PROJECT REPORT

ON

Predict Sales Data with Linear Regression

Submitted By

IRESH RASTOGI (18318008)

DEEPAK SINGH (18318009)

SHUBHAM KUMAR SINGH (18318010)

KUMAR SATYAJEET (18303008))

Submitted To

DR. BHARAT GUPTA

JAYPEE INSTITUTE OF INFORMATION AND TECHNOLOGY

ABSTRACT

The task is to forecast the "Sales" column for the test set using linear regression. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of results can be quite varied. Reliable sales forecasts enable store managers to create effective staff schedules that increase productivity and motivation.

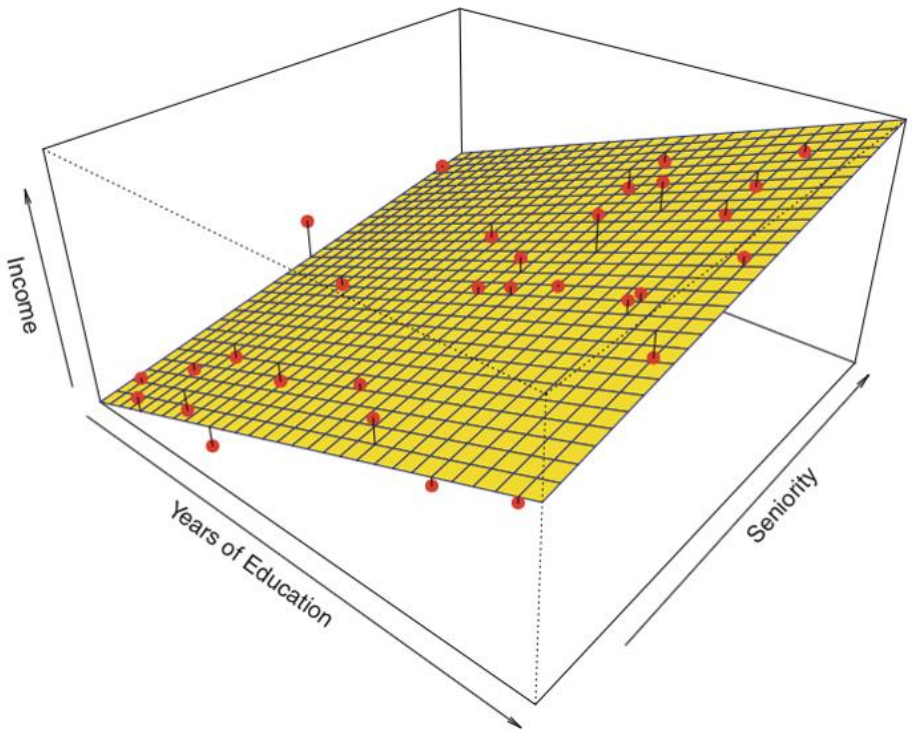
OBJECTIVE

Our objective is to predict the sales of the stores and compare the performance of the model with actual data.

INTRODUCTION

LINEAR REGRESSION ANALYSIS

Linear regression is used for finding linear relationship between target and one or more predictors. There are two types of linear regression- Simple and Multiple.



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.

$$\text{Error} = \sum_{i=1}^{n} (actual\_output - predicted\_output) ** 2$$

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

For model with one predictor,

$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

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.

$$\text{Theta} = (X^T X)^{-1} X^T Y$$

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.

Optimizing using gradient descent

Complexity of the normal equation makes it difficult to use, this is where gradient descent method comes into picture. Partial derivative of the cost function with respect to the parameter can give optimal co-efficient value.

Residual Analysis

Randomness and unpredictability are the two main components of a regression model.

Prediction = Deterministic + Statistic

Deterministic part is covered by the predictor variable in the model. Stochastic part reveals the fact that the expected and observed value is unpredictable. There will always be some information that are missed to cover. This information can be obtained from the residual information.

Let's explain the concept of residue through an example. Consider, we have a dataset which predicts sales of juice when given a temperature of place. Value predicted from regression equation will always have some difference with the actual value. Sales will not match exactly with the true output value. This difference is called as residue.

Residual plot helps in analyzing the model using the values of residues. It is plotted between predicted values and residue. Their values are standardized. The distance of the point from 0 specifies how bad the prediction was for that value. If the value is positive, then the prediction is low. If the value is negative, then the prediction is high. 0 value indicates prefect prediction. Detecting residual pattern can improve the model.

Non-random pattern of the residual plot indicates that the model is,

- Missing a variable which has significant contribution to the model target
- Missing to capture non-linearity (using polynomial term)
- No interaction between terms in model

Characteristics of a residue

- Residuals do not exhibit any pattern
- Adjacent residuals should not be same as they indicate that there is some information missed by system.
- ***Metrics for model evaluation***
- ***R-Squared value***
- This value ranges from 0 to 1. Value '1' indicates predictor perfectly accounts for all the variation in Y. Value '0' indicates that predictor 'x' accounts for no variation in 'y'.

1. Regression sum of squares (SSR)

- This gives information about how far estimated regression line is from the horizontal 'no relationship' line (average of actual output).

$$\text{Error} = \sum_{i=1}^{n} (\text{Predicted\_output} - \text{average\_of\_actual\_output})\text{^}2$$

2. Sum of Squared error (SSE)

- How much the target value varies around the regression line (predicted value).

$$\text{Error} = \sum_{i=1}^{n} (Actual\_output - predicted\_output)\text{^}2$$

3. Total sum of squares (SSTO)

- This tells how much the data point move around the mean.

$$\text{Error} = \sum_{i=1}^{n}(\text{Actual\_output} - \text{average\_of\_actual\_output})\text{^2}$$

R^2 = 1 – (SSE/SSTO)

Correlation co-efficient (r)

This is related to value of 'r-squared' which can be observed from the notation itself. It ranges from -1 to 1.

$r = (+/-) \text{ sqrt}(r^2)$

If the value of b1 is negative, then 'r' is negative whereas if the value of 'b1' is positive then, 'r' is positive. It is unitless.

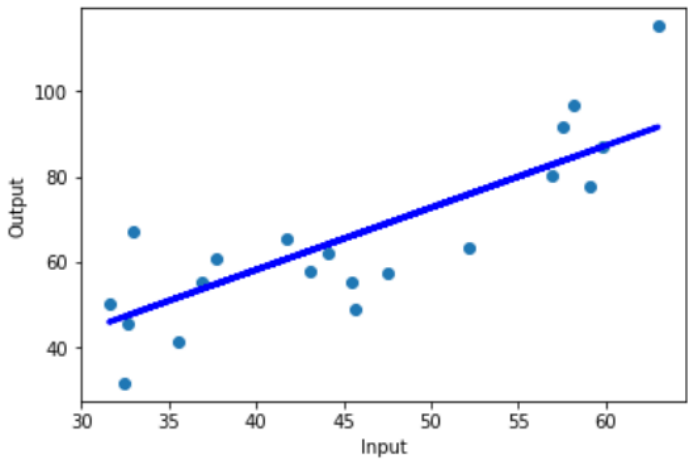
Null-Hypothesis and P-value

Null hypothesis is the initial claim that researcher specify using previous research or knowledge.

Low P-value: Rejects null hypothesis indicating that the predictor value is related to the response

High P-value: Changes in predictor are not associated with change in target

Obtained Regression line



k-Fold Cross-Validation

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. 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. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    1. Take the group as a hold out or test data set
    2. Take the remaining groups as a training data set
    3. Fit a model on the training set and evaluate it on the test set
    4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.

It is also important that any preparation of the data prior to fitting the model occur on the CV-assigned training dataset within the loop rather than on the broader data set. This also applies to any tuning of hyperparameters. A failure to perform these operations within the loop may result in data leakage and an optimistic estimate of the model skill.

The results of a k-fold cross-validation run are often summarized with the mean of the model skill scores. It is also good practice to include a measure of the variance of the skill scores, such as the standard deviation or standard error.

Configuration of k

The k value must be chosen carefully for your data sample.

A poorly chosen value for k may result in a mis-representative idea of the skill of the model, such as a score with a high variance (that may change a lot based on the data used to fit the model), or a high bias, (such as an overestimate of the skill of the model).

Three common tactics for choosing a value for k are as follows:

- **Representative**: The value for k is chosen such that each train/test group of data samples is large enough to be statistically representative of the broader dataset.
- **k=10**: The value for k is fixed to 10, a value that has been found through experimentation to generally result in a model skill estimate with low bias a modest variance.
- **k=n**: The value for k is fixed to n, where n is the size of the dataset to give each test sample an opportunity to be used in the hold out dataset. This approach is called leave-one-out cross-validation.

A value of k=10 is very common in the field of applied machine learning, and is recommend if you are struggling to choose a value for your dataset.

First we explore our data which will provide us a basic understanding of our dataset using various methods like

head() returns few rows of the dataset which gives us an intuation of what kind of data to expect.

describe() displays few summary statistics

count_unique method to count all unique values

isnull() to check for the missing values

EXPERIMENTS AND RESULTS

Finally our data is ready to train and make some predictions. First we separate our dataset into the values we want to predict (*Sales*) and the values to train the algorithm with (all our features like *Promo*, *DayOfWeek_x*, etc).

X is the matrix that contains all data from which we want to be able to predict sales data. So before assigning the values of transformed_data to X we drop the *Sales* column.

y contains only the sales numbers.

The print statement shows us that X is a 1017209 by 14 matrix (14 features and 1017209 training examples).

Then we initialize the LinearRegression model and KFold with 4 folds. This splits our dataset into 4 parts. Our dataset is sorted by date and store-ID so the first fold will contain the oldest data from stores with low IDs and so on, so to ensure that the examples in these folds are random we need to set shuffle=True and set the random_state to a specific value (in this case 42) to get consistent results when we rerun the training and testing.

We use our linear regression model lr, our dataset X, y and kfolds to run cross validation.

Finally cross_val_score runs cross validation four times (because of our KFold with 4 folds) on our data and returns a list of these 4 scores.

Now we'll train our algorithm on sales data from all stores except store 150 (so we don't train and test with the same data) and then predict sales numbers for store 150.

DATASET INFORMATION

Data fields

- **Store** - a unique Id for each store
- **Sales** - the turnover for any given day (this is what you are predicting)
- **Customers** - the number of customers on a given day
- **Open** - an indicator for whether the store was open: 0 = closed, 1 = open
- **StateHoliday** - indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None
- **SchoolHoliday** - indicates if the (Store, Date) was affected by the closure of public schools
- **Promo** - indicates whether a store is running a promo on that day

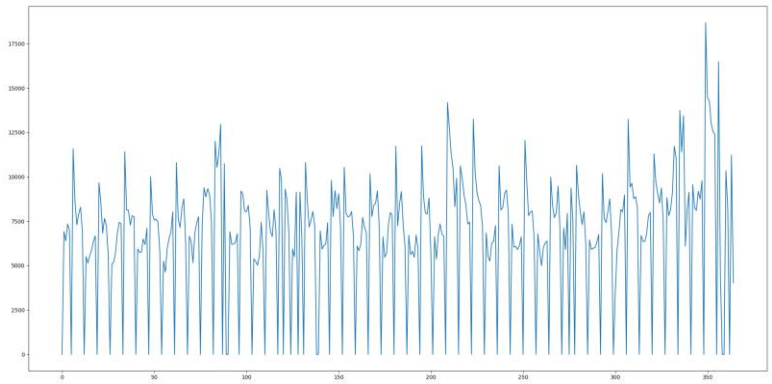
Store	int64
DayOfWeek	int64
Date	object
Sales	int64
Customers	int64
Open	int64
Promo	int64
StateHoliday	object
SchoolHoliday	int64

Number of rows : 1017209

Number of features : 9

RESULT

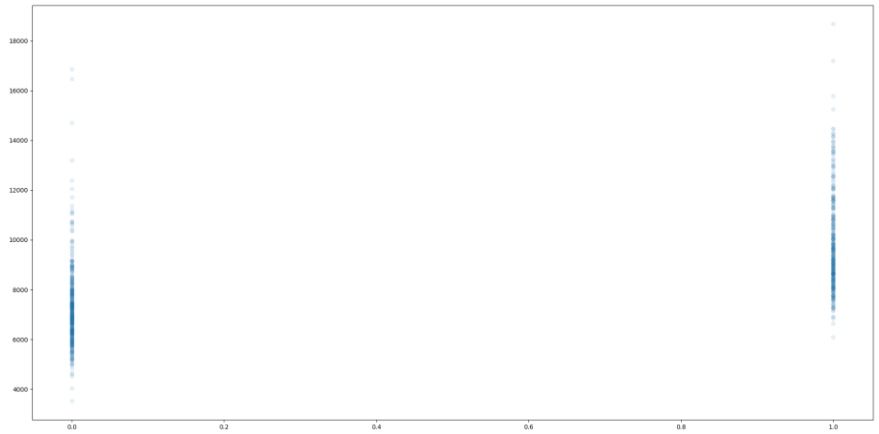
After knowing about data we visualize our data



This graph takes sales number from a specific store and plots first 365 days

We can clearly infer from this graph that the stores are closed on Sunday and sales increases every second week. We can also say that the store must be running sales or promotional offers to increase the sales.

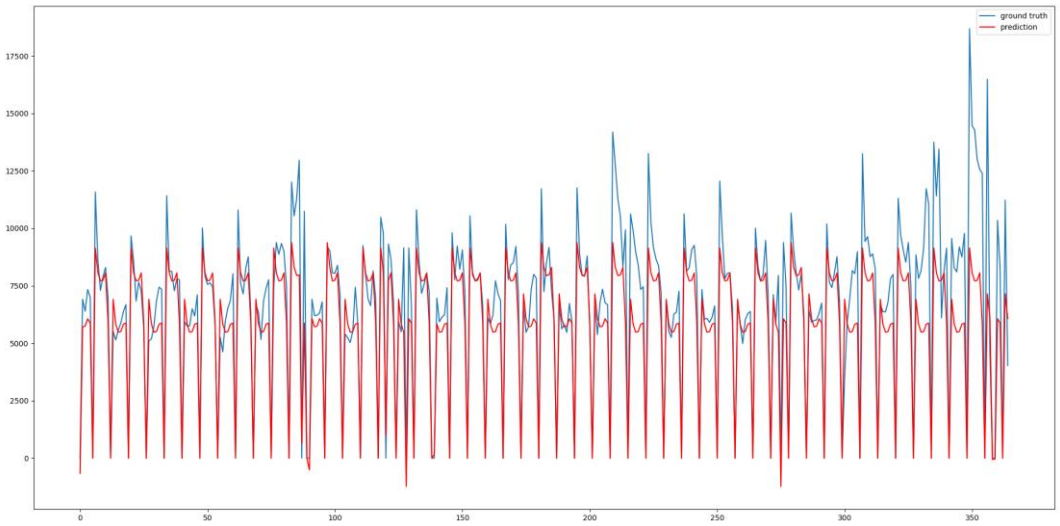
To test that correlation we will plot a scatter plot between the days of promo and sales on that day



Apparently sales are higher when they run a promo on the same day.

After exploring the data we can say something about the feature which will help us to predict sales numbers. We want to predict sales numbers for a specific day and store with a set of features that we know beforehand. So features like Store ,Date and Customers won't be of much help and besides these features we have nominal feature which is DayOfWeek. So we will replace the feature with the binary feature for each value (also called one-hot encoding).

After performing linear regression on the dataset we were able to get 54.594 % accuracy and following plot represents the correlation between actual (blue) and predicted values(red).



CONCLUSION

Regression Analysis is used in the broader sense; however, primarily it is based on quantifying the changes in the dependent variable(regressed variable) due to the changes in the independent variable by using the data on the dependent variables. This is because all the regression models whether linear or non-linear, simple or multiples relate the dependent variable with the independent variables.