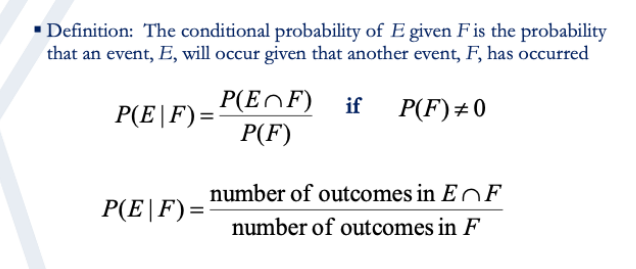
**Module 5 Notes – Naïve Bayes Classification**

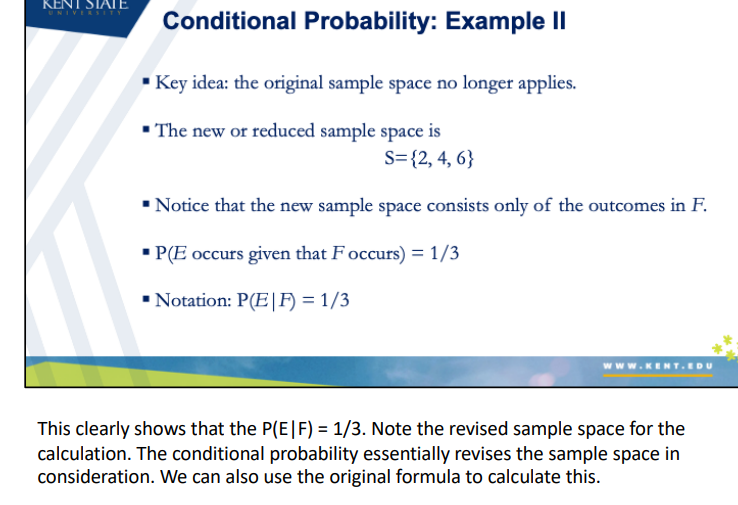
The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods.

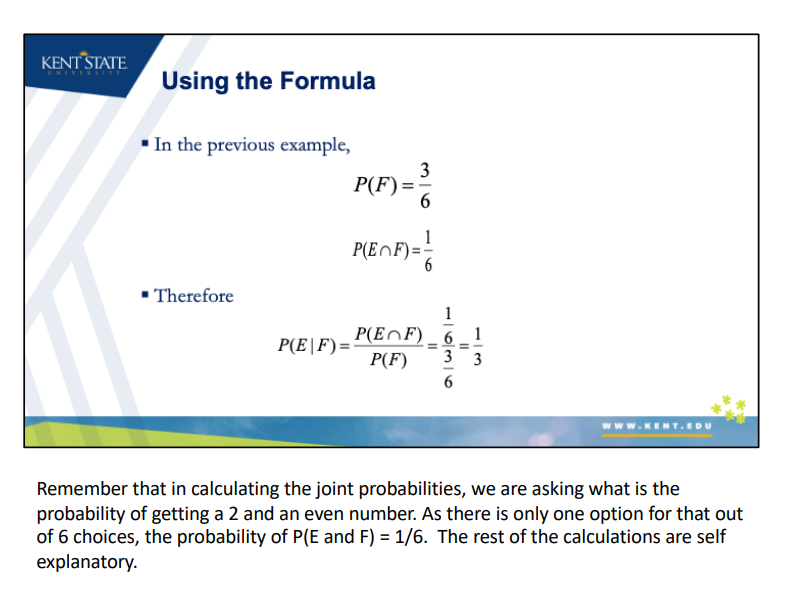
**Review of Probability and Conditional Probability**

* **Conditional Probability**
  + The conditional probability of E given F is the probability that an event, E, will occur given that another event, F, has occurred
  + The joint probability of E and F happening divided by the marginal probability of F

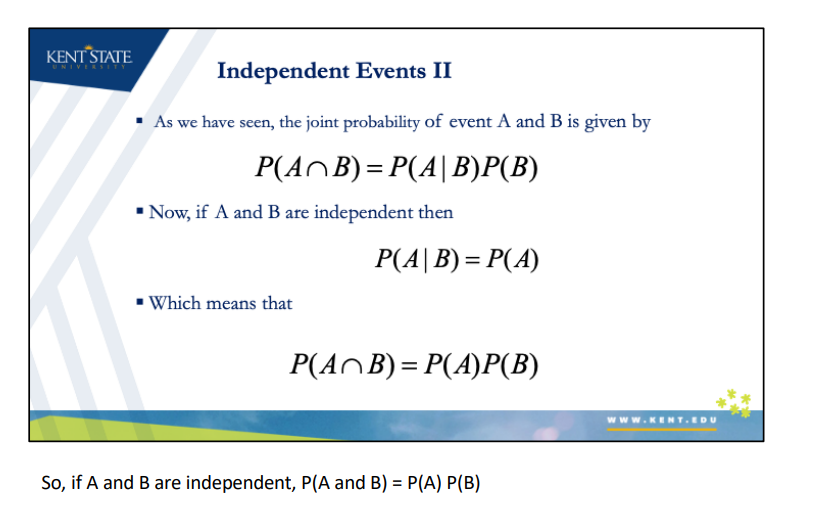


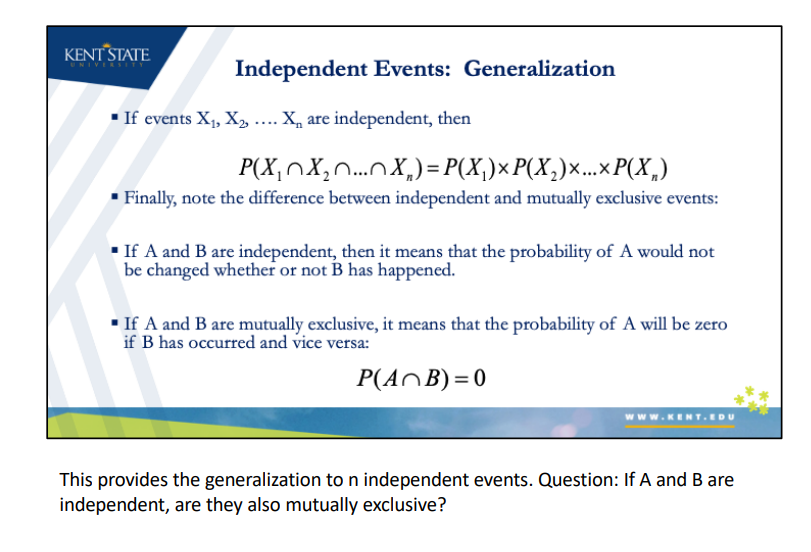
* **Example**
  + Toss a balanced die once and record the number on the top face
  + Let *E* be the event that a 2 shows on the top face
  + Let *F* be the event that the number on the top face is even
  + What is probability of *E,* P*(E)* ?
  + What is the probability of the event *E* if we are told that the number on the top face is even, that is, we know that the event *F* has occurred





**Independent Events**

* If the probability of the occurrence of event A is the same regardless of whether or not an outcome B occurs, then the outcomes A and B are said to be independent of one another
* Formally speaking, if P(A|B) = P(A), then A and B are independent events
* If A and B are independent, the conditional probability of A given B is the same as the probability of A
* **Example – Independent Events**
  + A coin is tossed and a single 6-sided die is rolled. Find the probability of getting a head on the coin and a 3 on the die
  + Probabilities:
    - P(coin is head) = ½
    - P(die is 3) = 1/6
  + The two events are independent and therefore:
    - P(coin is head and die is 3) = ½ \* 1/6 = 1/12



**Naïve Bayes Classifier**

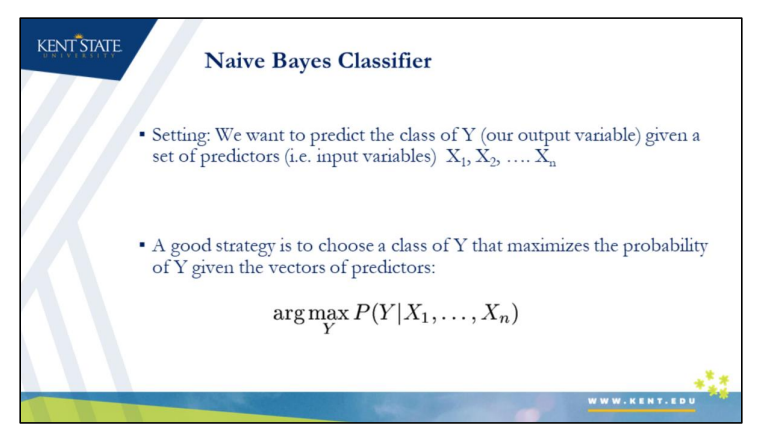
* A branch of Bayesian statistics

**Generative vs. Discriminative Classifiers**

* Training classifiers involves estimating f: X Y, or P(Y|X)
* Discriminative classifiers:
  + Assume some functional form for P(Y|X)
  + Estimate parameters of P(Y|X) directly from training data
  + Ex. logistic regression
* Generative classifiers:
  + Assume some functional form for P(X|Y),P(X)
  + Estimate parameters of P(X|Y),P(X) directly from training data
  + Use Bayes rule to calculate P(Y|X = xi)
* Example: classifying a disease is malignant or benign based on external factors
  + Y label is either benign or malignant
  + X label is the set of predictors (age, gender, etc.)

**Naïve Bayes Classifier**

* We are interested in predicting the class of Y based on the values of a set of predictors. Assign record to the class of Y that maximizes the probability of seeing Y given the values seen for the set X in that observation
  + Find all other records with the same predictor profile (predictor values are the same)
  + Determine the probability that those records belong to the class of interest

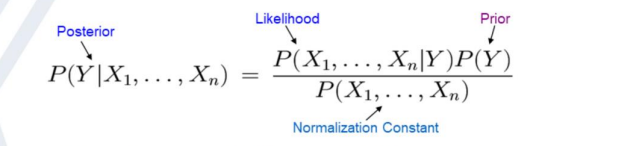


**Example:**

* Assume the input variables are the intensity (brightness) of each of the pixels of an image. The task is to classify a handwritten digit as 5 or 6
* We can calculate the probability of the output being 5 given the input, and also calculate to calculate probability of the output being 6 given the input
* We then compare the probabilities and decide whether the input was a 5 or 6 based on those probabilities – assign the observation to the label with the highest probability
* The object of the classifier is to determine the probability of an observation belonging to a certain class.

**Bayes Classifier**

* Key idea: the original sample space no longer applies
* We use the prior and likelihood probabilities to calculate posterior
* The dominator is not important as it is the same for all class probabilities (i.e. does affect the comparison

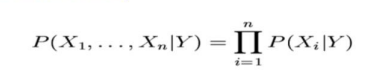


* NB uses conditional probability (probability of A given that event B has occurred)
* Posterior Probability – the probability of the record belonging to class Yi given that its predictor values are x1, x2, etc.
  + P(Yi|x1,…,xn)
  + To classify a record, we compute its probability of belonging to each of its classes in this way, then we classify the record to the class that has the highest probability or use the cutoff probability to decide whether it should be assigned to the class of interest
  + Bayesian classifier works only with categorical predictors.
    - Numerical predictors – unlikely that multiple records will have identical values
    - Numerical predictors must be converted to categorical predictors

**The Naïve Bayes Model**

The problem with explicitly modeling P(X1,…Xn|Y) is that there are usually way too many parameters

* We’ll run out of space, time, and we’ll need tons of training data

The Naïve Bayes Assumption: Assume that **all features are independent** given the class label Y. That is to say

* Finding all records in the sample that exactly like the new record to be classified is not practical for large datasets with many predictors
* NB model makes a key assumption of conditional independence
  + We use the entire dataset

**Steps:**

1. For class Y1, estimate the individual conditional probabilities for each predictor P(xj|Y1) – these are the probabilities that the predictor value in the record to be classified occurs in class Y1. For example, for X1 this probability is estimated by the proportion of x1 values among the Y1 records in the training set
2. Multiply these probabilities by each other, than by the proportion of records belonging to class Y1. This gives the probability in the numerator for the equation on the previous page
3. Repeat Steps 1 and 2 for all the classes. This provides the numerator for all classes Yi in the equation on the previous page. We can further calculate the actual probability by doing the following:
4. Estimate a probability for class Yi by taking the value calculated in Step 2 for class Y1 and dividing it by the sum of such values for all classes

**The Naïve Bayes Model II**

* Naïve Bayes is so called because the independence assumptions we have just made are indeed very naïve for a practical scenario
* Despite this assumption, Naïve Bayes classifier often does surprisingly well and is widely used because in many cases it outperforms more sophisticated classification methods
* Naïve Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets

**Example Applications of Naïve Bayes Classifiers**

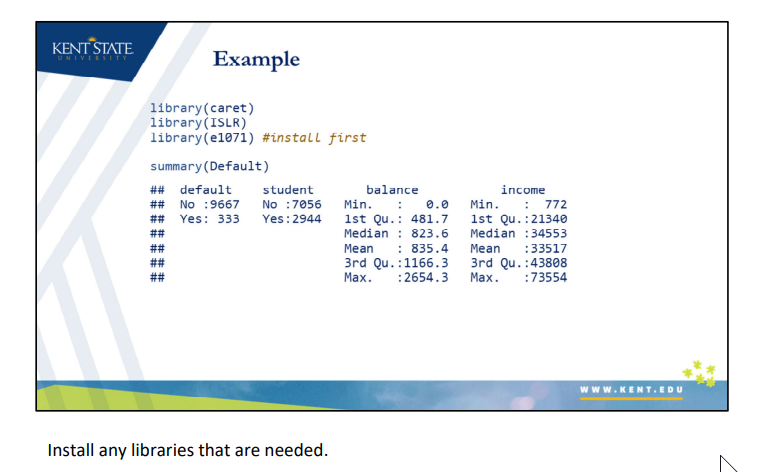
* **Real-time Prediction:** As Naïve Bayes is super fast, it can be used for making predictions in real time
* **Multi-class Prediction**: This algorithm can predict the posterior probability of multiple classes of the target variable
* **Text classification / Spam Filtering / Sentiment Analysis:** Naïve Bayes classifiers are mostly used in text classification (due to their better results in multi-class problems and independence rule)
* Not used for credit scoring

**Naïve Bayes Classifier in R**

* Due to its popularity and wide applications, there are several packages to apply Naïve Bayes (i.e. e1071, klaR, naivebayes, bnclassify)
* We will use the e107 package
* naiveBayes() function can be used to train a model. The function computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule

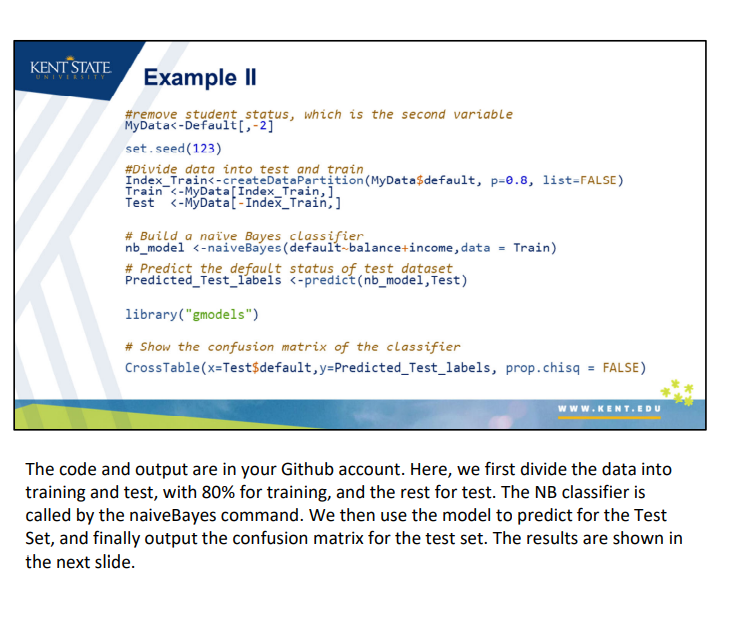
**Example**

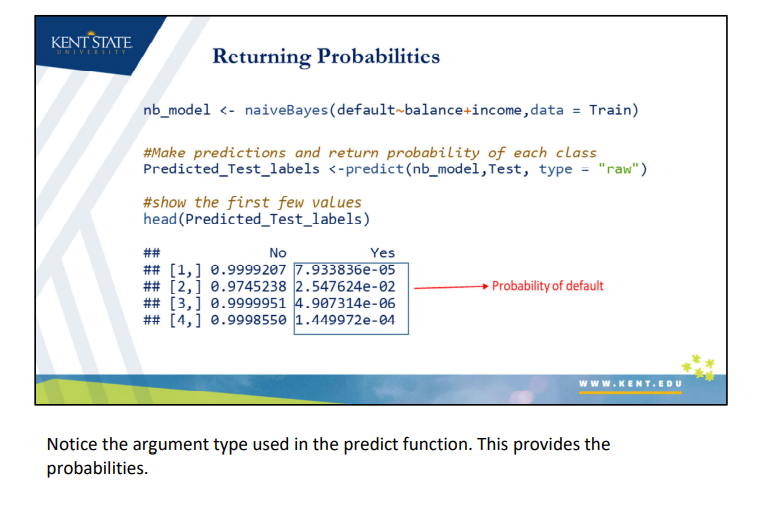
* Let us consider again the credit card default example
* The goal is to create a classification model that can predict credit card defaults based on the balance on the card as well as the card holder income
* A Naïve Bayes classifier will assume that income and credit card balance are two independent variables



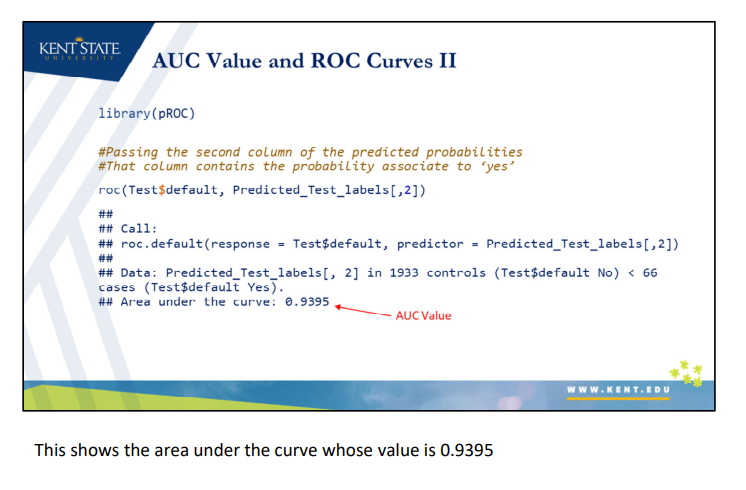
**Probabilities**

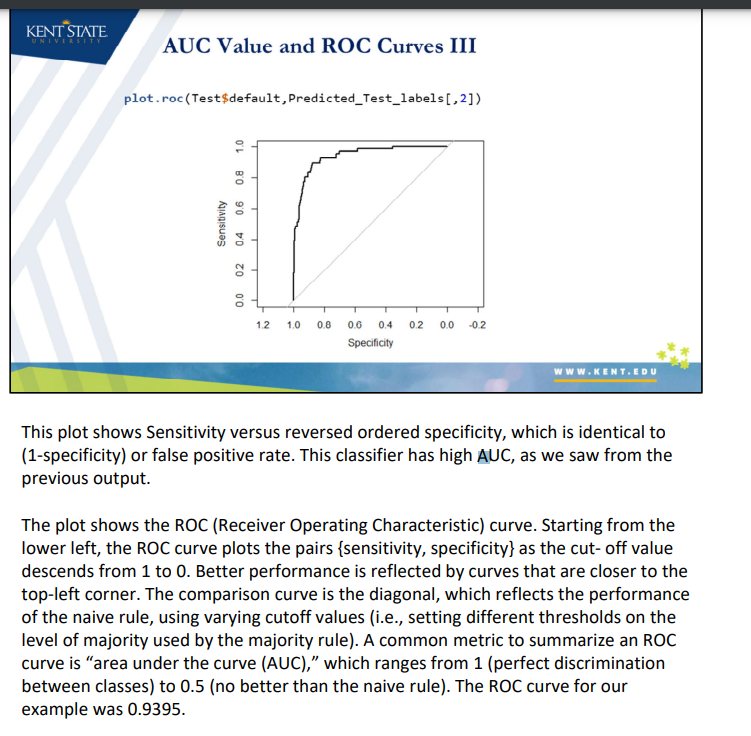
* Sometimes, it is preferred to have raw prediction probabilities rather than predicted class labels
* naiveBayes() can return probabilities if the ‘type’ argument in the predict function is set to ‘raw’
* The default value of ‘type’ argument is ‘class’ which returns the actual predicted class of the input
* The probabilities can be used to compute the AUC of the model





* **AUC Value and ROC Curves**
  + We can use the package ‘pROC’ to compute the Area Under the Curve (AUC) of any classifier (not limited to Naïve Bayes)
  + Roc() is a function to compute the AUC value. The function takes the actual outcomes (i.e. ground truth labels) as the first argument and the predicted probabilities as the second argument and return the AUC
  + Plot.roc(), takes the same inputs and can be used to plot the ROC curve
  + Note that the plot is shown using sensitivity and reversed ordered specificity which are the same as true positive and false positive rates as we saw before
  + A **receiver operating characteristic curve**, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied
  + The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings (i.e. recall/sensitivity)
  + False positive rate is also known as the fall-out or probability of false alarm
    - Can be calculated as (1 – specificity).

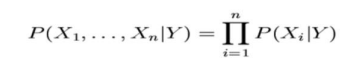




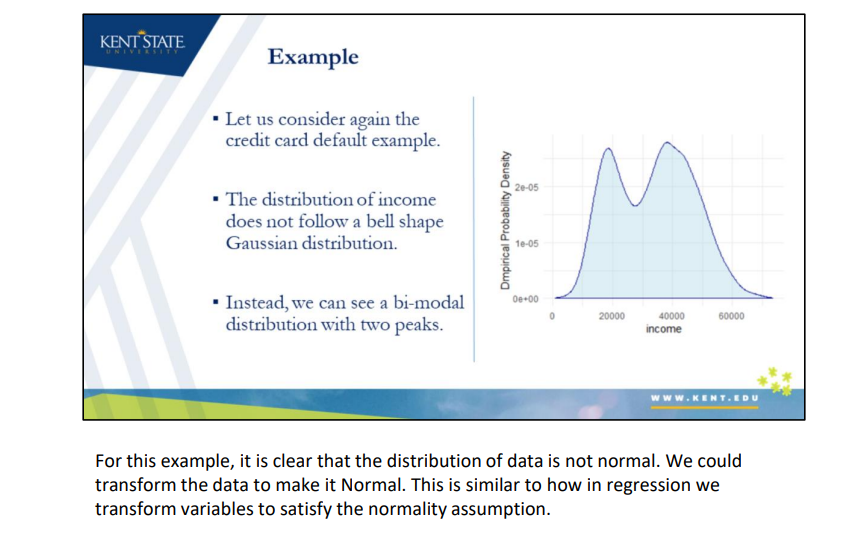
**Bayes Classifier Practical Considerations**

**Assumption of Normality**

* Recall that Naïve Bayes model assumes that all features are independent given the class label Y. That is to say

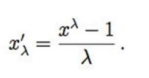


* For categorical variables, this calculation is very straightforward
* However, when dealing with numerical variables, it is often assumed that data follows a Gaussian distribution, so that to make it easy to calculate probabilities
* This assumption, however, may not hold true all the time

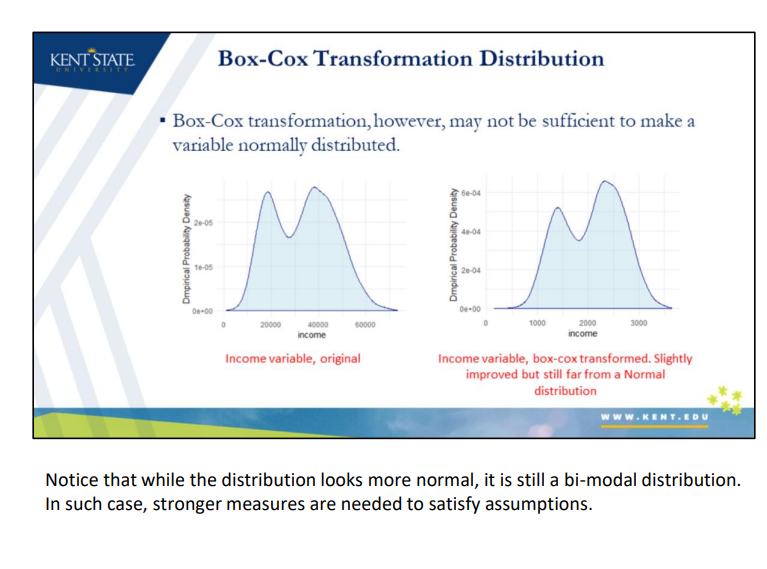


**Box-Cox Transformation**

* Box-Cox transformation can be applied to make a non-normal distribution to look more like a normal distribution
* The Box-Cox transformation of the variable x is also indexed by lambda, and is defined as



* The preProcess() function allows us to determine the transformation that is needed to get non-normal data to look more normal. This transformation is part of the “caret” package



**Kernel Estimator and Laplace Smoother**

* We can also use non-parametric kernel density estimator to try to get a more accurate representation of continuous variable probabilities
* Moreover, since naïve Bayes uses the product of feature probabilities conditioned on each class, we run into a serious problem when new data includes a feature value that never occurs for one or more levels of a response class
* The conditional probability for this feature will be zero and will result the product of all probabilities to be zero
* A solution to this problem involves using the **Laplace smoother**. The Laplace smoother adds a small number to each of the counts in the frequencies for each feature to avoid this issue

**Three difficulties in applying the NB model**

* Requires a very large number of records to obtain good results
* Where a predictor category is not present in the training data, NB assumes that a new record with that category of the predictor has zero probability. This can be a problem if this rare predictor value is important. A popular solution in such cases is to replace zero probabilities with non-zero values using a method called smoothing (e.g., Laplace smoothing can be applied by using argument laplace = 0 in function naiveBayes())
* When the goal is to estimate the probability of class membership (propensity), ranking of records according to their probability can provide biased results (ex. credit scoring)

**Tuning a Naïve Bayes Model**

* We can use caret package to tune a Naïve Bayes Model with respect to the discussed considerations. The following hyperparameters are used:
  + **Usekernel** allows the model to use a kernel density estimate for continuous variables versus a Gaussian density estimate
  + **Adjust** allows the model to adjust the bandwidth of the kernel density (larger numbers mean more flexible density estimate),
  + **fL** allows us to incorporate the Laplace smoother

