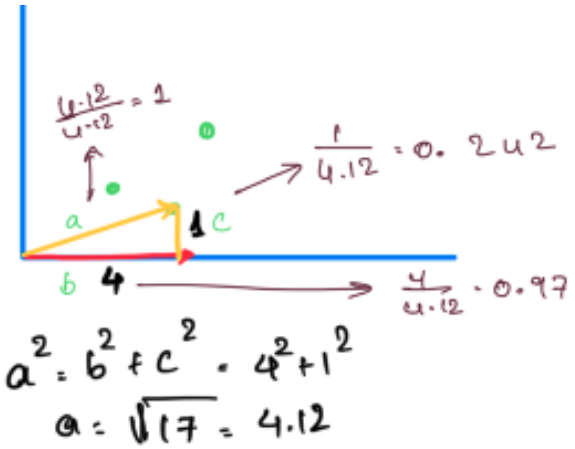


    - In the above diagram, we can see that as the distance of b increases, the length of c decreases and vice versa.
    - Of the two, PCA finds the line that max the projection area that is C.
    - To get the line, draw a random line that passes through origin and potentially through the points, calculate the distance between the points and the origin. (d1, d2, d3, ….)
    - Find the square of these distances to eliminate the negative values and sum up all the squared distances. = sum of squared distances(SS) distances.
    - Update the line, calculate the SS distances like mentioned in above steps. Repeat until get the line with largest SSD between points and origin.
    - This line is called PC1.

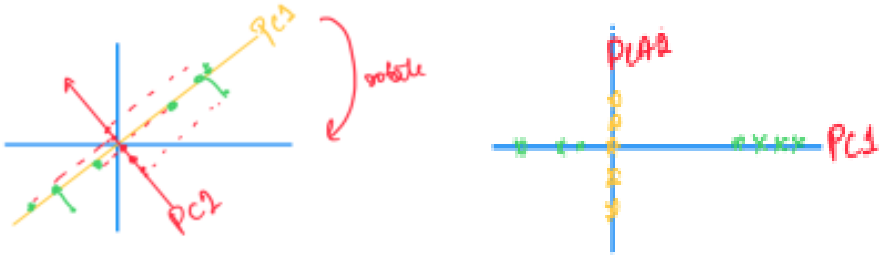


* If Slope is 0.25
* for every 4 Step along x we take 1 Step along y.
* i.e., the data Step along x.
* for every 4 data points of x feature, 1 y feature.

- It is called Linear Combination of variables.
- Scaling the PC values:



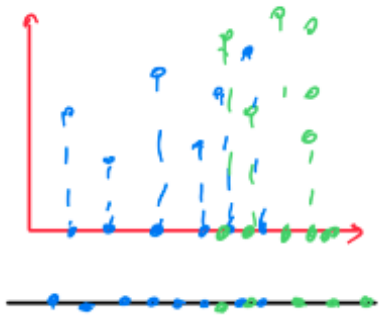
  - We have the unit vector.
  - This unit vector with 0.97 and 0.242 is called Singular Vector or Eigen vector.
  - The SSD for the best fit line is called Eigen values for PC1. SSD/(n-1)
  - Sqrt(SS(for PC1)) is singular value for PC1.
- PC2
  - It is simply the line through the origin that is perpendicular to PC1, without any further optimization that must be done.
  - i.e., 0.242, 0.97 for PC1, then -0.242, 0.97 for PC2.



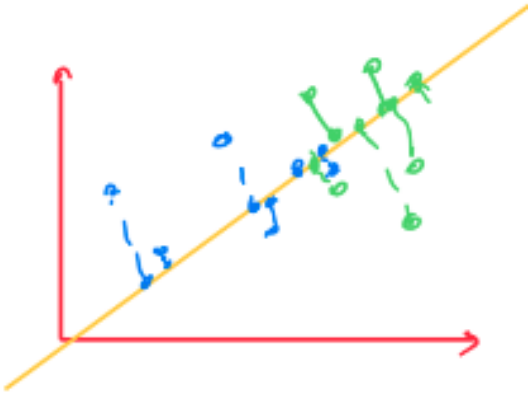
- Eigen values are the measure of variation.
- If variation, PC1 = 15, PC2 = 3, the total variation is 15+3 = 18.
- That is PC1 = 15/18 = -.83 or 83 % of total variation around PC1
- And 3/18 = 17% of variation around PC2.
- The number of PCs is either no of variables or the number of samples, whichever is smaller.

LDA: Linear Discriminant Analysis

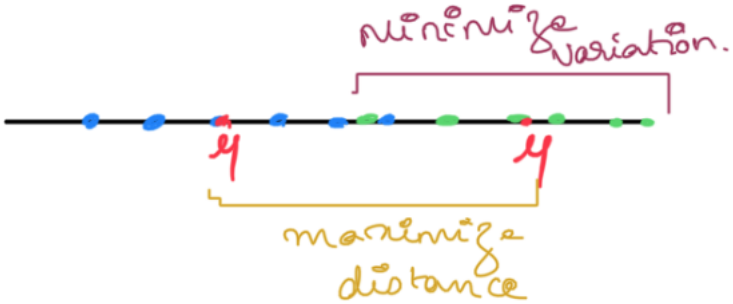
- It is like PCA, but it focuses on maximizing the separability among the known categories.
- It also reduces dimension.



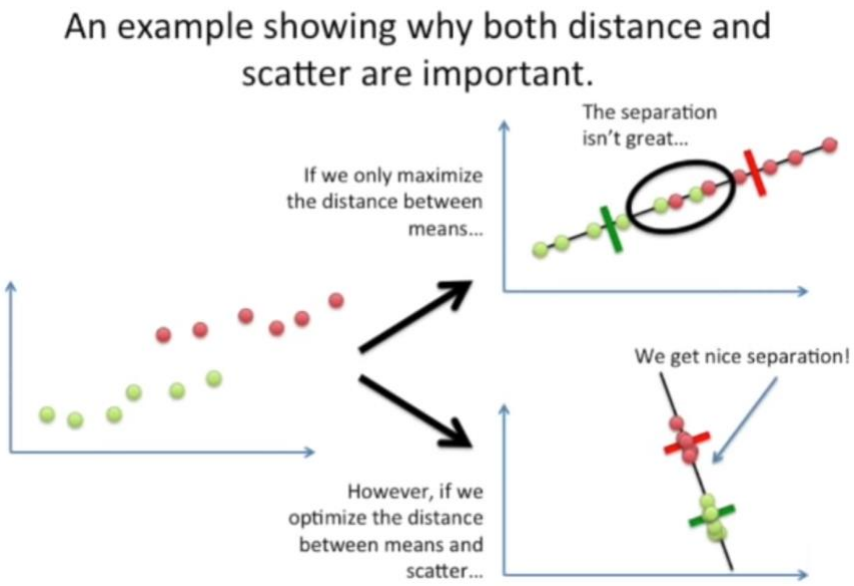
- In the above example we are trying to reduce the 2D to 1D, but we can see that the data are not completely separated, they are overlapping.
- LDA tries two information from both the categories and create a new axis, project the data onto the new axis in a way to maximize the separation of two categories.



- How create new axis?
    - There are two criterions.
        - 1. After projecting the data onto new axis, we need to maximize the distance between means of two data.
        - 2. Minimize the variation within each category.

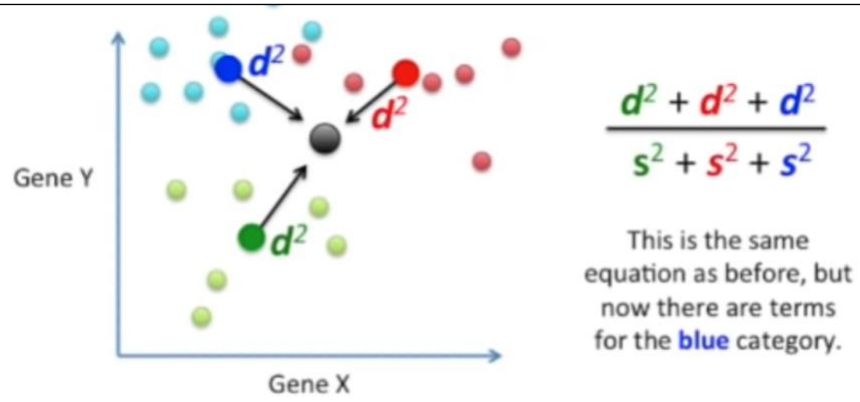


- $(\text{mean } 1 - \text{mean } 2)^2 / S_1^2 + S_2^2$.

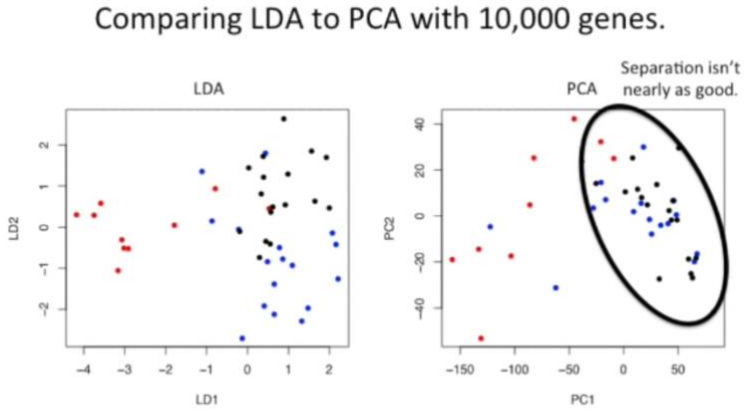
- 

LDA for 3 categories

1. Find the center of all data.
2. Find the center of each data.
3. Calculate the distance between center of categories and center of graph(all data)
4. Apply calculations as shown below.

- 

- LDA creates 2 axes here.
- LDA to PCA comparison:

- 

Similarities between PCA and LDA

- Both rank the new axes in order of importance.
  - PC1 (the first new axis that PCA creates) accounts for the most variation in the data.
    - PC2 (the second new axis) does the second best job...
  - LD1 (the first new axis that LDA creates) accounts for the most variation between the categories.
    - LD2 (the second new axis) does the second best job...
- Both can let you dig in and see which genes are driving the new axes.

QDA: Quadratic Discriminant Analysis (QDA) is a generative model. QDA assumes that each class follow a Gaussian distribution.

- The class-specific prior is simply the proportion of data points that belong to the class.
- The class-specific mean vector is the average of the input variables that belong to the class.

Stochastic Gradient Descent

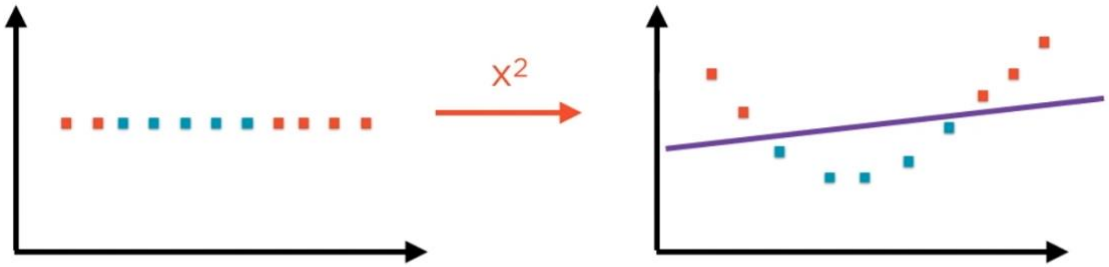
Gradient descent process starts off at some point somewhere with some initial values of W(weights) and B(biases) and walks down the slope to find the best possible values of W and B thus this process of gradient descent. It iteratively converges to the best classification model.

Support Vector Machine

- SVM classifiers finds the hyperplane that best separates points in a hypercube.
- If the data is linearly separable and if it's possible for to draw hard decision boundary between the data points observe the points of each category that are the closest to the decision boundary.
- The nearest instances on either side of the boundary are called support vectors.
- These are the points that give this classification algorithm its name support vector machines.
- When we train our support vector machines algorithm it tries to fit the widest possible margin between the nearest points on either side.
- Hard margin classifiers are sensitive to outliers.
- Hard margin classifiers require perfectly linear separability and data and do not tolerate outliers.
- Soft margin classifiers allow some violations of the decision boundary.

Non – separable data

- For non-separable data we can apply smart transformations also known as the kernel trick.
- To make data separable SVM classification can be extended to almost any data using this kernel trick.
- let's apply the X square (X^2) transformation to this data.



Original Data

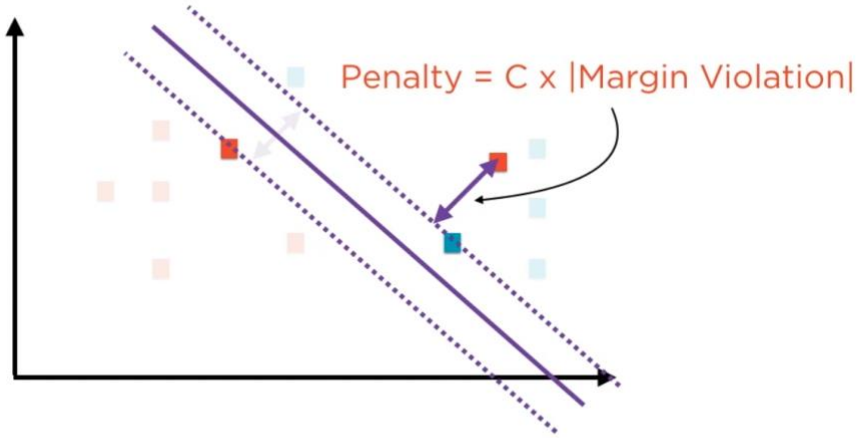
Not linearly separable

Square of original data

Now linearly separable!

- Simple classification problem classifier : review as positive or negative.
  - Decision boundary = W 1X1 + W 2 X 2 + b = 0
  - SVM algorithm will determine the values of W1, W2 and those are the model parameters so for any review we'll have values for X1 and X2 now the decision plane will separate points based on whether W 1X1 , W 2X2, B is =, <, > 0
  - Positive reviews might lie on the side of the plane where this formula is greater than 0.
  - Negative reviews might lie on the side of the plane where this formula is less than 0.
  - SVM is to find values of W 1, W 2 and B which satisfy this formula and separates your data neatly.
  - To avoid or minimize outliers by penalizing during the process of optimization,
    - Calculate the magnitude of the margin violation, how far is the outlier point from your decision boundary.
    - Calculate for each point which is on the wrong side of the boundary.
    - Multiply every point which has this margin violation by a penalty factor C.
    - If the value of C is high, it is hard margin classification.
    - If the value of C is small, it is soft margin classification.

Classification Using the Decision Boundary



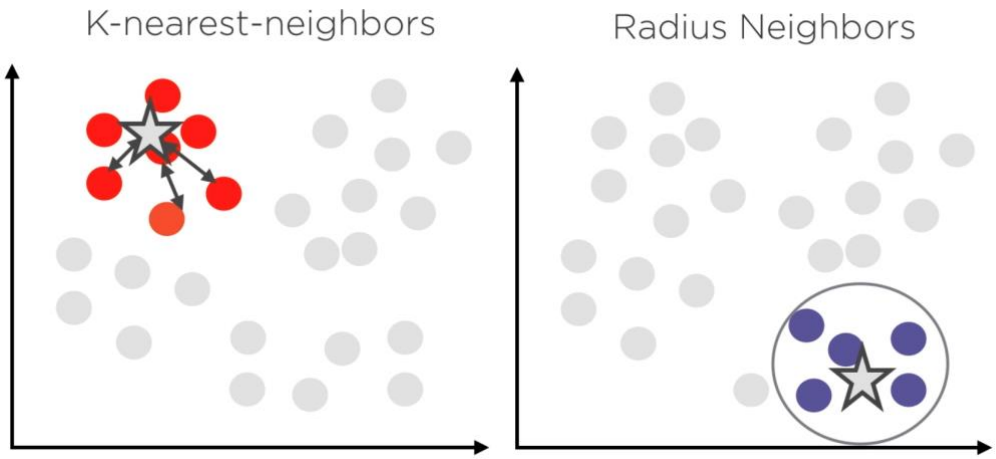
Very large values of C ~ hard margin classification

Very small values of C ~ soft margin classification

  - SVM goal:
    - Maximize the boundary street.
    - Minimize the margin violation.

Nearest Neighbor model

- NN classifier works based on the similarity measures.
- It classifies the new data based on the entire training data.
- It finds the sample that is like the new data, and it sets the label associated with the sample.
- We use distance measures to calculate the similarities.
- There are different distance measures that can be used to figure out the nearest neighbors:
    - Manhattan distance
    - Euclidean distance
    - Hamming distance
- Two categories of NN classifier:
    - K nearest neighbors : consider K nearest neighbors and predict the label based on commonality.
        - Perform voting amongst these K- NN and take the category which has the highest board.
        - K is hyperparameter.
    - Radius neighbors' classification: where you consider all the data points within a certain radius as neighbors to determine the predicted category of an incoming sample.
        - you'll perform voting amongst all neighbors who are within that radius.
        - Radius is a hyperparameter.



Decision tree for classifier

- Decision Trees are a non-parametric supervised learning method used for classification and regression.
- Decision trees set up a tree structure on the training data and this tree structure helps make decisions based on rules.
- DT gained the knowledge from the training data and make it to set of rules.
- The decisions are based on thresholds.
- The thresholds for these decisions is determined in the training process of the decision tree model.
- To summarize, the objective of the training process of the decision tree classifier is to fit the knowledge that it has gained from the training data into rules and every rule involves a threshold.
- The order of the decision variables also matter.
- Decision trees are most popularly called "CART" - Classification And Regression Tree.

Naïve Bayes

- Follows Bayes theorem of conditional probabilities for classification.
- A priori probabilities : The upfront information, without details.
    - P(event)/total
- A Conditional probability: Probability of occurrence of certain event upon gaining some additional info.
    - P(A|B)
- Once we have prior probability and conditional probability, we apply bayes theorem.
- Calculate the probability for the events, pick the label that has highest probability.
- If P(A) > P(B), label it A.
- It's an extreme powerful algo, that can give us robust results.

Assumptions:

- It's called naïve since it assumes that the features are independent of each other.

Hyperparameter tuning with scikit-learn.

- Grid search: for hyperparameter tuning it uses grid search to find the best candidate model .
- It uses a technique called cross validation to find the best model on the training data.
- If there are number of different values for the different hyperparameters for the estimator object, then GS performs an exhaustive search with different parameter combinations to find the best possible model.
- Grid Search is very straightforward, easy to use with scikit learn.
- Drawbacks:
    - Grid search is computationally very expensive.
    - The cost and complexity of grid search can grow very quickly.
    - Training on a cloud platform can quickly become very expensive.
    - Grid search does not differentiate between important and trivial hyperparameters, it reads all hyperparameters the same way.
- An alternative to Grid search for hyperparameter tuning is Random search of the hyperparameter space.

Image Classification

- Color images – Multi channel image
    - It is represented as (R, G, B) – (256, 256, 256)
    - R = (255, 0 , 0)
    - G = (0, 255, 0)
    - B = (0, 0, 255)
    - The image grid can be represented as, (6, 6, 3)
        - The 3 in tuple says, the image is multi-channel(color)
        - 6, 6 is height and width of the image or grid size.
- Grayscale image – single channel image
    - It is represented as the intensity of the pixel.
    - The value ranges between 0.0 – 1.0
    - Image grid represented as (6, 6, 1)
        - 1 to say it's a single channel image.
        - 6,6 is the grid size.
- When there or multiple images to work on in list, it is represented as a 4D.
    - (10, 6, 6, 3)
        - 10, no of images
        - 6,6 – height and width of image
        - 3, multi-channel image

Reference:

1. Pluralsight
2. https://devopedia.org/principal-component-analysis
3. PCA: https://www.youtube.com/watch?v=FgakZw6K1QQ
4. LDA: https://www.youtube.com/watch?v=azXCzI57Yfc