1. What is the definition of a target function? In the sense of a real-life example, express the target function. How is a target function's fitness assessed?

Answer: A target function, in machine learning, is a method for solving a problem that an AI algorithm parses its training data to find. Once an algorithm finds its target function, that function can be used to predict results (predictive analysis). The function can then be used to find output data related to inputs for real problems where, unlike training sets, outputs are not included.

The target function is essentially the formula that an algorithm feeds data to in order to calculate predictions. As in algebra, it is common when training AI to find the variable from the solution, working in reverse. The function as defined by f is applied to the input (I) to produce the output (I), Therefore O= f(I).

The target variable will vary depending on the business goal and available data. For example, let’s say you want to use sentiment analysis to classify whether tweets about your company’s brand are positive or negative. Some aspects of a tweet that can be useful as features are word tokens, parts of speech, and emoticons. A model cannot learn how those features relate to sentiment without first being given examples of which tweets are positive or negative (the target). Targets are often manually labeled in a dataset, but there are ways to automate this process.

Without a labeled target, supervised machine learning algorithms would be unable to map available data to outcomes, just as a child would be incapable of figuring out that cats are called “cats” without having been told so at least a few times. It is important to have a well-defined target since the only thing an algorithm does is learn a function that maps relationships between input data and the target. The model’s outcomes will be meaningless if your target doesn’t make sense.

2. What are predictive models, and how do they work? What are descriptive types, and how do you use them? Examples of both types of models should be provided. Distinguish between these two forms of models.

Answer:

Predictive Analytics will help an organization to know what might happen next, it predicts future based on present data available. It will analyze the data and provide statements that have not happened yet. It makes all kinds of predictions that you want to know and all predictions are probabilistic in nature.

Examples: Sentimental Analysis, Credit score analysis, forecasts reports for company etc.

Descriptive Analytics will help an organization to know what has happened in the past, it would give you the past analytics using the data that are stored. For a company, it is necessary to know the past events that help them to make decisions based on the statistics using historical data. For example, you might want to know how much money you lost due to fraud and many more.

Examples: Sales Report, Revenue of the company, Performance analysis etc.

|  |  |  |
| --- | --- | --- |
| Comparison | Descriptive Analytics | Predictive Analytics |
| Describes | What happened in the past? By using the stored data. | What might happen in the future? By using the past data and analyzing it. |
| Process Involved | Involves Data Aggregation and Data Mining. | Involves Statistics and forecast techniques. |
| Definition | The process of finding useful and important information by analyzing the huge data. | This process involves in forecasting the future of the company, which are very useful. |
| Data Volume | It involves in processing huge data that are stored in data warehouses. Limited to past data. | It involves analyzing large past data and then predicts the future using advance techniques. |
| Examples | Sales report, revenue of a company, performance analysis, etc. | Sentimental analysis, credit score analysis, forecast reports for a company, etc. |
| Accuracy | It provides accurate data in the reports using past data. | Results are not accurate, it will not tell you exactly what will happen but it will tell you what might happen in the future. |
| Approach | It allows the reactive approach | While this a proactive approach |

3. Describe the method of assessing a classification model's efficiency in detail. Describe the various measurement parameters.

Answer: Classification performance metrics such as Log-Loss, Accuracy, AUC(Area under Curve) etc.

The metrics that you choose to evaluate your machine learning model is very important. Choice of metrics influences how the performance of machine learning algorithms is measured and compared.

1. Confusion Matrix is one of the most intuitive and easiest (unless of course, you are not confused) metrics used for finding the correctness and accuracy of the model. It is used for Classification problem where the output can be of two or more types of classes.
2. Accuracy in classification problems is the number of correct predictions made by the model over all kinds predictions made. Accuracy is a good measure when the target variable classes in the data are nearly balanced.
3. Precision is a measure that tells us what proportion of patients that we diagnosed as having cancer, actually had cancer. The predicted positives (People predicted as cancerous are TP and FP) and the people actually having a cancer are TP.
4. Recall or Sensitivity: Recall is a measure that tells us what proportion of patients that actually had cancer was diagnosed by the algorithm as having cancer. The actual positives (People having cancer are TP and FN) and the people diagnosed by the model having a cancer are TP. (Note: FN is included because the Person actually had a cancer even though the model predicted otherwise).
5. Specificity: Specificity is a measure that tells us what proportion of patients that did NOT have cancer, were predicted by the model as non-cancerous. The actual negatives (People actually NOT having cancer are FP and TN) and the people diagnosed by us not having cancer are TN. (Note: FP is included because the Person did NOT actually have cancer even though the model predicted otherwise). Specificity is the exact opposite of Recall.
6. F1 Score: We don’t really want to carry both Precision and Recall in our pockets every time we make a model for solving a classification problem. So it’s best if we can get a single score that kind of represents both Precision(P) and Recall(R). One way to do that is simply taking their arithmetic mean. i.e (P + R) / 2 where P is Precision and R is Recall. But that’s pretty bad in some situations.

4.

i. In the sense of machine learning models, what is underfitting? What is the most common reason for underfitting?

ii. What does it mean to overfit? When is it going to happen?

iii. In the sense of model fitting, explain the bias-variance trade-off.

Answer:

1. Underfitting: A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data. (It’s just like trying to fit undersized pants!) Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough. It usually happens when we have less data to build an accurate model and also when we try to build a linear model with fewer non-linear data. In such cases, the rules of the machine learning model are too easy and flexible to be applied on such minimal data and therefore the model will probably make a lot of wrong predictions. Underfitting can be avoided by using more data and also reducing the features by feature selection. In a nutshell, Underfitting – High bias and low variance
2. Overfitting: A statistical model is said to be overfitted when we train it with a lot of data (just like fitting ourselves in oversized pants!). When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. Then the model does not categorize the data correctly, because of too many details and noise. The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees. In a nutshell, Overfitting – High variance and low bias.
3. Bias: Assumptions made by a model to make a function easier to learn.

Variance: If you train your data on training data and obtain a very low error, upon changing the data and then training the same previous model you experience a high error, this is variance.

5. Is it possible to boost the efficiency of a learning model? If so, please clarify how.

Answer: Methods to Boost the Accuracy of a Model:

1. Add more data: Having more data is always a good idea. It allows the “data to tell for itself,” instead of relying on assumptions and weak correlations. Presence of more data results in better and accurate models.
2. Treat missing and Outlier values: The unwanted presence of missing and outlier values in the training data often reduces the accuracy of a model or leads to a biased model. It leads to inaccurate predictions. This is because we don’t analyse the behaviour and relationship with other variables correctly. So, it is important to treat missing and outlier values well.
   1. Missing: In case of continuous variables, you can impute the missing values with mean, median, mode. For categorical variables, you can treat variables as a separate class. You can also build a model to predict the missing values.
   2. Outlier: You can delete the observations, perform transformation, binning, Imputation (Same as missing values) or you can also treat outlier values separately.
3. Feature Engineering: This step helps to extract more information from existing data. New information is extracted in terms of new features. These features may have a higher ability to explain the variance in the training data. Thus, giving improved model accuracy.

Feature engineering is highly influenced by hypotheses generation. Good hypothesis result in good features. Always suggestted to invest quality time in hypothesis generation. Feature engineering process can be divided into two steps:

* 1. Feature transformation: There are various scenarios where feature transformation is required:

1. Changing the scale of a variable from original scale to scale between zero and one. This is known as data normalization.
2. Some algorithms works well with normally distributed data. Therefore, we must remove skewness of variable(s). There are methods like log, square root or inverse of the values to remove skewness.
3. Some times, creating bins of numeric data works well, since it handles the outlier values also. Numeric data can be made discrete by grouping values into bins. This is known as data discretization.
   1. Feature Creation: Deriving new variable(s ) from existing variables is known as feature creation. It helps to unleash the hidden relationship of a data set. Let’s say, we want to predict the number of transactions in a store based on transaction dates. Here transaction dates may not have direct correlation with number of transaction, but if we look at the day of a week, it may have a higher correlation. In this case, the information about day of a week is hidden. We need to extract it to make the model better.
4. Feature Selection: Feature Selection is a process of finding out the best subset of attributes which better explains the relationship of independent variables with target variable.
   1. Domain Knowledge: Based on domain experience, we select feature(s) which may have higher impact on target variable.
   2. Visualization: As the name suggests, it helps to visualize the relationship between variables, which makes your variable selection process easier.
   3. Statistical Parameters: We also consider the p-values, information values and other statistical metrics to select right features.
      1. PCA: It helps to represent training data into lower dimensional spaces, but still characterize the inherent relationships in the data. It is a type of dimensionality reduction technique. There are various methods to reduce the dimensions (features) of training data like factor analysis, low variance, higher correlation, backward/ forward feature selection and others.
5. Multiple algorithms: Hitting at the right machine learning algorithm is the ideal approach to achieve higher accuracy. But, it is easier said than done. This intuition comes with experience and incessant practice. Some algorithms are better suited to a particular type of data sets than others. Hence, we should apply all relevant models and check the performance
6. Algorithm Tuning: We know that machine learning algorithms are driven by parameters. These parameters majorly influence the outcome of learning process. The objective of parameter tuning is to find the optimum value for each parameter to improve the accuracy of the model. To tune these parameters, you must have a good understanding of these meaning and their individual impact on model. You can repeat this process with a number of well performing models.
7. Ensemble methods: This is the most common approach found majorly in winning solutions of Data science competitions. This technique simply combines the result of multiple weak models and produce better results. This can be achieved through many ways: Bagging (Bootstrap Aggregating) and Boosting. It is always a better idea to apply ensemble methods to improve the accuracy of your model. There are two good reasons for this: a ) They are generally more complex than traditional methods. b) The traditional methods give you a good base level from which you can improve and draw from to create your ensembles.
8. Cross Validation: To find the right answer of this question, we must use cross validation technique. Cross Validation is one of the most important concepts in data modeling. It says, try to leave a sample on which you do not train the model and test the model on this sample before finalizing the model. This method helps us to achieve more generalized relationships.

6. How would you rate an unsupervised learning model's success? What are the most common success indicators for an unsupervised learning model?

Answer: In case of unsupervised learning, the process is not very straight forward as we do not have the ground truth. In the absence of labels, it is very difficult to identify KPIs which can be used to validate results.

There are two classes of statistical techniques to validate results for cluster learning. These are:

* Internal validation
* External validation

Most of the literature related to internal validation for cluster learning revolves around the following two types of metrics –

* Cohesion within each cluster
* Separation between different clusters

Twin-Sample Validation: This validates the results of our unsupervised learning model in the absence of true cluster labels. This step takes it as a given that we have already performed clustering on our training data and now want to validate the results. The approach consists of following four steps:

* Creating a twin-sample of training data
* Performing unsupervised learning on twin-sample
* Importing results for twin-sample from training set
* Calculating similarity between two sets of results

7. Is it possible to use a classification model for numerical data or a regression model for categorical data with a classification model? Explain your answer.

Answer: For numerical data, choices are too many - starting from basic decision trees, naive bayes, SVM, logistic regression, ensemble methods (bagging, boosting), Random forest, multi-layer perceptron etc.

For categorical data - naive bayes, decision trees and their ensembles including Random forest, Minimum distance classifiers or KNN type with a cost function different than euclidean distance e.g. hamming distance

8. Describe the predictive modeling method for numerical values. What distinguishes it from categorical predictive modeling?

Answer:

1. A categorical variable has too many levels. This pulls down performance level of the model. For example, a cat. variable “zip code” would have numerous levels.
2. A categorical variable has levels which rarely occur. Many of these levels have minimal chance of making a real impact on model fit. For example, a variable ‘disease’ might have some levels which would rarely occur.
3. There is one level which always occurs i.e. for most of the observations in data set there is only one level. Variables with such levels fail to make a positive impact on model performance due to very low variation.
4. If the categorical variable is masked, it becomes a laborious task to decipher its meaning. Such situations are commonly found in data science competitions.
5. You can’t fit categorical variables into a regression equation in their raw form. They must be treated.
6. Most of the algorithms (or ML libraries) produce better result with numerical variable. In python, library “sklearn” requires features in numerical arrays. Look at the below snapshot. I have applied random forest using sklearn library on titanic data set (only two features sex and pclass are taken as independent variables). It has returned an error because feature “sex” is categorical and has not been converted to numerical form.

9. The following data were collected when using a classification model to predict the malignancy of a group of patients' tumors:

i. Accurate estimates – 15 cancerous, 75 benign

ii. Wrong predictions – 3 cancerous, 7 benign

Determine the model's error rate, Kappa value, sensitivity, precision, and F-measure.

Answer:

Cancerous = -ve, benign = +ve

True Positive = 75

True Negative = 15

False Positive = 7

False Negative = 3

Accuracy = (TP+TN)/(TP+FP+FN+TN) = (75+15)/(75+7+3+15) = 0.90

Sensitivity or Recall = TP/(TP+FN) = 75/(75+3) = 0.96

Precision = TP/(TP+FP) = 75/(75+7) = 0.91

F-measure or F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

= 2\*(0.96 \* 0.91) / (0.96 + 0.91) = 1.7472 / 1.83 = 0.95

Kappa = (total accuracy – random accuracy) / (1- random accuracy) = (90 – 75) / (1-75)

model's error rate = (FP+FN)/(P+N) = (7+3)/(82+18) = 0.1

10. Make quick notes on:

1. The process of holding out

2. Cross-validation by tenfold

3. Adjusting the parameters

Answer:

1. Hold-out is when you split up your dataset into a 'train' and 'test' set. The training set is what the model is trained on, and the test set is used to see how well that model performs on unseen data.
2. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation. Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data.
3. Model parameters are estimated from data automatically and model hyperparameters are set manually and are used in processes to help estimate model parameters. Model hyperparameters are often referred to as parameters because they are the parts of the machine learning that must be set manually and tuned.

11. Define the following terms:

1. Purity vs. Silhouette width

2. Boosting vs. Bagging

3. The eager learner vs. the lazy learner

Answer:

1. Within the context of cluster analysis, Purity is an external evaluation criterion of cluster quality. It is the percent of the total number of objects(data points) that were classified correctly, in the unit range [0..1]. The Average Silhouette Width (ASW) is a popular cluster validation index to estimate the number of clusters.
2. Bagging is a way to decrease the variance in the prediction by generating additional data for training from dataset using combinations with repetitions to produce multi-sets of the original data. Boosting is an iterative technique which adjusts the weight of an observation based on the last classification.
3. Lazy learning methods simply store the data and generalizing beyond these data is postponed until an explicit request is made. Eager learning methods use the same approximation to the target function, which must be learned based on training examples and before input queries are observed.