**Predictive Modeling**

1. What could be some issues if the distribution of the test data is significantly different than the distribution of the training data?

Example of a situation where data shift could happen:

If you are training a time series model on data that has yearly seasonality but your training window doesn't have enough data to determine the yearly pattern, then a test set, which happens later in the year, will look like it has a distribution different from the training.

Now on to issues:

**What does this mean?** The issue means some level of imbalance is going on to say the least. The level of imbalance could be because of missing class of data or missing data/hyper represented majority class, or skewed features. is an example of this scenario.

**Examples:**

A dataset with imbalanced classes

**Example of skewed features:** I once built a classifier for attractions in a city based on user-generated reviews. I found that the classifier concentrated on features talking about hotels, ignoring other categories. This was because the review lengths were skewed - people prefer to review hotels to to review, say, golf courses. Truncating the hotel reviews - thereby reducing the skew - solved the problem.

Say, one has built a model with such an imbalanced dataset. What problems could happen?

* The model would fail to classify/predict the test data accurately. What it actually does, in such a scenario, is that it predicts/classifies test data depending on the training it had received using the training data at high accuracy score. But in fact it had failed to pick signals from or about:

The underrepresented class

Missing trend in the training data

Examples: credit fraud,

* Resources will be wasted in tuning the uninteresting or inefficient parameters.
* Model fails to understand the business context and therefore fails to generalize in reality.

Possible solutions:

At feature level:

* Oversampling the weak class (SMOTE)
* Undersampling the class that’s over represented (Tomek Links)
* Could try to add more data that are representative of the missing trend or class if possible.

At model level:

* Picking models that already involve mechanisms to counter imbalanced classes. For example: tree based models like Decision Tree, Random Forest, and Ada Boost are robust and deal with imbalanced classes weights.
* Use **Penalized Models**. Like penalized-SVM and penalized-LDA. They put additional cost on the model for making classification mistakes on the minority class during training. These penalties can bias the model towards paying attention to minority class. This is a way to create cost sensitive models – meaning a way to account for misclassification cost)
* Hellinger distance decision trees are gaining some bus for working well on unbalanced data

At algorithmic level:

* Ensemble learning
* Assigning more weights to weaker class in a way to promote them
* Boosting (gradient or adaptive) can work well

At metrics level

* The Area Under the ROC curve (AUC)
* The [F1 Score](https://en.wikipedia.org/wiki/F1_score) is the harmonic mean of precision and recall
* [Cohen’s Kappa](https://en.wikipedia.org/wiki/Cohen%27s_kappa) is an evaluation statistic

2. Have to figure out how to deal with a situation of balanced classes but skewed features.

3. If we oversample or downsample the training set and test on the original dataset, do we need to make some adjustments to the probabilities or threshold to account for the difference in distribution?

4. What are some ways one can make their model more robust to outliers?

Here are some changes that one can make to their model:

* **Use a model that's resistant to outliers.** Tree-based models are generally not as affected by outliers, while regression-based models are. If you're performing a statistical test, try a non-parametric test instead of a parametric one.

What arenon-parametric test?

* **Use a more robust error metric**. Switching from mean squared error to mean absolute difference (or something like Huber Loss) reduces the influence of outliers.

Robust estimators minimize the sum of the absolute values of the errors instead of the sum of squares.  This makes them more resistant to outliers.

Here are some changes that one can make to their data:

* **Winsorize your data.**Artificially cap your data at some threshold. See [What are some applications of winsorization?](https://www.quora.com/What-are-some-applications-of-winsorization)
* **Transform your data**. If your data has a very pronounced right tail, try a log transformation.
* **Remove the outliers**. This works if there are very few of them and you're fairly certain they're anomalies and not worth predicting.

Other suggestions:

Univariate method: This method looks for data points with extreme values on one variable.

Multivariate method: Here we look for unusual combinations on all the variables.

Minkowski error: This method reduces the contribution of potential outliers in the training process.

5. Why is the median a measure of central tendency? It doesn't have anything to do with any other values of the data set, so how does it "describe" the data set?

The median is the point that minimizes the expected distance to any data point. Mean, Median, and Mode are all measures of central tendency that minimize some sort of distance metric.  
  
The mean minimizes the L2 norm (squared distance), the median minimizes the L1 norm (distance), and the mode minimizes the L0 norm (0-1 loss).

When the distribution you're looking at is symmetrical, the mean and median will be the same and both of them will capture an intuitive notion of "middle". However, when the distribution is skewed, the the mean and the median no longer are the same data point! The mean is the default measure of central tendency, but one main problem is that it can be overly influenced by outliers. This is why distributions like household income or average house value is usually summarized by the median rather than the mean.  
  
The mean household income in 2004 was 60.5k, while the median household income was 43.3k. The median household income is more representative of an intuitive notion of the "middle" house, whereas the mean income is much higher than the "middle" household since it includes the incomes of people making 7 figures and more.

6. What are some differences you would expect in a model that minimizes squared error, versus a model that minimizes absolute error? In which cases would each error metric be appropriate?

**Minimizing the squared error (**L2L2**) over a set of numbers results in finding its mean, and minimizing the absolute error (**L1L1**) results in finding its median.** (And minimizing the L0L0 error results in finding the modes.)

**Using the squared error is easier to solve for and using the absolute error is more robust to outliers.**

The reason that the squared error is easier to solve for is that the derivatives are continuous. The RMSE (root mean squared error) is more appropriate to represent model performance than the MAE when the error distribution is expected to be Gaussian.

It depends what you want. If you think that outliers are merely corrupted data that should be somewhat ignored, then absolute error might be better to use. If you want to avoid very large errors and still fit outliers somewhat reasonably, then squared error might be better to use.

7. What error metric would you use to evaluate how good a binary classifier is? What if the classes are imbalanced? What if there are more than 2 groups?

What metric should one use really depends on the type of problem they are working on, problem domain, and business context. In most cases we are worried about how well a model predicts with respect to each class and so we might better benefit adapting specific metrics like precision, recall, f1 score, and ROC in evaluating the model’s performance.

Error metrics:

ROC

F1 score

Precision

Recall

Accuracy

Matthews Correlation Coefficient (MCC)

Specificity

Sensitivity/Recall

Accuracy as a metric can be misleading in measuring a model’s performance, especially, in case of imbalanced classes. Why?

For example, in a problem where there is a large class imbalance, a model can predict the value of the majority class for all predictions and achieve a high classification accuracy, the problem is that this model is not useful in the problem domain. As we saw in our breast cancer example.

This is called the **Accuracy Paradox**. For problems like, this additional measures are required to evaluate a classifier.

Metrics to use for multiclass classification problem:

Typical metrics used in multiclass are the same as the metrics used in the binary classification case. The metric is calculated for each class by treating it as a binary classification problem after grouping all the other classes as belonging to the second class. Then the binary metric is averaged over all the classes to get either a macro average (treat each class equally) or weighted average (weighted by class frequency) metric.

* Macro average at the level of algorithmic setting.
* F1 score could be used too
* It is useful to review the *confusion matrix* for multiclass problems. The confusion matrix is a table that shows each class in the evaluation data and the number or percentage of correct predictions and incorrect predictions.

6. What are various ways to predict a binary response variable?

Can you compare two of them and tell me when one would be more appropriate? What’s the difference between these? (SVM, Logistic Regression, Naive Bayes,

Decision Tree, etc.)

Various ways to predict a binary response variable?

Mostly, y-value that measures a binary outcome: 0 or 1; yes or no; True or False is called a binary response variable.

Regression

Logistic Regression

Classification

Random forest

Naïve Bayes

Decision Tree

Ada Boost

Xg Boost

SVM

7. What is regularization and where might it be helpful? What is an example of using regularization in a model?

8. Why do we need to do Laplace Smoothing in Naive Bayes?

If a word has never before appeared in a document of a certain class, the probability will be 0. Since we are multiplying the probability (argmaxtopic P(topic|document)=argmaxtopic P(topic) summationword P(word|topic)wordcount ), the whole probability becomes 0 We basically lose all information. By using Laplace Smoothing we are assuming that every word has been seen one extra time. This will guarantee that no value will be 0.

9. Suppose N = 100 represents a dense sample for a three dimensional feature space. To achieve same density in an eight dimensional feature space, how many points would we need?

Answer: 1008 data points – a lot more than what’s required in three dimensional feature space.

10. The first step in the K-means algorithm involves randomly assigning data points to clusters, and as such, only finds local minimums. How do we typically deal with this?

By iterating over all possible clustering options.

11. Describe the process of varying K in K-means. Contrast this with the process of varying K in the hierarchical clustering setting.

There’s no need to set K-number of clusters in Hierarchical clustering. Depending on which distance strategy is chosen to link or merge the data points into clusters, the first cluster is formed. Hierarchical clustering is different from K-means in that it clusters the clusters and keep iterating until every cluster (and the data points within those clusters) is clustered under one main cluster.

Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively. This hierarchy of clusters is represented as a tree (or dendrogram). The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample. The Agglomerative Clustering object performs a hierarchical clustering using a bottom up approach: each observation starts in its own cluster, and clusters are successively merged together. The linkage criteria determines the metric used for the merge strategy:

**Ward** minimizes the sum of squared differences within all clusters. It is a variance-minimizing approach and in this sense is similar to the k-means objective function but tackled with an agglomerative hierarchical approach.

**Maximum or complete** linkage minimizes the maximum distance between observations of pairs of clusters.

**Average** linkage minimizes the average of the distances between all observations of pairs of clusters.

* **Single linkage** clusters looks at all the pairwise distances between the items in the two clusters and takes the distance between the clusters as the minimum distance.
* **Complete linkage**, which is more popular, takes the maximum distance.
* **Average linkage** takes the average, which as it turns out is fairly similar to complete linkage. More commonly used and less affected by outliers.
* **Centroid linkage** sounds the same as average linkage but instead of using the average distance, it creates a new item, which is the average of all the individual items and then uses the distance between averages. Popularly used in genomics.

12. Curse of Dimensionality

In data mining problems, data samples often have a large number of features or attributes. If each sample can be represented as a vector in a vector space, then the vector is defined in a very high dimensional space. This high dimensional data set causes problems known as the ”curse of dimensionality”. The size of a data set in an n-dimensional space increases exponentially with dimensions. Because if a one-dimensional sample containing m data points has a satisfactory level of density, then to achieve the same density of points in n dimensions, we need mn data points. For example, if 100 data points gives sufficiently dense samples in one dimension, then to obtain the same density in a sample space of 10 dimensions, we will need

10010 = 1020 data samples! The curse of dimensionality describes a scenario in which you can't possibly obtain an exponential number of training points in high dimensional space.

As the dimensionality, n, increases, so does the radius needed to enclose a given fraction of the data points in an n-dimensional space. Suppose all the data points in an n-dimensional space lies within the n-dimensional hypercube. For a given fraction of data points, p, the edge length, en(p), of a hypercube enclosing that fraction of data points is given by

En(p) = p(1/n). For example if one wishes to enclose 10% of the sample (p =

0.1), the corresponding edge for a two-dimensional space will be e2 (0.1) = 0.32, for a three-dimensional space e3 (0.1) = 0.46, for a 10-dimensional space

e10 (0.1) = 0.80, and for a 15-dimensional space e15 (0.1) = 0.96.

This shows that a large neighborhood is required to capture even a small portion of the data in a high-dimensional space.

13. What’s an EM algorithm?

In statistics, an expectation–maximization (EM) algorithmis an iterative method to find maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables.

14. Which machine algorithms require data scaling/normalization?

All algorithms that are distance based require scaling. This includes all curve fitting algorithms (linear/non-linear regressions), logistic regression, KNN, SVM, Neural Networks, clustering algorithms like k-means clustering etc.

Algorithms that are used for matrix factorization, decomposition or dimensionality reduction like PCA, SVD, Factorization Machines etc. also require normalization.

Algorithms that do not require normalization/scaling are the ones that rely on rules. They would not be affected by any monotonic transformations of the variables. Scaling is a monotonic transformation.

Examples of algorithms in this category are all the tree-based algorithms - CART, Random Forests, Gradient Boosted Decision Trees etc. These algorithms utilize rules (series of inequalities) and do not require normalization.

Also, Algorithms that rely on distributions of the variables, like Naive Bayes also do not need scaling.