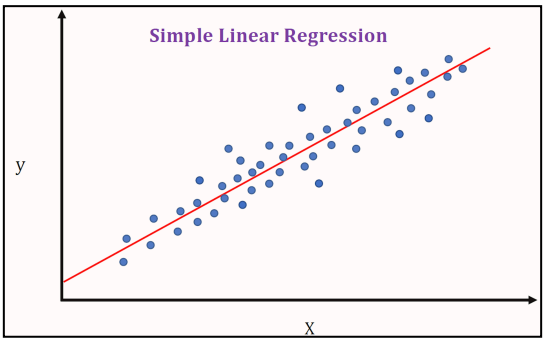
SUPERVISED MACHINE LEARNING

LINEAR REGRESSION

1. Linear regression is a parametric machine learning model used for regression problems.
2. It aims to find a line that best fits the data with least amount of loss function and gives a real number as an output.

— Simple Linear Regression

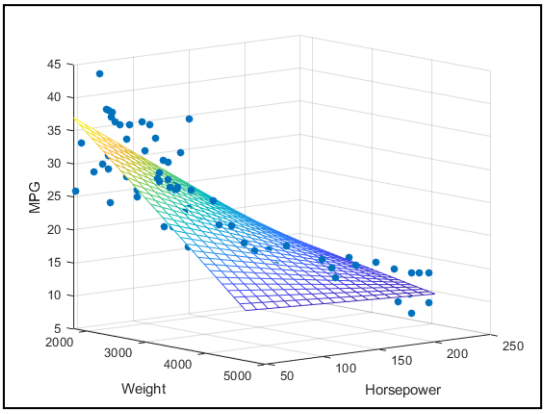


1. The output is based on a single feature.
2. The task is to predict output Y based on single input X

$$Y = B0 + B1.X$$

3. Where B0 and B1 are model parameters, B0 is intercept and B1 is slope or model coefficient.
4. This model assumes a linear relationship between X and Y and finds a best fit line by iteratively reducing a loss function to a minimum.

— Multiple Linear Regression



1. The output is based on 2 or more features.
2. The task is to predict output Y based on more than one input [X0, X1, X2, … Xn]

$$Y = B0 + B1.X1 + B2.X2 + \ldots + Bn.Xn$$

3. Where B0 is an intercept and B1, B2 … Bn are model coefficients.
4. Works same as simple linear regression but in multiple dimensions.

— Ordinary Least Square Method

1. OLS is a method used for linear regression. It is called so because we iteratively reduce a loss function called Residual Sum of Squares.
2. RSS is equal to sum of squares of difference between predicted and actual values. The differences are squared because some differences are positive and some are negative which cancel each other out.

$$RSS = \sum (Y - Yi)^2$$

3. First iteration takes mean as the reference to predict values.
4. OLS method is ideal for data having a linear relationship between predictor and output.
5. When number of observations is far more than the number of features and relationship is linear, OLS should be used straight away.

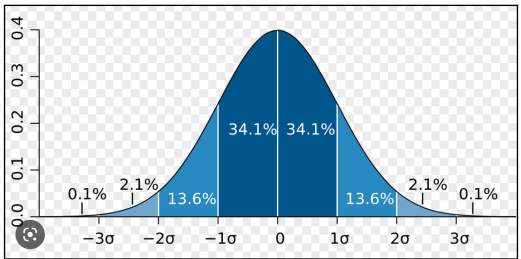
EVALUATION

1. Standard Error

1. SE tells us about how much a sample mean can vary if we use another sample of data.
2. SE is equal to standard deviation of means of multiple sample datasets.
3. Population model parameters are also calculated from sample model parameters using SE.

$$[\ \bar{B} - 2SE(\bar{B}) \ ,\ \bar{B} + 2SE(\bar{B})\ ]$$

4. The is a 95% chance that the value of population parameters lie within this range which comes from the fact that 95% of values are within 2 standard deviations from the mean.
5. Calculation —
   a. Assume we create 5 sample sets from a single data set.
   b. Find mean and standard deviation for all the samples sets.
   c. All of them will have different mean and standard deviation.
   d. We find standard deviation of the means.



e. This standard deviation of the means is called the standard error.

2. T-Value

1. T-value is statistical measure calculated to test a null hypothesis where the null hypothesis states an assumption that there is no relationship between a feature and the output.
2. Null hypothesis
    1. Ho : B = 0, no relationship between feature and output.
    2. Ha : B != 0, there is a relationship.
3. For a linear regression model to be good, the output and features should have a linear relationship with each other.
4. T-value is a statistical measure of how linear that relationship is

$$\text{T-value} = \text{estimated coefficient/standard error}$$
$$= B - B(0) / SE(B)$$

where B is the coefficient and B(0) is the hypothesised horizontal slope equal to 0

5. Here B - B(0) is the change in slope to estimate the linearity of a relationship between a feature and the output. SE(B) as a denominator standardises the T-value.
6. But T-test alone is not enough to reject the null hypothesis.

3. P-Value

1. P-value is the probability of validation weather T-test is statistically significant or not.
2. It is the probability of confidence that null hypothesis is true and there is no significant relationship between output and feature.
3. Even if we reject the null hypothesis based on T-value, P-value of 0.05 would imply that there is still a 5% chance that null hypothesis is True.
4. P-value is calculated from T-value using a T-distribution table and degree of freedom(n-p-1)
5. It should be lower than the level of significance.
6. A level of significance is the threshold value above which the P-value becomes statistically significant. Most commonly used level of significance is 0.05.

4. R^2 Score

1. R^2 is the goodness-of-fit measure for a model. It is also called co-efficient of determination and tells us about how good a regression model fits the data.
2. R^2 is equal to the ratio of explainable variation (SSR) to total variation (TSS)
3. SSR is the difference between total actual variation (TSS) and unexplained variation (RSS) which gives us an idea of how much variation is explainable by our regression model.

$$R^2 = (TSS - RSS)/TSS = SSR/TSS$$

4. Total Sum of Squares [TSS] - sum of squares of difference between mean and actual output.
5. Residual Sum of Squares [RSS] - sum of squares of difference between predicted and actual output.
6. Regression Sum of Squares [SSR] - sum of squares due to regression.
7. A high value of R^2 like 80% would mean that the variation has reduced by 80% in the model or we can say that 80% of the variation is explained by the model.
8. Higher the R^2 value, the better the regression model fits the data.
9. The value of R^2 should be really close to 1 for linear data.
10. Value of R^2 can also be negative. It would suggest that the model has performed even worse than simply predicting the mean.

5. Adjusted R^2 Score

1. Adjusted R^2 is an optimised version of R^2, adjusted on the number of features.
2. R^2 increases with addition of new features whereas adjusted R^2 does not necessarily increase with addition of new features but only when a new feature actually enhances the model performance.

$$\text{Adjusted } R^2 = 1 - (RSS/TSS)$$

3. The value of adjusted R^2 goes up when a new feature enhances the model performance more than what was expected.
4. Adjusted R^2 is preferred over R^2. This is because the value of R^2 goes up whenever a new feature is added but the adjusted R^2 does not get affected by a change in number of features only.
5. The going up of R^2 by adding a new feature can tempt the users to add more features which often leads to overfitting.

6. Adjusted R^2 is always less than R^2.

6. F-Statistic

1. F-statistic is a statical measure calculated to test a null hypothesis where the null hypothesis states an assumption that no feature has any relationship with the output.
2. Null hypothesis
    1. Ho : B1 = B2 = B3 = 0, no feature has any relationship with output
    2. Ha : at least 1 feature has relationship with output

$$\text{F-statistic} = \frac{(TSS-RSS)/p}{RSS/(n-p-1)}$$

3. The formula is similar to R^2 but degree of freedom is taken into consideration. R^2 explains how our model fits to the data and f-statistic explains the total significance of model.
4. The difference between t-value and f-value is that t-value is calculated for a single feature whereas f-value is calculated for all features.
5. Even if we reject the null hypothesis for individual features but have a large set of total features, there is still a mathematical probability that none of the features have any significant relationship with the output.
6. A high f-value and low p-value is a good indication that there is at least one feature that has a significant relationship with the output.

REGULARISATION / SHRINKAGE METHODS

Need for Regularisation

- Regularisation methods are primarily used to solve the problem of high variance caused by multicollinearity. Multicollinearity occurs when there is high correlation between 2 or more features.
- It causes a problem because when there is high correlation between 2 features, they both have very similar impact on the output that leads to an overfit model, model that fits to the data too well.
- In other words, 2 features with similar impact on output creates a high variance model. Shrinkage methods adds some amount of bias to the model and creates a good bias-variance balance thus enhancing model performance.
- Basically, these methods are used to improve the model performance by just reducing variance.
- Regularisation methods involve training a model with all p number of parameters but shrink the feature coefficients towards zero.

1. Ridge [L2 regularisation]

1. Ridge regression is an optimised implementation of linear regression to reduce variance by introducing a penalty term to the loss function.
2. The task of ridge regression is to enhance the model performance by introducing a small amount of bias to reduce a significant amount of variance which is done by reducing slope.
3. RSS is reduced to find the best fit line but with an added term called shrinkage penalty.

$$RSS' = RSS + \text{shrinkage penalty}$$
$$\text{where shrinkage penalty} = \lambda \cdot \sum(Bi^2)$$

4. $\sum(Bi^2) \rightarrow$ penalty and $\lambda \rightarrow$ tuning parameter or penalty factor that determines the severity of penalty.

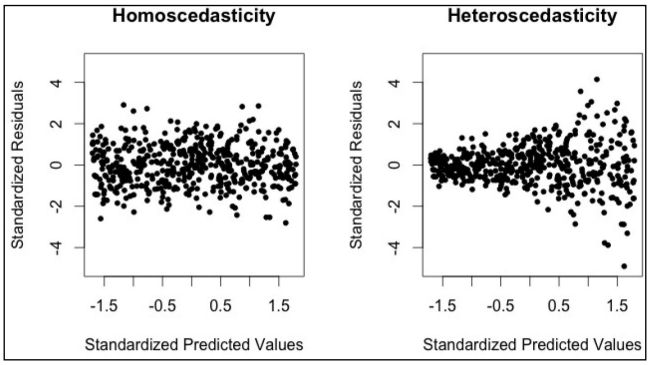
2. Lasso [L1 regularisation]

1. Ridge regression shrinks the feature coefficients towards zero but does not make them zero. Lasso regression will make the feature coefficients zero but not all of them. This method helps in feature reduction step also.
2. This regularisation method has a relatively stronger penalty term. Shrinkage penalty will use the absolute value of feature coefficients rather than the squares.

$$\text{shrinkage penalty} = \lambda \cdot \sum(Bi)$$

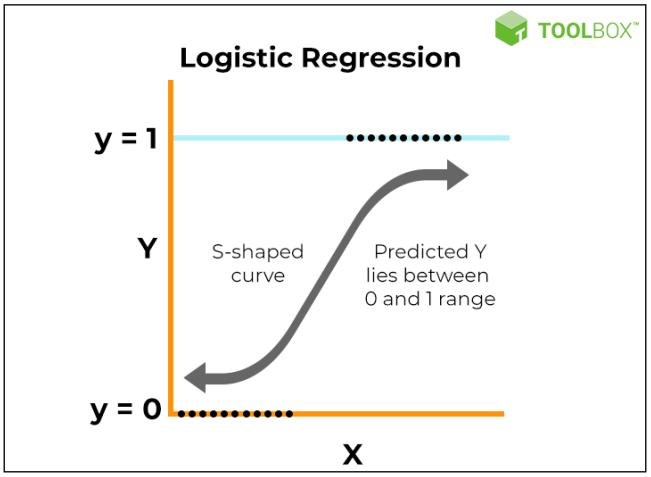
— Heteroscedasticity

1. While finding the best fit regression line, we make an assumption that variance is constant. But this assumption is possible for minimal number of cases. On the other hand, for most of the real-word cases, the variance will vary. This is known as heteroscedasticity.
2. This problem can be solved by scaling down the the observation [y] by using functions like logarithmic function, root function etc.



LOGISTIC REGRESSION

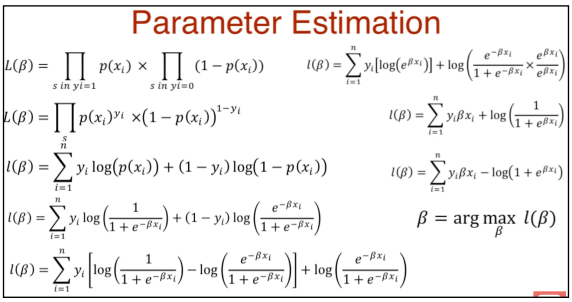
1. Logistic regression is a parametric machine learning model used for binary classification problems.
2. Logistic function is used to fit the data points, also called as sigmoid function having S shape.



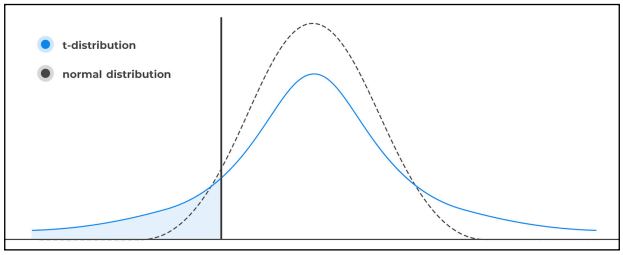
3. Linear regression finds the best fit line where output ranges from (-∞, +∞) but for classification, we want a probabilistic output ranging from 0 to 1 which is why logistic function is used.
4. Mathematical representation of logistic function:

   P(Y=1|x) = 1 / 1 + e^(-z)

   where P(Y=1|x) is probability that output for data point x is 1 and z = B0 + B1.x
5. We can see above that sigmoid function is basically converting the output of a linear function z into a probability. The goal of logistic regression model is to find probability that a data point belongs to a particular category and use boundary value to classify that data point.
6. Maximum Likelihood Estimation
   1. MLE is used to find the best fit sigmoid function for our data.
   2. The goal is to evaluate model parameters B0 and B1 that maximise the likelihood function.
   3. The same way we calculate RSS in Linear Regression and reduce it, we will calculate likelihood function L(B0, B1) and maximise it.
   4. Likelihood function is equal to product of probability of all data points. Evaluate the probability for all data points to get actual output.
   5. For output 1 — p and for output 0 — (1 - p)
   6. Find the product of all probabilities to get likelihood function. Generally log of likelihood function is used to find model parameters for relatively easy computation.



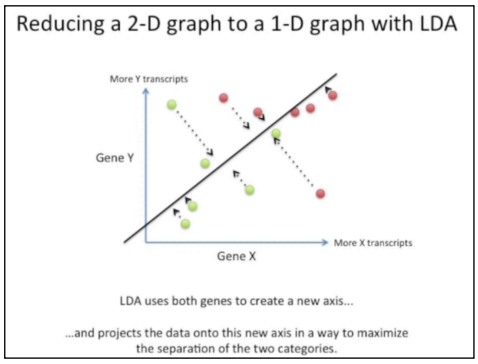
7. Boundary value is a value used to classify the probability as 0 and 1. Assuming we have 0.3 as a boundary value. All the values above 0.3 will be classified as 1 and all the values below will be classified as 0.
8. Like linear regression, logistic regression has a model summary having evaluation parameters like standard error, z-value and p-value. The role of model summary is to understand how the data and our logistic regression model fit together. All evaluation parameters in logistic regression serve the same purpose as linear regression.
9. Z-value and T-value have different names but serve similar purpose of rejecting the null hypothesis which states that there is no relationship between predictor variable and output. T-value is based on T-distribution and Z-value is based on standard normal distribution.
10. T-distribution is also symmetric and bell shaped like normal distribution but has tails approaching 0 later than normal distribution.

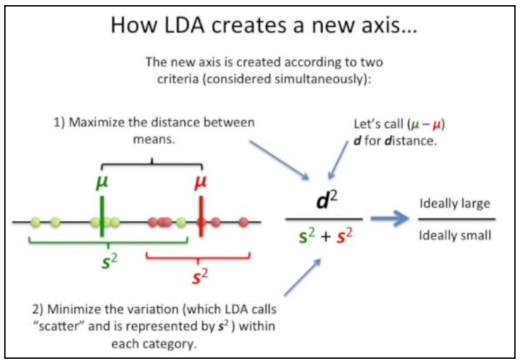


11. T-distribution is based on parameter known as degree of freedom. As degree of freedom (n-p-1, or we can say sample size) increases, T-distribution approaches normal distribution.
12. Advantages
    1. A simple and efficient model with less computation power to classify simple data.
    2. Assumes a linear relationship between predictor and output so performs well for data having a linear relationship.
    3. Gives a probabilistic output and uses a boundary value to classify. Hence it gives us a [predict_proba( ) function] and a freedom to control the classification based on our needs.
13. Disadvantages
    1. Good performance is restricted to data having a linear relationship. So may not perform well on non-linear data.
    2. By default, this model is limited to binary classification. Although there are extensions like multinomial logistic regression for multinomial classification.

LINEAR DISCRIMINANT ANALYSIS [LDA]

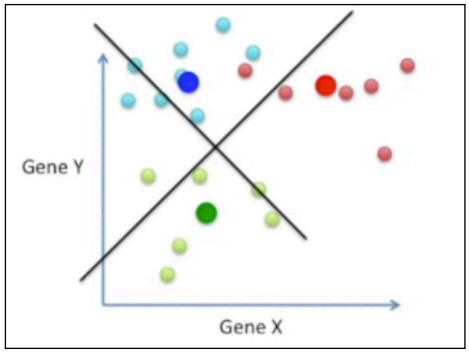
1. LDA is a parametric machine learning model used for dimensionality reduction and classification problems [generally more than 2 classes]
2. The main objective is to find a linear combination of features from already existing features in our dataset that best separates 2 or more classes. This is done by projecting the data points onto a lower dimensional space to find the best separation for classes.
3. Conditions for projecting data points onto lower dimensional space
    1. Maximise the distance between means. [Squared to get positive output only]
    2. Minimise the variance of separate classes.

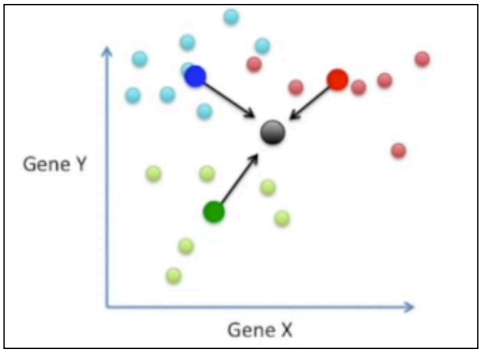




4. LDA for classification of 3 or more classes

1. LDA will create 2 axes for separating 3 classes. This means than LDA will create n-1 axes for n number of classes.
2. Instead of maximising the distance between means of 2 classes, we find a centre point and maximise the sum of squares of distance between mean and centre point while minimising the variance of individual classes.



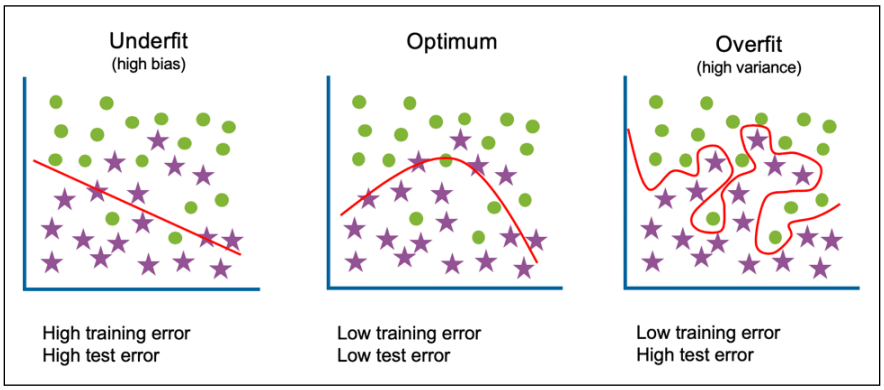


5. Assume that instead of 2 genes, we have 1000 genes [1000 features] and a total of 3 classes. LDA makes it possible to iteratively project 1000 genes down to 2 genes and find 2 best separable axes for classification.
6. Advantages
    1. LDA excels at feature reduction. Can be used as a preprocessing step to solve problems like collinearity, curse of dimensionality and high computation expense before feeding the data to another machine learning model.
    2. LDA can also be used as an unsupervised machine learning model.
7. Disadvantages
    1. LDA makes an assumption that data follows a normal distribution for each class. Hence performance of LDA can downgrade if the data does not follow a normal distribution.
    2. LDA can only find linear boundaries between classes. If the data is more complex and not linearly separable, the performance of LDA will not be upto the mark.
8. Extensions of LDA
    1. Quadratic Discriminant Analysis — LDA assumes all classes have the same variance. QDA relaxes this assumption, allowing each class to have its own variance structure, making it more flexible for complex data.
    2. Flexible Discriminant Analysis — LDA finds linear boundaries between classes. FDA allows for non-linear decision boundaries using techniques like splines, which can be useful when data isn't linearly separable.
    3. Regularised Discriminant Analysis — LDA can be sensitive to irrelevant features and suffer from overfitting. RDA introduces regularisation to the model, reducing this sensitivity and improving performance.

K-NEAREST NEIGHBORS

1. KNN is a non-parametric machine learning model generally used for both classification and regression problems.
2. It does not assume or create a mathematical relationship between predictors and output like other machine learning algorithms and it only stores data in the memory space during its training phase.
3. Prediction
    1. The model calculates the distance between new observation and all other data points stored inside the memory.
    2. The model finds the data points with shortest distance [nearest neighbours] to the new observation. The number of nearest neighbours that the model finds depends on K hyperparameter. If K = 5, model will find 5 nearest data points to the new observation.
    3. Classification — the model will calculate probability of classes for K nearest neighbours and allocate the class with highest probability to the new observation.
    4. Regression — the model will calculate the average of outputs for K nearest neighbours and allocate the value to new observation.

4. K Hyperparameter
    1. If K is very small, the model will become highly sensitive to individual data points and overfit.
    2. If K is very high, the model will be unable to interpret the underlying trend of the data and under fit.
    3. Cross-validation is a good technique to find the ideal value for K.



5. It is also important to normalise the data before using KNN model because KNN calculates distance between a new observation and all data points. While calculating the distance, the predictor variables with larger range will have larger impact on the output.
6. Distance metrics
    1. Feature Scaling: Euclidean distance can be sensitive to features with different scales. In such cases, normalisation or standardisation might be necessary.
    2. Data Types: Manhattan distance is suitable for categorical data, while Hamming distance is ideal for binary data.
    3. Curse of Dimensionality: As dimensionality increases, some metrics like Euclidean distance might lose their effectiveness.
7. Advantages
    1. KNN works well for non-linear data and multinomial classification unlike logistic regression that is limited to binary classification and performs well for only linear data.
    2. KNN does not make any prior assumptions like most of the other machine learning models.
8. Disadvantages
    1. KNN is computationally very expensive because it calculates distance between a new observation and all other data points. So it is not preferable to use KNN for large datasets and hence it is not very scalable.
    2. KNN can be a burden on memory space because all KNN does during training is store the data in memory space and use it to classify a new observation.

SUPPORT VECTORS

1. Maximal Margin Classifier [MMC]
    1. Maximal Margin Classifier is a non-parametric machine learning model used for binary classification problems.
    2. The basic idea would be to find a hyperplane that best separates the datapoints of both classes.
    3. Now there may be infinitely possible hyperplanes that separates the datapoints. This is where the concept of maximum margin comes into play. It makes sure that the hyperplane chosen out of the all possible hyperplanes is the most optimal one.
    4. The data points nearest to a hyperplane are called support vectors.
    5. The distance between a hyperplane and support vectors is called margin.
    6. The goal is to find a separation where the margin is maximum, distance between support vectors is maximum. This way we get a hyperplane that separates the datapoints in a way that datapoints of both classes are most away from each other.
    7. Mathematical explanation —
        A. For decision boundary [hyperplane] —

1. We know that [y = mx + c] represents a line where m is slope and c is an intercept. And it can be changed as [mx - y + c = 0] where x and y tell us that this line exists for 2D space. For more than 2D space —

$$\vec{w}^T.x + b = 0$$

$$\text{or } (\vec{w}_1 x_1 + \vec{w}_2 x_2 + \ldots + \vec{w}_i x_i) + b = 0$$

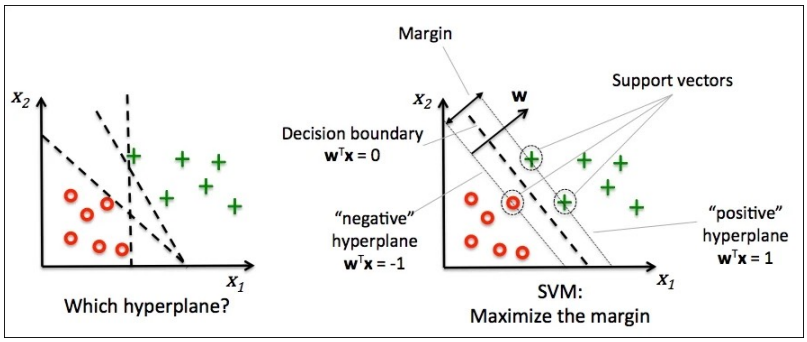
and where w is slope, x is data points coordinates and b is intercept

2. So, decision boundary for SVM linear model which allows us to categorise data points with a decision boundary. —

$$\vec{w}^T.x + b = 0$$

where $\vec{w}^T.x + b > 0$ infers True output

and $\vec{w}^T.x + b < 0$ infers False output



B. For maximum margin [support vectors] —

1. true side marginal plane can be represented as — $[\vec{w}^T.x + b = +k]$
2. false side marginal plane can be represented as — $[\vec{w}^T.x + b = -k]$
3. where k is the distance between hyperplane and support vectors.
4. maximise the distance between marginal planes for best separation —

$$\text{true side marginal plane - false side marginal plane}$$

$$(\vec{w}^T.x_1 + b = +k) - (\vec{w}^T.x_2 + b = -k)$$

where x_1 is true side SV and x_2 is false side SV

$$\vec{w}^T.(x1 - x2) = 2k$$

5. slope w^T consists of 2 components — vector and magnitude. To get vector, / both sides by magnitude |w|

$$\vec{w}^T.(x1 - x2) / |w| = 2k / |w|$$

6. maximise $2k / |w|$ for best separation [or minimise $|w| / 2k$]

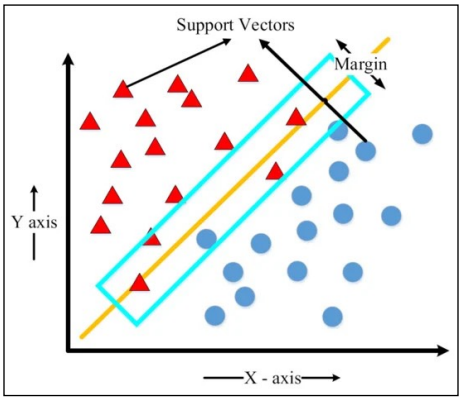
C. From above —

1. for accurate predictions, $y_i = 1$ when $\vec{w}^T x + b > 1$ —— $y_i * (\vec{w}^T x + b) > 1$
2. $y_i = -1$ when $\vec{w}^T x + b < -1$ —— $y_i * (\vec{w}^T x + b) > 1$
3. Hence for any accurate prediction, true or false, $y_i * (\vec{w}^T x + b)$ will always be greater than 1.

D. Constraint — minimise cost function = (|w| / 2k) with constraint $y_i * (\vec{w}^T x + b) > 1$

8. Disadvantages
   1. MMC can only classify data that is linearly and perfectly separable.
   2. Sensitive to outliers — the position of hyperplane changes if new datapoints are added to the training set. Now if outliers are added to the data, the position of hyperplane changes significantly.
   3. No relevance to real-world problems — because real world data is more complicated and never separable perfectly by a single hyperplane.

2. Support Vector Classifier [Soft Margin Classifier]
   1. Support Vector Classifier is a non-parametric machine learning model used for classification problems. It is an optimised implementation of maximal margin classifier.



2. Maximal margin classifier calculates hard margin to find the hyperplane that best separates the datapoints of both classes whereas SVM calculates soft margin to find the best hyperplane.
3. A margin that separates the datapoints of both classes perfectly is called a hard margin. A margin that separates the datapoints but allows some amount of misclassifications is called soft margin.
4. Support vectors are the data points that define the decision boundary like maximal margin classifier but the number of support vectors in support vector classifier is a lot more because this method allows misclassifications and data points inside soft margin.
5. Additional features of SVC over MMC
    1. Real-world data in most of the cases is not perfectly separable by a hyperplane. There may be some data points of class A lying in the region of class B.
    2. To solve this problem, SVC finds the best hyperplane and at the same time allows some amount of misclassifications and presence of data points inside the margin.
    3. Mathematically, SVC calculates soft margin to find the best hyperplane instead of hard margin. Soft margin introduces a penalty term and penalises each misclassified datapoint and datapoints present inside the margin.
    4. The size of soft margin can be controlled by controlling the cost-multiplier [C] — how many points we can allow to get misclassified.

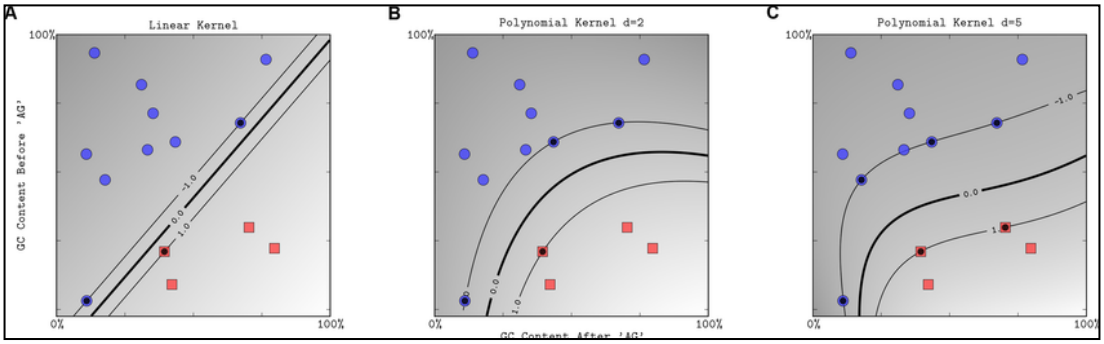
        Cost function = (|w| / 2k) + C. ∑ |eta|

        where C is the number of misclassifications allowed and eta is the distance between misclassified data point with marginal hyperplane and |eta| because distance can be negative and positive.

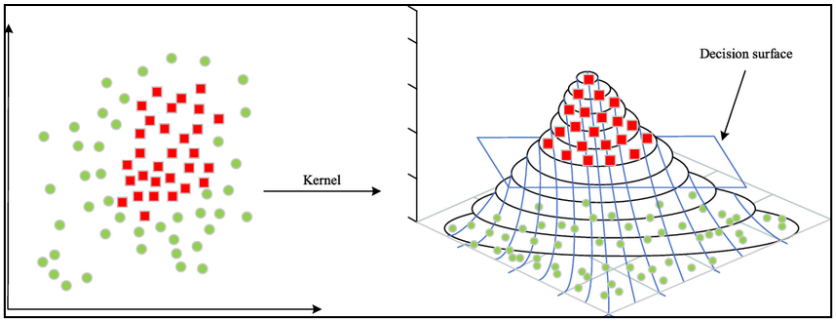
        1. Together [C. ∑] eta is called Hinge Loss in SVC.
        2. A high value of C will result in small margin which will allow less misclassifications.
        3. A small value of C will result in large margin which will allow more misclassifications. This will improve generalisation of the model, prevent overfit [good testing accuracy]
        4. **Constraint — minimise cost function = (|w| / 2) + C. ∑ |eta| with constraint $y_i * (w^T x+b) > 1$**
        5. **The task to minimise the cost function is same for SVC and SVR but they follow separate constraints.**
6. Advantages
    1. Application to some real-world problems — we can classify real-world data that is not perfectly separable, have class outliers.
    2. Generalisation — SVC allows misclassification that makes it robust to outliers and improves generalisation of the model. In other words, SVM prevents overfit.
7. Disadvantages
    1. SVC can only classify data that is linearly separable. But most of the real-world data is non-linearly separable i.e. can not be separated by a linear hyperplane.

3. Support Vector Machine [Kernel Trick]
    1. Support Vector Machine is a non-parametric machine learning model used for classification problems. It is an optimised implementation of support vector classifier.
    2. SVM uses kernels to classify real-world data that is non-linearly separable. In other words, SVM can handle the drawback of SVC of not being able to classify real-world data that is inseparable by a linear hyperplane.
    3. Kernels are basically mathematical functions that projects our data points from a complex low-dimensional space to a simple high dimensional space where it can be classified linearly with a linear hyperplane.

4. So SVM uses the same method of finding a linear hyperplane to classify data points but transforms those datapoints beforehand in a way where they become linearly separable. As shown below.
5. Types of kernel functions used in support vector machines —
    1. Linear kernel
        1. Used for already linearly separable data.
        2. They do not project data to a higher dimensional space.
        3. So SVMs using linear kernels are basically SVCs.
    2. Polynomial kernel
        1. Used for non-linearly separable data [data separable by curves]
        2. Projects data to a higher dimensional space using polynomial function.
        3. Degree, a polynomial kernel hyperparameter refers to the highest exponent used to create new features for the data points. It essentially controls the complexity of the features used to represent the data in a higher-dimensional space.
        4. High value of degree can capture more complex non-linear relationships but can overfit. Low value of degree can make the model more generalised but may not be able to capture complex relationships.

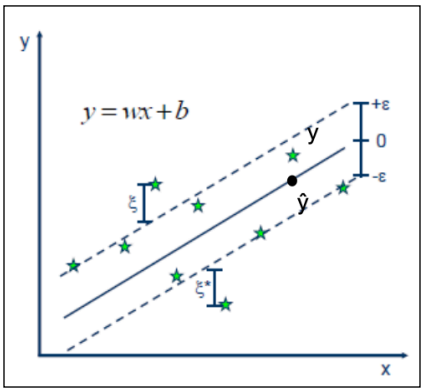


    3. Radial Basis Function [RBF] kernel
        1. Used for non-linearly separable data.
        2. Projects data to a higher dimensional space.
        3. Radial function calculates similarity based on distance between data points. Low distance means high similarity and vice-versa.
        4. Gamma, a radial kernel hyperparameter that handles the influence of distance on similarity between data points. High value of gamma makes similarity more sensitive to distance i.e. only data points very close to each other will be considered similar.
        5. High value of gamma can lead to overfit the model whereas low value of gamma can create more generalised model.



4. Support Vector Regressor
    1. SVR is a non-parametric machine learning model used for regression problems.
    2. The main idea behind SVR is to find a best fit line and marginal planes like SVC.
    3. But here we do not minimise the error for our best fit line like linear regression, but maximise the margin between hyperplane and support vectors.
    4. Mathematical explanation —
        1. For hyperplane and margin —
            1. hyperplane — $\vec{w}^T.x + b = 0$

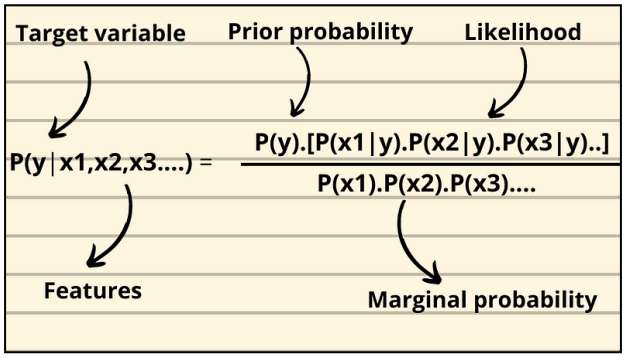
2. As per picture below, marginal planes — $\vec{w}^T$.x + b = +ε and $\vec{w}^T$.x + b = -ε where ε [epsilon] is the margin.
3. Calculation same as SVC — minimise (|w| / 2ε) + C.∑ |eta| to maximise margin.
4. where C.∑ |eta| is hinge loss. Hinge loss will be 0 for data points inside margin because eta is the distance between marginal plane and data points outside margin only.



2. Constraint —
    1. ŷ — predicted value on hyperplane
    2. y — actual value where y = $\vec{w}^T$.x + b]
    3. Looking at picture above, y - ŷ < ε infers a good prediction. But if y was below ŷ, then ŷ - y < ε would infer good prediction. As a whole —
    $$|y - \hat{y}| < \varepsilon$$
    4. But again, this is a constraint for data points inside margin.
    5. For data points outside margin —
    $$|y - \hat{y}| < \varepsilon + |eta|$$
3. **Final — minimise (|w| / 2ε) + C.∑ |eta| with constraint |y - ŷ| < ε + |eta|**

NAIVE BAYES

1. Naive Bayes is a non-parametric machine learning model used for classification problems and is based on Bayes Theorem.
2. Bayes theorem is a concept in probability which allows us to calculate conditional probability for dependant events. Conditional probability is the probability of an event B happening given an event A has already occurred.
$$P(B \mid A) = (P(A \mid B) * P(B)) / P(A)$$
3. Example — The task is to find if an email is spam or not based on a set of features. The idea behind Naive Bayes algorithm is to calculate the probability of an email as spam given the features.
4. Assumption — Naive Bayes makes a strong assumption of feature independence. It assumes that the features are conditionally independent of each other. In other words, the presence or absence of one feature does not influence the probability of another feature given the class.
5. Prediction
    1. Assume a dataset has features — [x1, x2, x3] and output — [Y, N] {see below picture}
    2. The model calculates —
        1. prior probability — the initial probability of each class.
        2. likelihood — product of conditional probability of features given the output.
    3. Marginal probability of features is not calculated because it is constant. Marginal probability while calculating conditional probability for all classes given the same features will always be the same.
    4. For prediction, an unseen data is fed to the model and the model will calculate conditional probability for each class given the unseen data [x1, x2, x3]
    5. The class with higher conditional probability will be allocated.
6. Advantages
    1. Low computational cost
    2. Handles Missing Values — ignores observations with missing values while calculating probability terms.



3. Small Training Data — works well with small training data. A good option for classification problems where big training data is not present.
4. High Feature Count — works well for data with high feature count. A good option when dealing with curse of dimensionality.
5. Interpretable — predict_proba( ) functions returns conditional probability of input belonging each class. This way we can identify how confident is the classification for specific input.

7. Disadvantages
    1. Assumption — Naive Bayes presumes that all the features are conditionally independent which is hardly possible for real-world data.
    2. Zero Frequency Problem — Naive Bayes also has a zero-frequency problem. If a category in testing set is not present in the training set, then the conditional probability for that category given the output will be 0 and hence the conditional probability will become 0.
8. Types of Naive Bayes
    1. Bernoulli Naive Bayes
        1. Most basic form of Naive Bayes.
        2. Works best with binary features [0, 1]
        3. Example — spam email classification where features represent the presence (1) or absence (0) of certain keywords. The model predicts an email as spam or not spam based on the presence of words in an email.
    2. Multinomial Naive Bayes
        1. An extension of Bernoulli Naive Bayes
        2. Handles discrete features that can take on multiple values [-1, 0 , 1, 2, 3]
        3. The features are usually counts.
        4. Example — text classification where features represent word counts in a document. The model predicts the class of a document based on word counts in a document.
    3. Gaussian Naive Bayes
        1. The most complicated Naive Bayes of the three
        2. Deals with continuous features, that can take on any real value within a range.
        3. Assumption — the features follow a Gaussian distribution [Normal distribution]
        4. Often used for tasks like predicting housing prices based on features like size, location, and number of bedrooms, where each feature can have a range of values.

CLASSIFICATION EVALUATION

1. Accuracy
    1. Measures the correctness of the model.
    2. Equal to ratio of number of true predictions to total number of predictions.

$$accuracy = (TP + TN) / (TP + TN + FP + FN)$$

    3. Example, 75% accuracy of spam emails. This means that 75% predictions of the total predictions were correctly predicted.

2. Precision
    1. Equal to ratio of number of true positives to total number of predicted positives.

$$precision = TP / (TP + FP)$$

    2. Example, 75% precision of spam emails. This means that out of all the emails predicted as spam, 75% were actual spam emails.

3. Recall [Sensitivity] [True Positive Rate]

1. Equal to ratio of number of true positives to total number of actual positives.

   recall = TP / (TP + FN)
2. Example, 75% recall of spam emails. This means that out of all the spam emails, 75% were predicted as spam.

4. F-1 Score
   1. Harmonic mean of precision and recall.

      F1 score = (2 * precision * recall) / (Precision + Recall)
   2. Provides a balance between precision and recall.
   3. Good for imbalanced dataset.

5. Specificity [True Negative Rate]
   1. Equal to ratio of number of true negatives to total number of actual negatives.

      Specificity = TN / (TN + FP)
   2. Example, 75% specificity of spam emails. This means that out of total number of non-spam emails, 75% were predicted as non-spam emails.

6. ROC Curve [Receiver Operating Characteristic]
   1. A graphical plot between True Positive Rate [TPR] and False Positive Rate [FPR]
   2. TPR is recall or sensitivity.
   3. FPR is equal to ratio of number of false positives to total number of actual negatives.

      FPR = FP / (FP + TN)
   4. We know that a machine learning model for classification problems give a probabilistic output and a boundary value is used to classify a data point. ROC curve is used to decide that optimal boundary value for high level of performance.

7. AUC [Area Under Curve]
   1. Total area under the ROC curve.
   2. Tells about the overall performance of classification model i.e how well the model classifies the data points.
   3. The range is (0, 1) i.e. 1 indicating a good classifier and 0 indicating a poor classifier.
   4. AUC can be calculated for multiple classification models to find the model with highest overall performance.