greatlearning
Learning for Life

Logistic Regression

greatlearning
Learning for Life

Agenda

1. Introduction to Logistic Regression
2. How is the model built
3. Logistic Regression Learning process
4. Assumptions of Logistic Regression
5. Evaluating Logistic Regression model
6. Variants of Logistic Regression
7. Applications of Logistic Regression
8. Advantages and disadvantages of Logistic Regression

What is Logistic Regression

Logistic Regression (What is it) -

1. Also known as Logit , Maximum-Entropy classifier, is a supervised learning method for classification. It establishes relation between dependent class variable and independent variables using regression
2. The dependent variable is categorical i.e. it can take only integral values representing different classes
3. The probabilities describing the possible outcomes of a query point are modeled using a logistic function
4. Belongs to family of discriminative classifiers. They rely on attributes which discriminate the classes well
5. There are two broad categories of Logistic Regression algorithms
    a. Binary Logistic Regression when dependent variable is strictly binary
    b. Multinomial Logistic Regression when the dependent variable has multiple categories. There are two types of Multinomial Logistic Regression
        I. Ordered Multinomial Logistic Regression (dependent variable has ordered values)
        II. Nominal Multinomial Logistic Regression (dependent variable has unordered categories)

Logistic Regression (What is it) -

6. Part of Sklearn linear model library. This implementation supports binary , One-vs-Rest, multinomial logistic regression with optional l1, l2, Elastic-Net regularization
7. The default approach is OVR (One Vs Rest) scheme , regularized using L2 method with ‘lbfgs’ solver**

greatlearning
Learning for Life

How logistic Regression models building blocks

Building Blocks of Logistic Regression (How is it built)-

1. Logistic Regression assigns probabilities to different classes to which a query point is likely to belong

2. To do so, it learns from the training set a vectors of weights and bias. Each weight (w_i) is assigned to one input feature Xi

3. The weight assigned to each feature represents how important that feature is for classification decision

4. The weights can be positive i.e. direct correlation of the feature with the class of interest** while a negative weight indicates inverse relation with the class of interest

5. To classify a query point, the classifier takes the weight sum of the features and the bias to represent the evidence of the query point belonging to the class of interest
    1. **$Z = w.x + b$**

Building Blocks of Logistic Regression (How is it built)-

6. Since the weights are running numbers and so is the bias term, Z can take values from – infinity to + infinity

7. To transform the value of Z into probability (range between 0 and 1) , Z is passed through Sigmoid function (mathematical transformation)
   1. **$P(y= 1) = Sigmoid(Z) = 1/(1 + e^{-z})$**
   2. **$P(y= 0) = 1 – P(y =1) = 1 – (1/(1 + e^{-z})) = e^{-z} / (1 + e^{-z})$**
   3. **y = 1 if P(y=1|X) > .5, else y = 0**

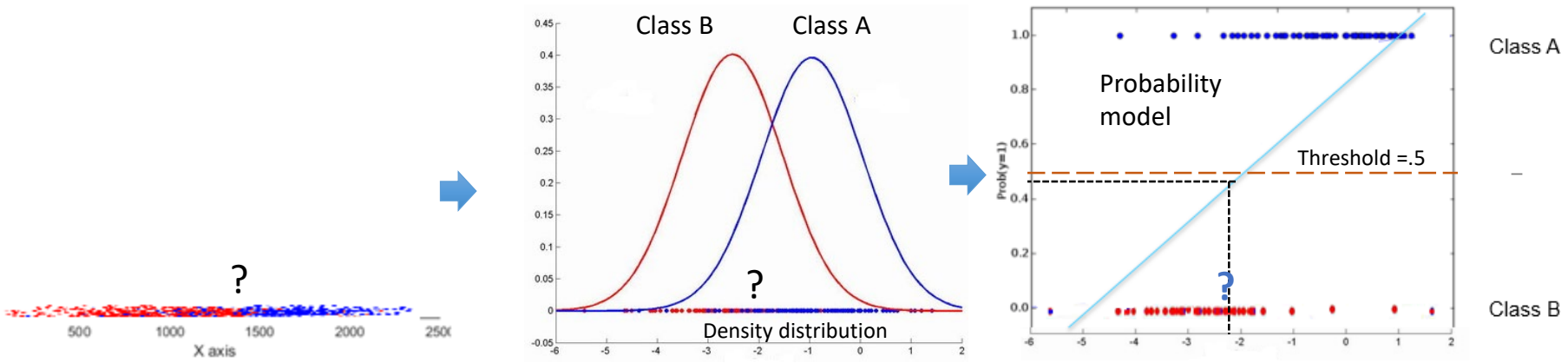
8. The algorithm uses cross-entropy loss function (negative log likelihood loss) to find the most optimal weights and bias across entire data set put together (N records)

$$logLoss = \frac{-1}{N}\sum_{i=1}^{N}(y_i(logp_i) + (1 - y_i)log(1 - p_i))$$

9. Most optimal weights and bias would be those that minimize overall all training error i.e. misclassification in the training data

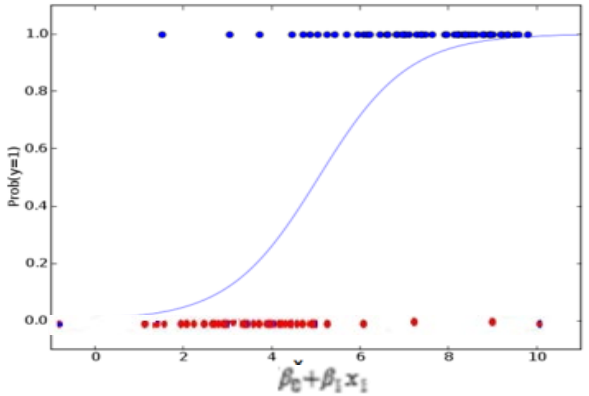
Building Blocks of Logistic Regression (How is it built)-

10. Imagine you have to build a model to identity potential defaulters. You found an interesting feature called MHE (Monthly Household Expenditure)
11. You notice that as MHE increases, the density of defaulters (class A - blue points) increases. There are relatively more non-defaulters(class B - red points) on lower side of MHE
12. A new data point (shown with "?") needs to be classified i.e. does it belong to class A or B. Let class A be 1 and class B be 0 on the vertical axis
13. When Y axis represents probability of default, it has a direct positive correlation with class A
14. One can fit a simple linear model (y = ßx +c) where y greater than a threshold means point most probably belongs to class A but for extreme values of x, probability is <0 or >1 which is **absurd**



Building Blocks of Logistic Regression (How is it built)-

14. The linear model is passed to a logistic function **p** = 1/ 1 + e^{-y} the result of which is values between 0 and 1. Thus **p** represents probability a data point belongs to class "A" given x



Class A

Class B

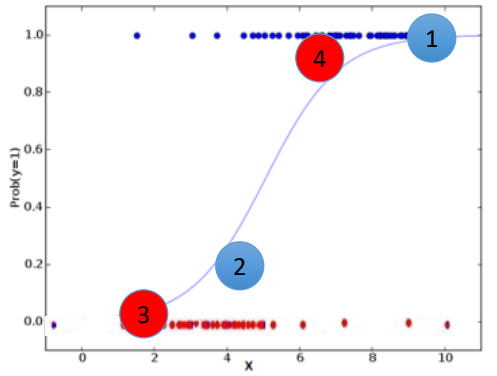
15. Instead of using y of linear model as dependent, it's function shown as "p" is used as dependent variable $p = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_q x_q)}}$. This is logistic response function

16. It is a two step model. In first step, the propensity to belong to class 1 i.e P(1|X), followed by next step of using cut-off to decide the class

Building Blocks of Logistic Regression (How is it built)-

17. There can be four difference cases for the value of yi and pi (predicted probability of class 1 (blue color)

Case 1 : $y_i = 1$, p_i = High , $1 - y_i = 0$, $1 - p_i$ = Low	Correct classification
Case 2 : $y_i = 1$, p_i = Low , $1 - y_i = 0$, $1 - p_i$ = High	Incorrect classification
Case 3 : $y_i = 0$, p_i = Low , $1 - y_i = 1$, $1 - p_i$ = High	Correct classification
Case 4 : $y_i = 0$, p_i = High , $1 - y_i = 1$, $1 - p_i$ = Low	Incorrect classification



18. The loss function being logloss or Cross Entropy

$$logLoss = \frac{-1}{N}\sum_{i=1}^{N}(y_i(logp_i) + (1 - y_i)log(1 - p_i))$$

19. Incorrect classification will add large magnitude to the loss function while correct classification will contribute very minimal to the loss function

20. Even correct classification will add to the loss function! But will be miniscule. For 0 loss the predicted probability should be exactly 1 or 0 which is not possible as those values are achieved only at infinity

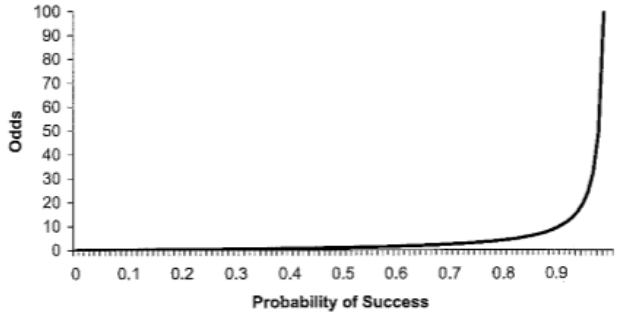
Note: -
log of 1 = 0
log of 0 = - ∞ (approach neg infinity)
log returns –ve numbers between 0 and 1
Yi = 1 or 0 only (numerical labels)

Logistic Regression models Learning Process

Building Blocks of Logistic Regression (Learning Process)-

1. The output is the probability of belonging to a class. Probability can also be expressed in from of odds.
2. Odds have a property of ranging from 0 to infinity that makes it easy to map a regression equation to odds. That is why logistic model uses odds
3. The odds of belonging to class y = 1 is defined as the ratio of probability of belonging to class 1 to probability of belonging to class 0

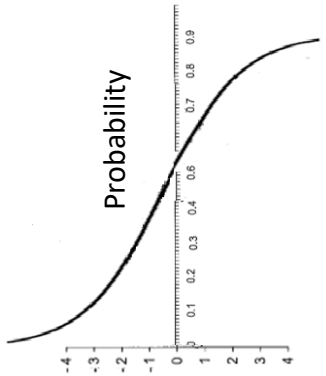
$$\text{Odds}(Y = 1) = \frac{p}{1-p}.$$



4. If probability of belonging to class Y=1 is .5 then Odds(Y=1) = 1

Building Blocks of Logistic Regression (Learning Process)-

5. Thus probability $= \frac{\text{Odds}}{1+\text{Odds}}$
6. Probability is also $p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_q x_q)}}$.
7. Therefore $e^{(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_q x_q)}$ = Odds (y = 1)
8. The expression reflects the relation between predictors and dependent variable
9. Take log on both sides $\log(\text{Odds}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q$. This is logit function
10. During the training stage, the ß coefficients are adjusted such as the average logloss (cross entropy is minimized)



Steps -

1. Given x1,x2…xn, in training set , find $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$
2. This is log(odds)
3. Find odds by raising it to e
4. Find the probability using the equation $\frac{\text{Odds}}{1+\text{Odds}}$

Building Blocks of Logistic Regression (Learning Process)-

Suppose the log(odds) = -17.2086 + (.5934 x)

For a given value of x = 31, log(odds) = -17.2086 + (.5934 * 31) = 1.1868

Odds = exp(log(odds)) = exp(1.1868) = 3.2766

Probability = odds / (1 + odds) = 3.2766/(1 + 3.2766) = .7662

greatlearning
Learning for Life

Assumptions in Logistic Regression

Assumptions of Logistic Regression

1. Dependent variable is categorical. Dichotomous for binary logistic regression and multi label for multi-class classification

2. Attributes and log odds i.e. log(p / 1-p) should be linearly related to the independent variables

3. Attributes are independent of each other (low or no multi-collinearity)

4. In binary logistic regression class of interest is coded with 1 and other class 0

5. In multi-class classification using Multinomial Logistic Regression or OVR scheme, class of interest is coded 1 and rest 0 (this is done by the algorithm)

Note: the assumptions of Linear Regression such as homoscedasticity , normal distribution of error terms, linear relation between dependent and independent variables are not required here.

greatlearning
Learning for Life

Evaluating Logistic Regression models

Classification Model Metrics

a. Confusion Matrix – A 2X2 tabular structure reflecting the performance of the model in four blocks

Confusion Matrix	Predicted Positive	Predicted Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

b. Accuracy – How accurately / cleanly does the model classify the data points. Lesser the false predictions, more the accuracy

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN)$$

c. Sensitivity / Recall – How many of the actual True data points are identified as True data points by the model . Remember, False Negatives are those data points which should have been identified as True.

$$\text{Recall} = TP / TP + FN$$

d. Specificity – How many of the actual Negative data points are identified as negative by the model

$$SPEC = \frac{TN}{TN + FP}$$

e. Precision – Among the points identified as Positive by the model, how many are really Positive

$$\text{Precision} = TP / TP + FP$$

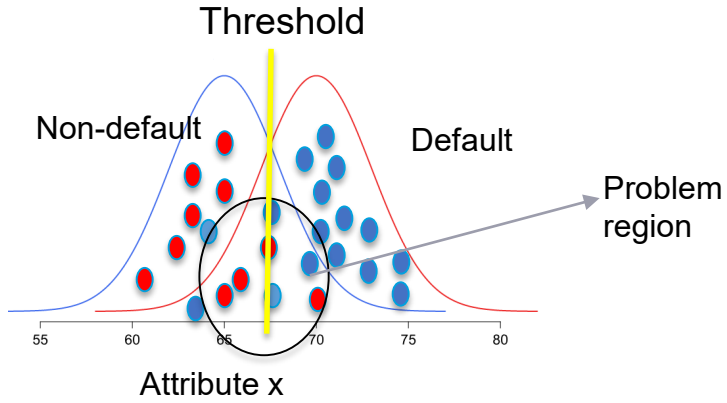
Classification Model Metrics

Assume model is identifying defaulters. In this binary classification defaulter class is class of interest and labeled as +ive (positive - 1) class, other class is –ve(negative - 0)

1. True Positives - cases where the actual class of the data point and the predicted is same. For e.g. a defaulter (1) predicted as defaulter (1)
2. True Negatives – cases where the actual class was non-defaulter and the prediction also was non-defaulter
3. False Positives – cases where actual class was negative (0) but predicted as defaulter (1)
4. False Negatives – cases where the actual class was positive (1) but predicted as non-defaulter (0)
5. Ideal scenario will be when all positives are predicted as positives and all negatives are predicted as negatives

Classification Model Metrics

6. In practical world this will never be the case. There will be some false positives and false negatives
7. Our objective will be to minimize both but the problem is, when we minimize one the other will increase and vice versa!
8. The problem is in the overlap region in the distributions



9. Objective will be to minimize one of the error types, either the false positive or false negative

Classification Model Metrics

10. Minimize false negatives - if predicting a positive case as negative is going to be more detrimental for e.g. predicting a potential defaulter (positive) as non-defaulter (negative)
11. Minimize false positives – if predicting a negative as positive is going to be more detrimental for e.g. predicting a boss's mail as spam!
12. Accuracy – over all correct predictions from all the classes to total number of cases. Should rely on this metrics only when all classes are equally represented. Not reliable if class representation is lopsided as algorithms are biased towards over represented class
13. Precision - TP/ TP+ FP. When we focus on minimizing false negatives, TP will increase but along with it FP will also increase. How much increase in TP starts hurting (due to increase in FP) ?
14. Recall – TP / TP + FN : when we reduce FN to increase TP, how much we gain ? Recall and precision will oppose each other. We want recall to be as close to 1 as possible without precision being too bad

greatlearning
Learning for Life

Variants of Logistic Regression

Variants of Logistic Regcression

Multinomial logistic regression

1. It is used to predict a nominal dependent variable given one or more independent variables.

2. It is sometimes considered as extension of binomial logistic regression to allow for a dependent variable with more than 2 categories.

3. It is used to model nominal outcome variables, in which the log odds of the outcomes are modelled as linear combination of predictor variables.

Eg: If a high school student wants to choose a program among general program, biological program and academic program, then their choice might be modelled using their normal scores and economic status.

Variants of Logistic Regression

Ordinal logistic regression

1. This regression is used to predict an ordinal dependent variable given one or more independent variables.

2. The model only applies to data that meet the proportional odd assumption. In this assumption, the event which is modelled does not have an outcome in a single category as the way it is done in the binary models and multinomial models.

3. In proportional odds assumption model, each end result has its own intercept but similar regression coefficients which means:
   a. The overall odds of any event can differ.
   b. The effect of the predictors on the odds of an event occurring in very subsequent category is the same for every category. It is often violated.

Applications of Logistic Regression

Applications of Logistic Regression

1. Predicting weather: you can only have few definite weather types. Stormy, sunny, cloudy, rainy and a few more.

2. Medical diagnosis: given the symptoms predict the disease patient is suffering from.

3. If loan has to be given a particular candidate depend on his identity check, account summary, any properties he hold, any previous loan, etc

Logistic Regression… pros and cons

Applications of Logistic Regression (Pros and Cons)

Advantages –

1. Simple to implement and easier to interpret the outputs coefficients
2. Provides both probabilities and classes as output
3. Quick to train as the error function (cross entropy) is convex , smooth and continuous

Disadvantages -

1. Assumes a linear relationships between log odds and independent variables.
2. Can stop learning (convergence of weights) in presence of good separators of classes as attributes. Such attributes will get a very high magnitude weights. That will need appropriate regularization to make the model learn and generalize
3. Outliers can have huge adverse impacts on the log odds regression
4. Assumes the attributes to be independent which is generally not the case

greatlearning
Learning for Life

Thank You

greatlearning
Learning for Life

Modelling Errors

Modelling Errors

All models are impacted by three types of errors which reduce their predicting power.

1. Variance errors
2. Bias error
3. Random errors

Variance errors

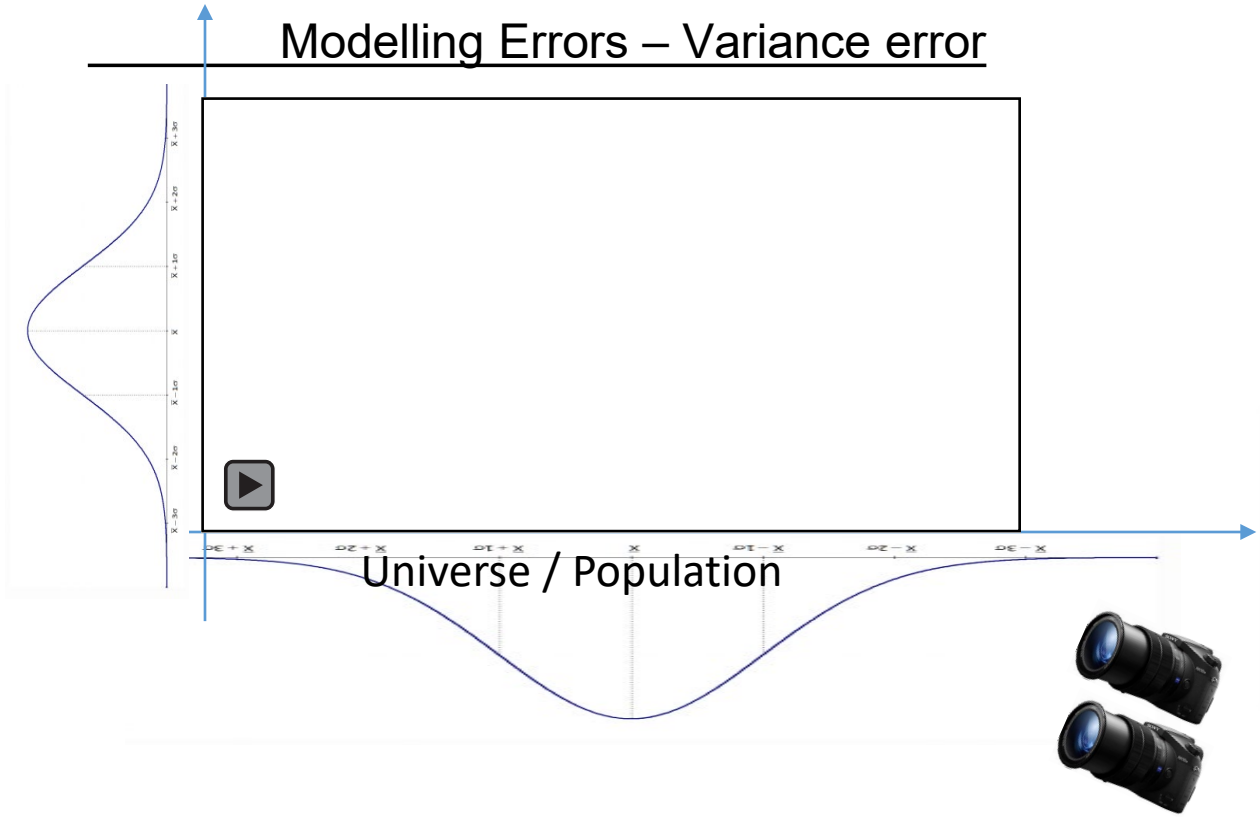
1. Caused by the random factors that impact the process that generate the data
2. The population / universe, representing the infinite data points continuously jiggle
3. Sample drawn from such universe is a snapshot of a small part of the universe
4. The model based on a sample will perform differently on different samples
5. Variance errors increase with increase in number of attributes in the model due to increase in degrees of freedom for the data points to wriggle in

Bias errors

1. Caused by our selection of the attributes and our interpretation of their influence on each other
2. The real model in the universe / population may have many more attributes and the attributes interacting in different ways not reflected in our model

Random errors

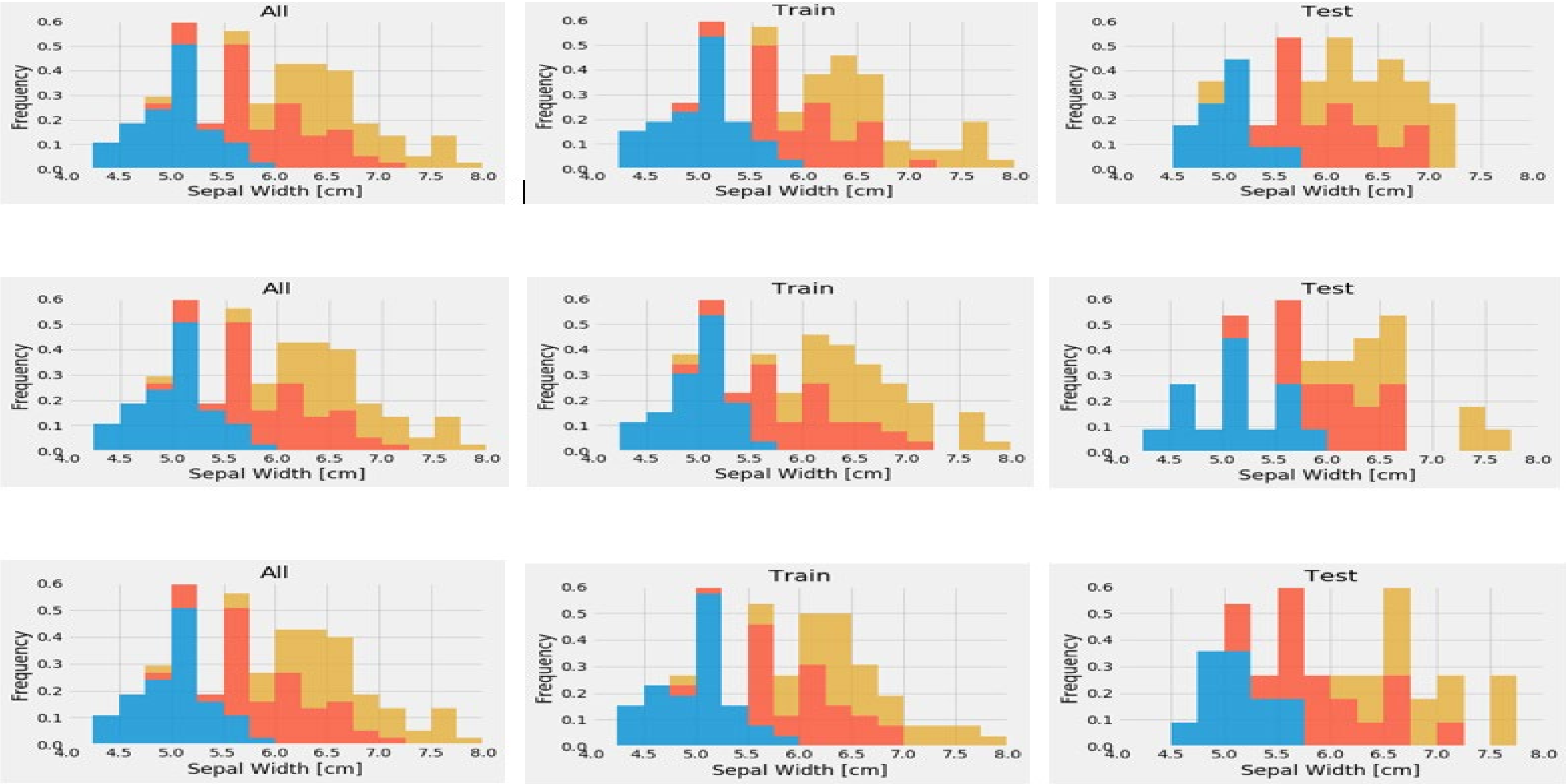
1. Caused by unknown factors. They cannot be modelled



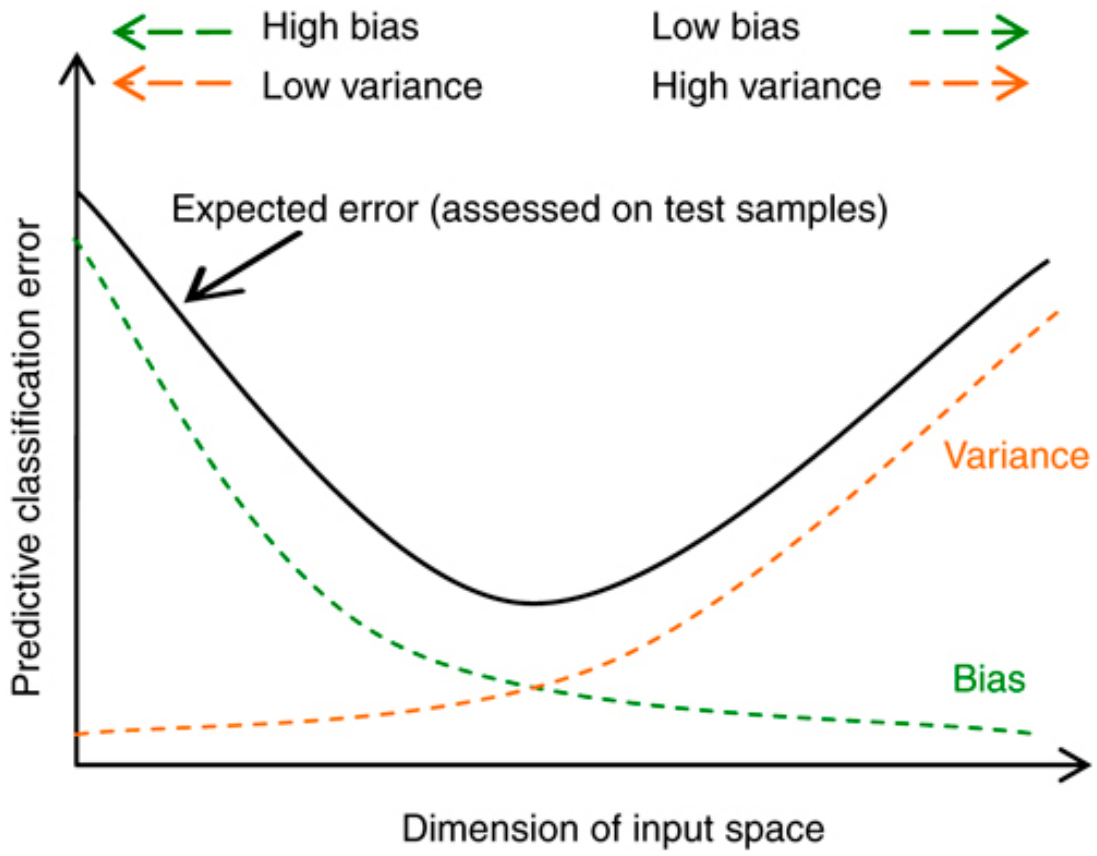
Sample / snapshot

Modelling Errors Visaual demo of variance in training and test data

Sample Data (Analytics Base Table) | **Three Random Training Sets From ABT** | **Three Random Test Sets From ABT**



Modelling Errors



1. We have to find the right attributes and the right number of dimensions such that the total effect of these two (indicated by black curve) minimizes.

2. The gap between variance curve and total error curve reflects presence of random errors in the model

Fitness of a Model

Generalize

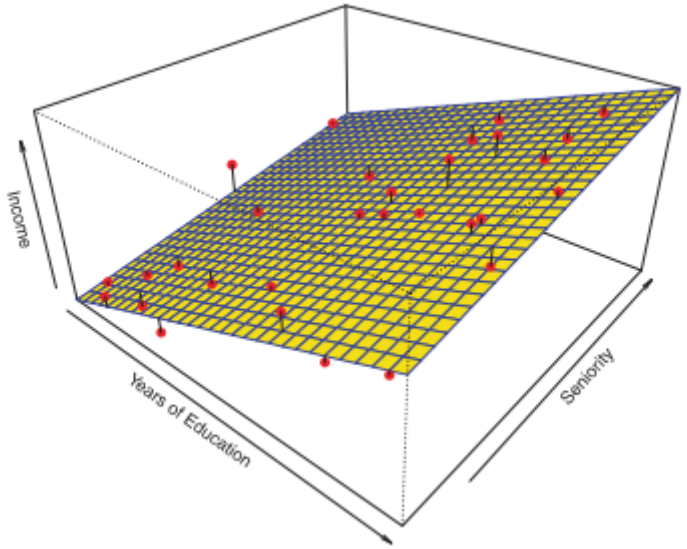
1. Models are expected to perform well (meet least accuracy thresholds) in production (real world data)
2. But data in real world is under flux / jiggle
3. Models have to perform in this context of continuous jiggle. Such models are said to generalize well
4. For models to generalize well, they should neither be underfit or overfit in the training data

Underfit models

1. Models that are over simplified i.e. models in which the independent and dependent attributes interact in a simple linear way (can be expressed in a linear form for e.g. y = mx + c).
2. The model could have been addressed as a quadratic form such as y = m1x + m2 x^2 +C
3. Underfit models result in errors as they fail to capture the complex interactions among the attributes in the real world
4. These models will not generalize in the real world

Overfit models

1. Models that perform very well (sometimes with zero errors) in training data
2. Are complex polynomial surfaces that twist and turn in the feature space to cleanly separate the classes
3. Adjust to the variance in the training data i.e. try to adjust to the positions of the data points though those positions are not the expected values of the data points (mean of the jiggle)
4. These models adapt to the variance error in the data set and will not generalize in the real world



Underfit

In overfit models, the models absorb the noise (variance) in the data points achieving almost 100% accuracy in controlled environment. But when used in production (where the data points have different variance, the models will perform poorly