**Predictive Analytics through Machine Learning-an Overview**

Pradeep Bilurkar

**Abstract:**

This review article covers basic concepts of Descriptive Analysis, Predictive Analytics and Prescriptive Analytics. While discussing Predictive Analytics, emphasis is given more on Machine learning techniques that are used in Predictive analytics. The challenges generally faced during machine learning applications/projects and their possible solutions are discussed in detail.

# Introduction:

Data Analytics is an analytic process designed to explore data (usually large amounts of data - typically business or market related - also known as "big data") in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. According to Gartner “analytics” **[1]** is used to describe statistical and mathematical data analysis that clusters, segments, scores and predicts what scenarios are most likely to happen.

The three dominant types of analytics are:

# Descriptive Analytics:

Descriptive analytics can be classified into three areas that answer certain kinds of questions:

* **Standard reporting and dashboards:** What happened? What is happening now?
* **Ad-hoc reporting:** How many? How often? Where?
* **Analysis/query/drill-down:** What exactly is the problem? Why is it happening?

Descriptive analytics are the most commonly used and most well understood type of analytics. Descriptive analytics categorizes, characterizes, consolidates and classifies data. Descriptive analytics includes dashboards, reports (e.g., budget, sales, revenue and costs) and various types of queries. Tools for descriptive analytics may provide mechanisms for interfacing to enterprise data sources. They typically include report generation, distribution capability and data visualization facilities. Descriptive analytics techniques are most commonly applied to structured data, although there have been numerous efforts to extend their reach to unstructured data, often through the creation of structured metadata and indices. Descriptive analytics help provide an understanding of the past as well as events occurring in real-time.

Descriptive analytics provides significant insight into business performance and enables users to better monitor and manage their business processes. Additionally, descriptive analytics often serves as a first step in the successful application of predictive or prescriptive analytics. Organizations that effectively use descriptive analytics typically have a single view of the past and can focus their attention on the present, rather than on reconciling different views of the past.

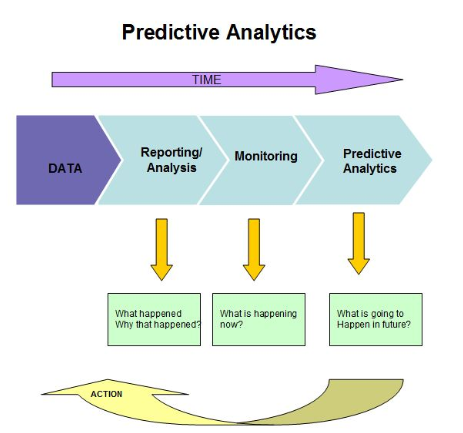
# Predictive Analytics

“Predictive Analytics” attempts to answer the question “what might happen in the future?”  In common usage, Predictive Analytics typically applies more advanced classical statistical techniques such as linear regression to answer a question such as “If I increase my advertising spending by 10%, how much will my sales increase next quarter?”

There are three basic components that underlie predictive analytics:

* The Data: A predictive model is only as good as the historical data that underlies it.  Google's Chief Economist Hal Varian was famous for saying that Google doesn’t have better models; it just has more data.
* The Statistics: This is the set of mathematical techniques, ranging from basic to advanced that are applied to the data to derive inference, meaning, and insight.  The most common statistical technique used in predictive analytics is linear regression, which is basically the iterative process of selecting and testing the impact of variables on the outcome.
* The Assumptions: These are the things that are presumed to be true, with the most common being that the future will continue to be like the past.

Predictive analytics uses many techniques from Data mining, statistics, modelling, artificial intelligence and machine learning to analyse current data and predict about the future **[2].**

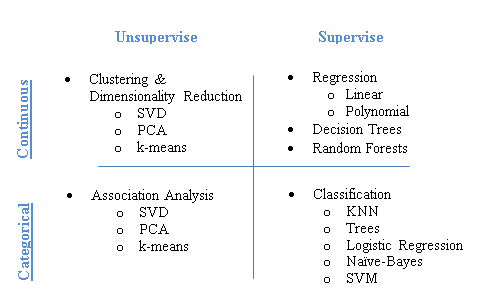


**Figure 1 Predictive Analytics Process**

## Machine learning

Machine learning is a method of data analysis that automates analytical model building. Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look. Machine learning is used in Web search, spam filters, recommender systems, ad placement, credit scoring, fraud detection, stock trading, drug design, and many other applications. As per report from the McKinsey Global Institute machine learning (a.k.a. data mining or predictive analytics) will be the driver of the next big wave of innovation **[3].**

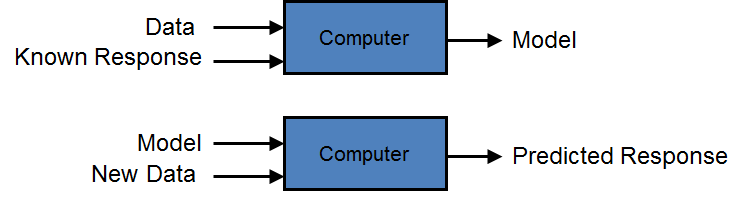
Machine learning Techniques can be broadly classified into two categories; Supervised Learning and Unsupervised Learning.



**Figure 2 : Machine Learning Techniques’**

### Supervised learning

Supervised learning (machine learning) takes a known set of input data and known responses to the data, and seeks to build a predictor model that generates reasonable predictions for the response to new data.



**Figure 3 : Supervised Machine Learning**

The fundamental goal of machine learning is to ‘**Generalize’** beyond the examples in the training set. This is cause, no matter how much data we have; it is very unlikely that we will see those exact examples again at test time. In Unsupervised Learning response is not available. Doing well on the training set is easy (just memorize the examples). The most common mistake among machine learning beginners is to test on the training data and have the illusion of success. If the chosen classifier is then tested on new data, it is often no better than random guessing.

#### Issues with Supervised Learning

##### Bias, Variance,

Imagine that we would like to build an algorithm which will predict the price of a house given its size. Naturally,, we’d expect that the cost of a house grows as the size increases, but there are many other factors which can contribute to the price of the house. Imagine we approach this problem with the polynomial regression machine learning technique. We can vary the degree *d* of the polynomial to get the best fit.

##### 

##### Figure 4 : Model Fitting Issues

In the above figure, we see fits for three different values of *d*. For *d* = 1, the data **is under-fit** and shows **low variance**.. This means that the model is too simplistic: no straight line will ever be a good fit to this data. In this case, we say that the model suffers from high *bias*. The model itself is biased, and this will be reflected in the fact that the data is poorly fit.

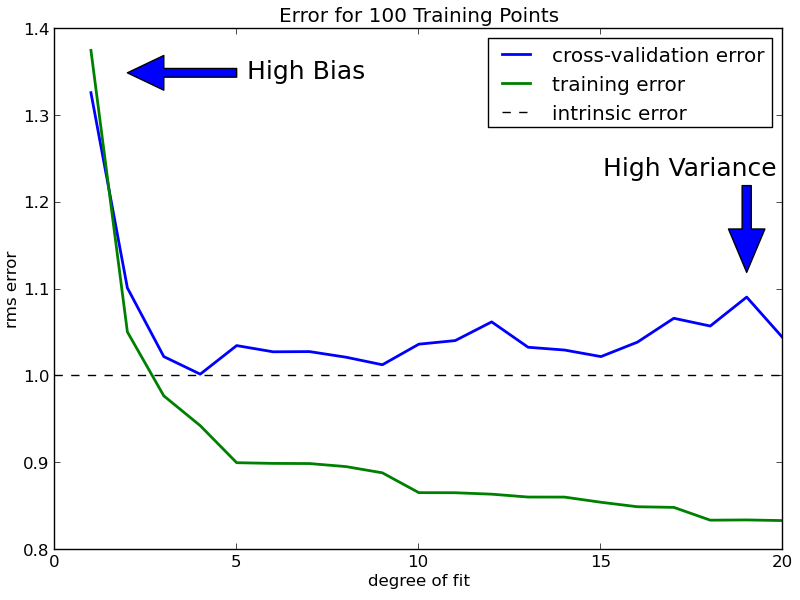
At the other extreme, for *d* = 6 the data is over-fit. This means that the model has too many free parameters (6 in this case) which can be adjusted to perfectly fit the training data. If we add a new point to this plot, chances are it will be very far from the curve representing the degree-6 fit. In this case, we say that the model is **over fit** and suffers from **high variance**

In the middle, for d = 2, we have found a good mid-point. It fits the data fairly well, and does not suffer from the bias and variance problems seen in the figures on either side. What we would like is a way to quantitatively identify bias and variance, and optimize the metaparameters (in this case, the polynomial degree d) in order to determine the best algorithm. This can be done through a process called **cross-validation**.

In order to quantify the effects of bias and variance and construct the best possible estimator, we will split our training data into three parts: a training set, a cross-validation set, and a test set. As a general rule, the training set should be about 60% of the samples, and the cross-validation and test sets should be about 20% each.

The general idea is as follows. The model parameters (in our case, the coefficients of the polynomials) are learned using the training set. . The error is evaluated on the cross-validation set, and the meta-parameters (in our case, the degree of the polynomial) are adjusted so that this cross-validation error is minimized. Finally, the labels are predicted for the test set. These labels are used to evaluate how well the algorithm can be expected to perform on unlabelled data.

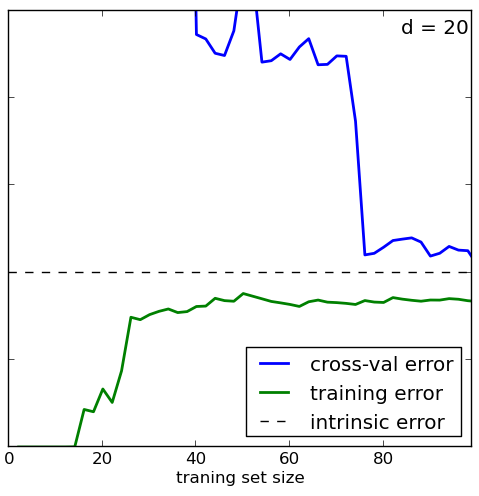
The cross-validation error as a function of the polynomial degree *d* is shown in the following figure:

[](http://www.astroml.org/sklearn_tutorial/auto_examples/plot_bias_variance_examples.html)

**Figure 5 : The training error and cross-validation error as a function of the polynomial degree** *d*.

On the left side of the above plot, we have very low-degree polynomial, which under-fits the data. This leads to a very high error for both the training set and the cross-validation set. On the far right side of the plot, we have a very high degree polynomial, which over-fits the data. This can be seen in the fact that the training error is very low, while the cross-validation error is very high. Plotted for comparison is the intrinsic error (this is the scatter artificially added to the data: click on the above image to see the source code). For this dataset, error = 1.0 is the best we can hope to attain. Choosing *d* = 5 in this case gets us very close to the optimal error.

A learning curve is a plot of the training and cross-validation error as a function of the number of training points. Note that when we train on a small subset of the training data, the training error is computed using this subset, not the full training set. These plots can give a quantitative view into how beneficial it will be to add training samples.



**Figure 6 : Learning Curve for case of high variance ( d = 20)**

In the figure above, we have the learning curve for d = 20. From the above discussion, we know that d = 20 is a high-variance estimator which over-fits the data. This is indicated by the fact that the training error is much less than the cross-validation error. As we add more samples to this training set, the training error will continue to climb, while the cross-validation error will continue to decrease, until they meet in the middle. In this case, our intrinsic error is 1 and we can see that adding more data will allow the estimator to very closely match the best possible cross-validation error.

With a degree-20 polynomial, observe that the training error is identically zero for training set size N<=20. This is because when the degrees of freedom are greater than the number of constraints, the problem should be perfectly solvable: a curve can be found which passes through every point (for example, imagine fitting a line to a single point**. [4]**

Besides cross-validation, there are many other methods to combat over fitting. The most popular one is adding a regularization term to the valuation function. A common misconception about over fitting is that it is caused by noise. But severe over fitting can occur even in the absence of noise. For instance, suppose we learn a Boolean classifier that is just the disjunction of the examples labelled “true” in the training set. (In other words, the classifier is a Boolean formula in disjunctive normal form, where each term is the conjunction of the feature values of one specific training example). This classifier gets all the training examples right and every positive test example wrong, regardless of whether the training data is noisy or not.

If algorithm shows **high bias**, it advisable to **Add more features** to the training and test sets that helps to improve a high-bias estimation.

If algorithm shows **high variance**, the following actions might help:

1. **Use fewer features.** Using a feature selection technique may be useful, and decrease the over-fitting of the estimator.
2. **Use more training samples**. Adding training samples can reduce the effect of over-fitting, and lead to improvements in a high variance estimator.

##### Curse of Dimensionality

Another biggest problem after over fitting in machine learning is the ‘**Curse of Dimensionality**.’ [5]

The Curse of Dimensionality, the expression which was coined by Bellman in 1961 refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience. When the dimensionality increases, the volume of the space increases so fast that; the available data becomes sparse. This sparsity is problematic for any method that requires statistical significance. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality [5]. The "curse of dimensionality" is not a problem of high-dimensional data, but a joint problem of the data and the algorithm being applied. It arises when the algorithm does not scale well to high-dimensional data, typically due to needing an amount of time or memory that is exponential in the number of dimensions of the data. When facing the curse of dimensionality, a good solution can often be found by changing the algorithm, or by pre-processing the data into a lower-dimensional form. For example, its known fact that any low-dimensional data space can trivially be turned into a higher-dimensional space by adding redundant (e.g. duplicate) or randomized dimensions and in turn many high-dimensional data sets can be reduced to lower-dimensional data without significant information loss.

In machine learning ‘**Generalizing’** becomes exponentially harder as the dimensionality (number of features) of the examples grows.

##### Feature Engineering

Success of Machine Learning Projects depends largely on the ‘Feature Selection’. Feature-selection is also called as variable selection or attribute-selection. Feature selection is different from dimensionality reduction. Both methods seek to reduce the number of attributes in the dataset, but a dimensionality reduction method do so by creating new combinations of attributes, where as feature selection methods include and exclude attributes present in the data without changing them.

Following is the suggested checklist that must be used during ‘Feature Selection’ **[6].**

* **Do you have domain knowledge?** If yes, construct a better set of ad hoc features
* **Are your features commensurate?** If no, consider normalizing them.
* **Do you suspect interdependence of features?** If yes, expand your feature set by constructing conjunctive features or products of features, as much as your computer resources allow you.
* **Do you need to prune the input variables (e.g. for cost, speed or data understanding reasons)?** If no, construct disjunctive features or weighted sums of feature
* **Do you need to assess features individually (e.g. to understand their influence on the system or because their number is so large that you need to do a first filtering)?** If yes, use a variable ranking method; else, do it anyway to get baseline results.
* **Do you need a predictor?** If no, stop
* **Do you suspect your data is “dirty” (have a few meaningless input patterns and/or noisy outputs or wrong class labels)?** If yes, detect the outlier examples using the top ranking variables obtained in step 5 as representation; check and/or discard them.
* **Do you know what to try first?** If no, use a linear predictor. Use a forward selection method with the “probe” method as a stopping criterion or use the 0-norm embedded method for comparison, following the ranking of step 5, construct a sequence of predictors of same nature using increasing subsets of features. Can you match or improve performance with a smaller subset? If yes, try a non-linear predictor with that subset.
* **Do you have new ideas, time, computational resources, and enough examples?** If yes, compare several feature selection methods, including your new idea, correlation coefficients, backward selection and embedded methods. Use linear and non-linear predictors. Select the best approach with model selection
* **Do you want a stable solution (to improve performance and/or understanding)?** If yes, subsample your data and redo your analysis for several “bootstrap”.

### Unsupervised Learning

Unsupervised learning tries to find the hidden structure in the absence of the output response. Since the examples given to the learner are unlabeled, there is no error to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning.

Approaches to unsupervised learning techniques include:

* Clustering
  + k-means,
  + mixture models,
  + hierarchical clustering
* Principal component analysis,
* Independent component analysis,
* Singular value decomposition.

With unsupervised learning it is possible to learn larger and more complex models than with supervised learning. This is because in supervised learning one is trying to find the connection between two sets of observations. The difficulty of the learning task increases exponentially in the number of steps between the two sets and thus supervised learning cannot, in practice, learn models with deep hierarchies.

In unsupervised learning, the learning can proceed hierarchically from the observations into ever more abstract levels of representation. Each additional hierarchy needs to learn only one step and therefore the learning time increases (approximately) linearly in the number of levels in the model hierarchy **[7].**

Prescriptive Analytics:

Prescriptive Analytics is a level above predictive analytics both in terms of complexity and value. Historical data is analysed, predictions of what will happen in the future are made using predictive analytics in the context of business rules that are modelled and then following this a recommendation about an optimal action is made. The final decision of which recommended action to follow can be automated through integration into machinery or dynamic control of workflow systems; such decisions are usually high frequency and low impact in nature. Alternatively, the recommendations can simply be provided to support human decision makers and are often higher impact and lower frequency types of decisions. The ability to simulate the future under different sets of assumptions allows scenario analysis to be performed and when combined with optimization techniques enable prescriptive analytics. Such “what if” or scenario modelling lies at the heart of powerful capabilities.

# References

1. http://www.gartner.com
2. http://www.predictiveanalyticstoday.com
3. J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers. Big data: The next frontier for innovation, competition, and productivity.
4. http://www.astroml.org/sklearn\_tutorial/practical.html
5. https://en.wikipedia.org/wiki/Curse\_of\_dimensionality
6. http://jmlr.csail.mit.edu/papers/volume3/guyon03a/guyon03a.pdf
7. http://users.ics.aalto.fi/harri/thesis/valpola\_thesis/node33.html