# Regression Exercise: Big Mart

BigMart have collected 2013 sales data for 1,559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a store.

Using this model, BigMart will try to understand the properties of products and stores which play a key role in increasing sales:

Problem description: <https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/>).

Data location: <https://www.kaggle.com/brijbhushannanda1979/bigmart-sales-data>,

Feature description: <https://code.datasciencedojo.com/tshrivas/dojoHub/tree/a152a17dee24dcfcc10bb75c77c2e88cdcf90212/Big%20Mart%20Sales%20DataSet>

## Supermarket Sales

Let us use *MRP (maximum retail price)* and the store *establishment year* to estimate sales.

(see also: https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/)

1. Which of the previous two is the most important for sales?d
2. How good is our model? let us use the R-square

lreg.score(x\_cv, y\_cv)

## Then let us improve it by adding more features, let us add *weight*

X = train.loc[:,['Outlet\_Establishment\_Year','Item\_MRP','Item\_Weight']]

x\_train, x\_cv, y\_train, y\_cv = train\_test\_split(X, train.Item\_Outlet\_Sales)

lreg.fit(x\_train,y\_train)

#It **contains NaN, infinity** etc, which is an error. It is because of empty values. Then lets us fill them in with with the averages

train['Item\_Weight'].fillna((train['Item\_Weight'].mean()), inplace=True)

X = train.loc[:,['Outlet\_Establishment\_Year','Item\_MRP','Item\_Weight']]

x\_train, x\_cv, y\_train, y\_cv = train\_test\_split(X, train.Item\_Outlet\_Sales)

lreg.fit(x\_train,y\_train)

pred = lreg.predict(x\_cv)

mse = np.mean((pred - y\_cv)\*\*2)

**# calculating coefficients**

coeff = DataFrame(x\_train.columns)

coeff['Coefficient Estimate'] = Series(lreg.coef\_)

**#calculating r-square**

lreg.score(x\_cv,y\_cv)

## Let us use all features.

## **# imputing missing values**

train['Item\_Visibility'] = train['Item\_Visibility'].replace(0,np.mean(train['Item\_Visibility']))

train['Outlet\_Establishment\_Year'] = 2013 - train['Outlet\_Establishment\_Year']

train['Outlet\_Size'].fillna('Small',inplace=True)

# creating dummy variables to convert categorical into numeric values

mylist = list(train.select\_dtypes(include=['object']).columns)

dummies = pd.get\_dummies(train[mylist], prefix= mylist)

train.drop(mylist, axis=1, inplace = True)

X = pd.concat([train,dummies], axis =1 )

**#Build the model**

import pandas as pd

from pandas import Series, DataFrame

import matplotlib.pyplot as plt

%matplotlib inline

train = pd.read\_csv('Train.csv')

test = pd.read\_csv('Test.csv')

**# importing linear regression**

from sklearn.linear\_model import LinearRegression

lreg = LinearRegression()

**# training a linear regression model on train**

lreg.fit(x\_train,y\_train)

**# predicting on cv**

pred\_cv = lreg.predict(x\_cv)

**# calculating mse**

mse = np.mean((pred\_cv - y\_cv)\*\*2)

mse

**# evaluation using r-square**

lreg.score(x\_cv,y\_cv)

## Polynomial Regression

File: regression-step2.py

1. Try polynomials of different degrees, and see how they compare against linear regression in predicting sales, based on ‘**Outlet establishement yea**r’
2. Do the same for the item ‘**MRP’**

## Feature Selection

## Regularization

**#checking the magnitude of coefficients**

predictors = x\_train.columns

coef = Series(lreg.coef\_,predictors).sort\_values()

coef.plot(kind='bar', title='Modal Coefficients')

## Ridge Regression

from sklearn.linear\_model import Ridge

**## training the model**

ridgeReg = Ridge(alpha=0.05, normalize=True)

ridgeReg.fit(x\_train,y\_train)

pred = ridgeReg.predict(x\_cv)

## calculating mse

mse = np.mean((pred\_cv - y\_cv)\*\*2)

We will consider Ridge Regression, try different values of \alpha, starting from small values e.g. 0.001, 0.01, 0.1, 1, 2

Then try ridge regression on the polynomial

## Checking conditions of regression

1. Check the multicollinearity assumption
2. Check the normal distribution of errors
3. Check the homoskedacity

## Appendix: Features

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Item\_Identifier** | Unique product ID |
| **Item\_Weight** | Weight of product |
| **Item\_Fat\_Content** | Whether the product is low fat or not |
| **Item\_Visibility** | The % of total display area of all products in a store allocated to the particular product |
| **Item\_Type** | The category to which the product belongs |
| **Item\_MRP** | Maximum Retail Price (list price) of the product |
| **Outlet\_Identifier** | Unique store ID |
| **Outlet\_Establishment\_Year** | The year in which store was established |
| **Outlet\_Size** | The size of the store in terms of ground area covered |
| **Outlet\_Location\_Type** | The type of city in which the store is located |
| **Outlet\_Type** | Whether the outlet is just a grocery store or some sort of supermarket |
| **Item\_Outlet\_Sales** | Sales of the product in the particulat store. This is the outcome variable to be predicted. |