**Estimating the amount of purchase done by supermarket customers using machine learning**

**ABSTRACT**

This scheme aims to provide an insight in the role intelligent forecasting methods can play in the world of Sales Management. An important assumption throughout this investigation is that data originating from real life processes can behave in a capricious and non-stationary manner. In such case adaptive methods, capable of making accurate forecasts while they need to respond to changes in demand over time, can bring relief and are a must to maintain control over the supply chain. In the context the fields are explored where sophisticated forecasting strategies can play a significant role. We pay attention to the difficulties and limitations that have to be overcome. To emphasize the surplus value these strategies can have for business we illustrate our story with some examples of recent implementations from major players in business intelligence. However see interesting opportunities too, as they have set in a trend with the development of a wide range of applications that are all presented as sophisticated demand forecasting algorithms. Unfortunately these implementations are more an exception than a rule. Therefore one of the goals of this scheme is to contribute to the publicity of these intelligent forecasting techniques and make the reader aware of the opportunities for businesses. The second part of this paper can be viewed as an introduction to the methods that lay on the foundation of these applications. The conventional techniques to solve forecasting problems have their roots in applied mathematics. However, in the last decade modern learning techniques, such as artificial neural networks, have caught up fast and their successful implementations are encouraging to build new applications. Since many companies rarely give away competitive information, we turned to literature and looked for mathematical as well as modern learning methods. Subsequently we selected the most common approaches, which are Linear Regression and Support Vector Machine to forecast the best possibilities for sales forecasting.

**INTRODUCTION**

Forecasting sales is a common task performed by organizations. This usually involves manually intensive processes using spreadsheets that require input from various levels of an organization. This approach introduces bias and is generally not accurate especially during the initial few weeks of a quarter. In fact that's the time when an accurate forecast has the most benefit after all there's little value in providing an accurate forecast in the last week of a quarter.

Though the process of forecasting tends to be complex it is straightforward to determine its accuracy. One simply has to wait until the end of a forecasting period (e.g. end of quarter) and then compare forecasts with actuals. We are confident about the accuracy of our models and are inviting sales leaders to our Man vs Machine Forecasting Duel - give us a day with your data and we'll provide an algorithm based, unbiased forecast. At the end of the quarter you can evaluate our number by comparing with your internal forecast.

**PROPOSED SYSTEM**

In the proposed system we are overcoming the previous systems anomalies and we are applying machine learning algorithms to predict the sales . This is an effective and more accurate way of prediction.

**ALGORITHMS**

**LINEAR REGRESSION**

### ****Simple Linear Regression****

Simple linear regression is useful for finding relationship between two continuous variables. One is predictor or independent variable and other is response or dependent variable. It looks for statistical relationship but not deterministic relationship. Relationship between two variables is said to be deterministic if one variable can be accurately expressed by the other. For example, using temperature in degree Celsius it is possible to accurately predict Fahrenheit. Statistical relationship is not accurate in determining relationship between two variables. For example, relationship between height and weight.

The core idea is to obtain a line that best fits the data. The best fit line is the one for which total prediction error (all data points) are as small as possible. Error is the distance between the point to the regression line.

**Real-time example**

We have a dataset which contains information about relationship between ‘number of hours studied’ and ‘marks obtained’. Many students have been observed and their hours of study and grade are recorded. This will be our training data. Goal is to design a model that can predict marks if given the number of hours studied. Using the training data, a regression line is obtained which will give minimum error. This linear equation is then used for any new data. That is, if we give number of hours studied by a student as an input, our model should predict their mark with minimum error.

Y(pred) = b0 + b1\*x

The values b0 and b1 must be chosen so that they minimize the error. If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.

IMG_256

Figure 2: Error Calculation

If we don’t square the error, then positive and negative point will cancel out each other.

For model with one predictor,

IMG_257

Figure 3: Intercept Calculation

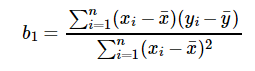


Figure 4: Co-efficient Formula

**Exploring ‘b1’**

* If b1 > 0, then x(predictor) and y(target) have a positive relationship. That is increase in x will increase y.
* If b1 < 0, then x(predictor) and y(target) have a negative relationship. That is increase in x will decrease y.

**Exploring ‘b0’**

* If the model does not include x=0, then the prediction will become meaningless with only b0. For example, we have a dataset that relates height(x) and weight(y). Taking x=0(that is height as 0), will make equation have only b0 value which is completely meaningless as in real-time height and weight can never be zero. This resulted due to considering the model values beyond its scope.
* If the model includes value 0, then ‘b0’ will be the average of all predicted values when x=0. But, setting zero for all the predictor variables is often impossible.
* The value of b0 guarantee that residual have mean zero. If there is no ‘b0’ term, then regression will be forced to pass over the origin. Both the regression co-efficient and prediction will be biased.

**Co-efficient from Normal equations**

Apart from above equation co-efficient of the model can also be calculated from normal equation.

IMG_259

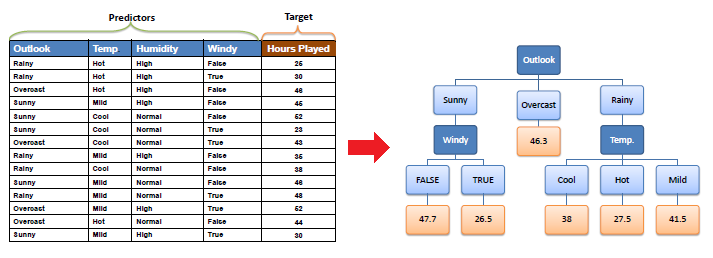
Figure 5: Co-efficient calculation using Normal Equation

Theta contains co-efficient of all predictors including constant term ‘b0’. Normal equation performs computation by taking inverse of input matrix. Complexity of the computation will increase as the number of features increase. It gets very slow when number of features grow large.

**DECISION TREE**

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

The core algorithm for building decision trees called **ID3** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. The ID3 algorithm can be used to construct a decision tree for regression by replacing Information Gain with *Standard Deviation* *Reduction*.



**SOFTWARE REQUIREMENTS:**

OS : Windows

Python IDE : Python 2.7.x and above

Pycharm IDE,

Anaconda 3.5

Setup tools and pip to be installed for 3.6.x and above

**HARDWARE REQUIREMENTS:**

RAM : 4GB and Higher

Processor : Intel i3 and above

Hard Disk : 500GB: Minimum