Machine Learning

**1.What features would you use to predict the time spent for a restaurant preparing food from the moment an order comes in?**

* The features could be -

Restaurant\_ID,Location, Cuisines, Average\_Cost, Minimum\_Order, Maximum\_Order, Order\_type ,Rating, Votes, Reviews, total\_items, total\_outstanding\_orders,estimated\_order\_place\_duration,min\_item\_price,

max\_item\_price,LOCALITY,book\_table, Food\_ preparing \_time for previous orders.

**2.Can you come up with a scenario in which you would rather under-predict versus over-predict?**

If our algorithm does not perform well with training dataset also not well with test dataset then such scenario called under-predict whereas if our algorithm works well with the training dataset but not well test dataset then such scenario called over-predict.

Consider a scenario ,

We are predicting house price.we have one feature square fit area of a home based of this feature we are trying to build a model that can predict the home price.

When we train our model,we split our dataset into training and test sample.

While we are training our model,the model fits fit all the datapoints and our model ended uo with overfit model.The overfit model tries to fit exactly to the training samples,where Training error close to 0.

But when we givr test data ,the test error become high.

Because the model memorizes the noise of the training data and fails to capture important patterns so it performs poorly for unseen test data.Probably the test dataset error will be 100.

When we dataset when we pick our training samples we pickup randomly.Lets say another person choose different samples for training ,that persons model look different.Here he use same model,same methodology ,so training dataset erroe still 0.

Because both trying to overfit the model.But another person’s testdataset error=27,Because he use another set of train and set data.

Based on selection of training datapoint,the test error varies greatly,this is called High Variance.

We select our training sample randomly so test error varies randomly,which is not good.

This is the common issue with overfit model.

Now,if I use very simple model using linear equation which is underfitting my training sample.Because linear equation can’t truly capture the pattern in training samples.The straight line can’t pass through all the training data points.Here training error is high.suppose we got train dataset error=43 and test dataset error=47.

Now when I select different set of training dataset line will different.And by using this dataset we create model got train dataset error=41,test dataset error=37.Based on whatever datapoints we select training sample,the test error doesn’t varies that much.

But the balanced fit model describe the pattern available in training dataset.Even if we select different set of training data pointd still my model selecting is such that train,test error both are still low.so we prefer balanced fit model.

**3.Analyzing the results of a model, how would you explain the tradeoff between bias and variance?**

When we train our model,we split our dataset into training and test sample.

While we are training our model,the model fits fit all the datapoints and our model ended uo with overfit model.The overfit model tries to fit exactly to the training samples,where Training error close to 0.

But when we give test data ,the test error become high.

Because the model memorizes the noise of the training data and fails to capture important patterns so it performs poorly for unseen test data.Probably the test dataset error will be 100.

When we dataset when we pick our training samples we pickup randomly.Lets say another person choose different samples for training ,that persons model look different.Here he use same model,same methodology ,so training dataset erroe still 0.

Because both trying to overfit the model.But another person’s testdataset error=27,Because he use another set of train and set data.

i.e , Based on selection of training datapoint,the test error varies greatly,this is called High Variance.Because there is a high variability in the test error.

We select our training sample randomly so test error varies randomly,which is not good.

This is the common issue with overfit model.

Now,if I use very simple model using linear equation which is underfitting my training sample.Because linear equation can’t truly capture the pattern in training samples.The straight line can’t pass through all the training data points.Here training error is high.suppose we got train dataset error=43 and test dataset error=47.

Based on whatever datapoints we select for training sample the test error doesn’t varies that mush,this is called Low Variance.

Bias is measurement of how accurately a model can capture a pattern in a training dataset.or we can say the gap in between predicted value by the model and the actual value of the data is called Bias.

How much scattered the predicted values are called variance.

**4. Explain how a Random Forest model actual works under the hood.**

Random forest is a *Supervised Machine Learning Algorithm* that is *used widely in Classification and Regression problems*. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing *continuous variables* as in the case of regression and *categorical variables* as in the case of classification. It performs better results for classification problems.

Random Forest use ensemble technique bagging

also known as *Bootstrap Aggregation* .Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as *row sampling*. This step of row sampling with replacement is called*bootstrap*. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as *aggregation*.

Random Forest use base learnes as Decision tree.Here we not only sample data rows we also sample features.

From dataset D,we have d number of rows and m number of features.

From dataset we will pick some sample of rows and some sample of features.(row sampling with replacement + feature sampling).Then we will give the sample to the decision tree1.Again we pick some row,column will give to decision tree2.Some of rows and feature may repeate for sampling. Like this Individual decision trees are constructed for each sample.All decision tree model will trained.Now when we give test data for the final output we will take majority vote or predictive probability for Classification and *Averaging for* regression.

Decision trees normally suffer from the problem of overfitting if it’s allowed to grow without any control. Random forests are created from subsets of data and the final output is based on average or majority ranking and hence the problem of overfitting is taken care of. Decision tree have two properties – Low Bias,High variance.In Random forest we are using multiple Decision trees.We know that each are every Decision tree will have high variance,but when we combine all Decision tree with respect to mazority vote the high variance will converted into low variance because when we are using row sampling and feature sampling giving records to Decision tree. The Decision tree tents to become an expert with respect to this specific (rows,columns)dataset they have.

Since we are giving different different records of each and every Decision tree.And to convert high variance to low variance we are taking majority vote,not depending on one decision tree output.

**5. How do you know if you have enough data for your model?**

It depends on the type of [machine learning](https://www.datarobot.com/wiki/machine-learning/) problem we want to solve.More is always better.

A typical image classification problem could require tens of thousands of images or more in order to create a classifier.

Sentiment analysis or document [classification](https://www.datarobot.com/wiki/classification/) problems can require thousands of examples due to the sheer number of words and phrases.

For many [regression](https://www.datarobot.com/wiki/regression/) problems, it’s suggested that you have 10x as many observations as you do features. A more general rule of thumb is that the number of observations should be proportional to 1/d^p where p = # of features and d = the maximum spacing between consecutive or neighboring data points after each feature is scaled to the range 0-1.

For [time series](https://www.datarobot.com/blog/ai-simplified-time-series/) problems, you should always have more observations than parameters (we elaborate more on this type of machine learning problem below).

**6. How do you evaluate a model? (F1 score, ROC curve, cross validation, etc…)**

**confusion matrix :**

A confusion matrix is a tool for summarizing the performance of a

classification algorithm. A confusion matrix will give us a clear

picture of classification model performance and the types of errors

produced by the model. It gives us a summary of correct and incorrect

predictions broken down by each category. The summary is

represented in a tabular form.

Four types of outcomes are possible while evaluating a classification

model performance. These four outcomes are described below:-

True Positives (TP) – True Positives occur when we predict an

observation belongs to a certain class and the observation actually

belongs to that class.

True Negatives (TN) – True Negatives occur when we predict an

observation does not belong to a certain class and the observation

actually does not belong to that class.

False Positives (FP) – False Positives occur when we predict an

observation belongs to a certain class but the observation actually

does not belong to that class. This type of error is called Type I

error.

False Negatives (FN) – False Negatives occur when we predict an

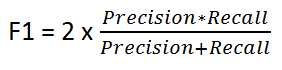
observation does not belong to a certain class but the observation

actually belongs to that class. This is a very serious error and it is

called Type II error.

**F1 score :**

The F-score, also called the F1-score, is a measure of a model’s accuracy on a dataset. It is used to evaluate binary classification systems, which [classify](https://deepai.org/machine-learning-glossary-and-terms/classifier) examples into ‘positive’ or ‘negative’.  F1 which is a function of Precision and Recall.



**ROC curve :**

AUC-ROC curve helps us visualize how well our machine learning classifier is performing.

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the ‘signal’ from the ‘noise’. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives. When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values. This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives. When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting random class or constant class for all the data points.

So, the higher the AUC value for a classifier, the better its ability to distinguish between positive and negative classes.

**Cross-validation :**

Cross-validation is a technique for evaluating ML models by training several ML models on subsets of the available input data and evaluating them on the complementary subset of the data. Use cross-validation to detect overfitting, ie, failing to generalize a pattern.

In Amazon ML, you can use the k-fold cross-validation method to perform cross-validation. In k-fold cross-validation, you split the input data into k subsets of data (also known as folds). You train an ML model on all but one (k-1) of the subsets, and then evaluate the model on the subset that was not used for training. This process is repeated k times, with a different subset reserved for evaluation (and excluded from training) each time.

**Probability**

1.Given uniform distributions X and Y and the mean 0 and standard deviation 1 for both, what’s the probability of 2X > Y?

1/2

2.There are four people in an elevator and four floors in a building. What’s the probability that each person gets off on a different floor?

4!/44would be the probability=3/32

3.What’s the probability that two people get off on the same floor?

9/64

4.Given a deck of cards labeled from 1 to 100, what’s the probability of getting Pick 1 < Pick2 < Pick3?

1/6