1. **If the linear regression coefficient of a predictor is 0.54 then what does it mean?**

ANS : Let us first define the formula for Linear Regression.

Y = B0 + B1\*X1.....

B0 is called the intercept and constant.

Y-> Dependent Variable

X1,X2... are independent variable

If we observe the above formula Regression coefficients represent the change in the response variable for one unit of change in the predictor variable while holding other predictors in the model constant. This statistical definition provides the importance of one predictor variable while holding others constant.

Let us define for above equation.

B0 = 2.3

Let the coeffecient of X1 that is B1=0.54.

Implies Y=2.3+0.54X1.

If we interpret the above derivation 2.3 is constant. For one unit of X1 changes Y changes by 0.54 times X1.

Example : If we define the relation between height and weigth as

W=-123+106H

W-> Weight

H-> Height

It implies for one unit of height increase weight increases by 106H times.

1. **How would you deal a data with Target class imbalance problem?**

Let us first define the imbalance problem.

Imbalance occurs when one class dominates over other.

Let us consider data with

0->-ve(100 data-points)

1->+ve(900 data-points).

We have severly imbalanced data that is class 1 dominates over 0. That is model building on this type of data may be biasing.To resolve this we go for oversampling/downsampling.

Undersampling --> Create new dataset with all the 0 class data and among the all 1 class data, randomly sample 100pts and train your model on this dataset. But there is a problem with undersampling as you are creating new dataset which is extremely less than the previous data. So, your model might not work well because you have thrown away many data which contains much information. You should avoid this situation.

Oversampling --> Create new dataset with all the 1 class data and repeat your minority class 9-times, as you have 900pts from +ve class and 100pts from -ve class.

EX:-

Your original data(100 -ve pts and 900 +ve pts) i.e. the ratio is 1:9 and you split it into 70%(700pts) Train and 30%(100pts) Test randomly then you will have your Train data containing 630 +ve and 70 -ve pts and test data containing 270 +ve and 30 -ve pts. Let, your model classify every pts as -ve on your test data then your accuracy(performence measure/ matrix) will be 90%(270/300).

1. **You have built a classification model with 90% accuracy but your client is not happy because False Positive rate was very high then what will you do?**

ANS : Accuracy is important factor for any model performance. It is the first statistical parameter to be considered for any model accuracy. But this should not be only the score to be considered for model performance. As given in the problem statement accuracy of model is very decent but false positive rate is very high which is not desirable in practical case.

Let us first define what is false positive rate: The false positive rate is defined as the ration between negative events wrongly categorised as positive and total number of actual negative events.

Formula for FPR from confusion matrix is given by (FP/FP+TN).

Confusion matrix :

|  |  |  |
| --- | --- | --- |
|  | Class 1 predicted | Class 2 Predicted |
| Actual class 1 | TP | FN |
| Actual class 2 | FP | TN |

• True Positive (TP) : Observation is positive, and is predicted to be positive.  
• False Negative (FN) : Observation is positive, but is predicted negative.  
• True Negative (TN) : Observation is negative, and is predicted to be negative.  
• False Positive (FP) : Observation is negative, but is predicted positive.

Accuracy and false positive rate are derived from confusion matrix and is error metric that is wrongly classified. This confusion matrix is specific for each model. Each model will have different confusion matrix. So we can go for developing various model using algorithms like logistic regression, Decision Tree, Random Forest, Bagging and Boosting etc. Even ensemble can be used. While developing the models so false positive rate must be lower than the previous mode while keeping the accuracy near to 90%.

1. **Does multicollinearity effects in Naïve Bayes? If yes/no then why?**

Ans : The important assumption of Naive Bayes algorithm is attribute values are conditionally independent given the target value. Here Navie Bayes assumes each independent variable contributes independently.



The above is formula for Naive Bayes algorithm. P is the probability of variable. Navie Bayes fails if variables are correlated or multiolinearity exists in between variables. If multi collinearity exits then probability of each variable is effected.

Let us consider class1 =+ve

Class2=-ve

Here +ve indicates the selection in basketball team and –ve not selected.

and independent variables are Gender, height and Weight.

If we gender and height are correlated then the individual probability will be effected.

Naive bayes works best when independent variables are not dependent.

Individual probability of height weigh and Gender are calculated and applied for target class.

For continuous data it Navie Bayes algorithm assumes data distribution is normal practically which is rare. So mostly Navie bayes applicable for classification problem.

1. **If we do not define number of trees to be built in random forest then how many trees random forest internally creates?**

ANS : Lets come to the conclusion related to number of decision tree in random forest by defining and analyzing the algorithm.

Random forest algorithm is a supervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracyresults.

If we summarize the steps for building random forest model:

1.Let the number of training cases be N, and the number of variables in the classifier be M. ..

2. We are told the number m of input variables to be used to determine the decision at a node of the tree; m should be much less than M.

3. m = sqrt(M)

4. Choose a training set for this tree by choosing n times with replacement from all N available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.

5. For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.

6. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

7. Number of decision trees depend on error rate - Build trees until the error no longer decreases.

8. For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the average vote of all trees is reported as random forest prediction.

If we clearly observe random forest algorithm whether two tree to built or not is dependent on the error rate. If error is constant then the above iteration stops. Therefore there is no actual definition for number of trees build. As from the documentation of R and python the default for each language is 500 and 10 respectively.