

Machine Learning Algorithms – CIA 2

Submitted by Hemanath Kumar J (21121015)

Different factors and firm characteristics that influence the compensation provided to the CEO of a firm

Introduction:

To understand the determinants of CEO compensation, one hundred observations were randomly selected from the 800 listed in the Forbes article. The sample of one hundred CEOs and their firms represent a cross-sectional sample of America's largest corporations. This article provides several measures of CEO compensation, as well as characteristics of the CEO and measures of his firm's performance. The data is used to study CEO and firm characteristics to determine the important factors influencing CEO compensation.

Problem Statement:

To study different factors and firm characteristics that influence the compensation provided to the CEO of a firm.

Variables:

COMP - Sum of salary, bonus and other 1991 compensation, in thousands of dollars (**Dependent Variable**)

AGE - The CEOs age, in years

EDUCATN - The CEOs education level, 1 for no college degree, 2 for a college undergraduate degree and 3 for a graduate degree

BACKGRD - Background type, 0 for unknown, 1 for technical, 2 for insurance, 3 for operations, 4 for banking, 5 for legal, 6 for marketing, 7 for administration, 8 for sales, 9 for financial and 10 for journalism

TENURE - Number of years employed by the firm

EXPER - Number of years as the firm CEO

SALES - 1991 sales revenues, in millions of dollars

VAL - Market value of the CEO's stock, in natural logarithmic units

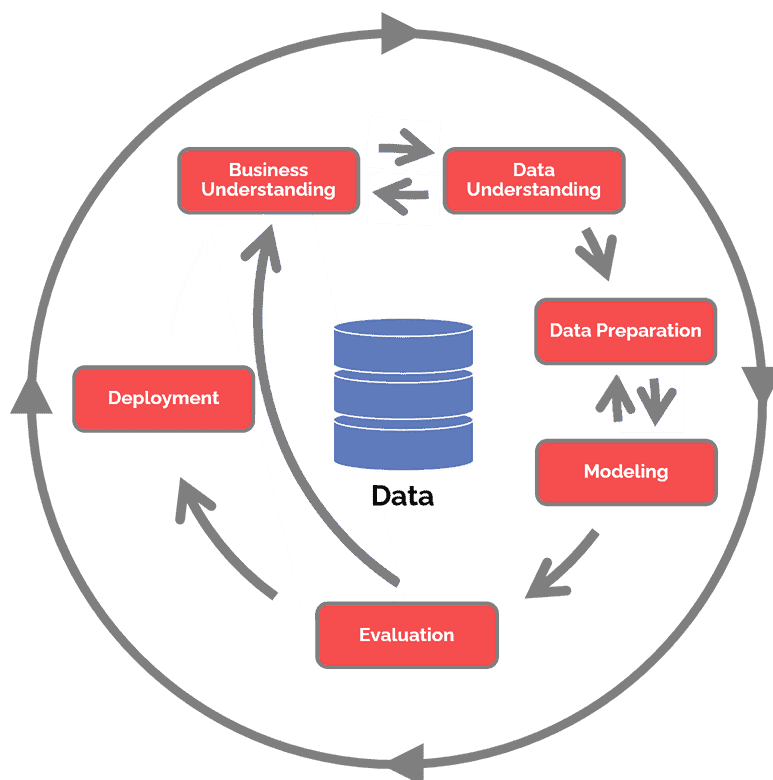
PCNTOWN - Percentage of firm's market value owned by the CEO

PROF - 1991 profits of the firm, before taxes, in millions of dollars

COMPANY - Company name

BIRTH - The CEOs birthplace

CRISP-DM METHODOLOGY(Cross-InduStrY Process for Data Mining)



The analysis is proceeded with the idea of CRISP-DM methodology. This is because, the CRISP-DM methodology provides a structured approach to planning a data mining project. It is a robust and well-proven methodology. It is known for its flexibility and its usefulness when using analytics to solve thorny business issues.

The first step in the CRISP-DM process is to figure out what you want to achieve in terms of business. There may be competing aims and limits in your organisation that must be properly balanced. The purpose of this stage of the process is to identify important aspects that may have an impact on the project's outcome.

The data indicated in the project resources must be acquired in the second stage of the CRISP-DM procedure. If data loading is required for data comprehension, this is included in the initial collection. If you use a certain tool for data analysis, for example, it makes perfect sense to import your data into that tool and analyse it.

Third stages include the process of Selecting the data, Cleaning the data, Construct the required data and integrate the data. Data manipulation comes into play in this stage.

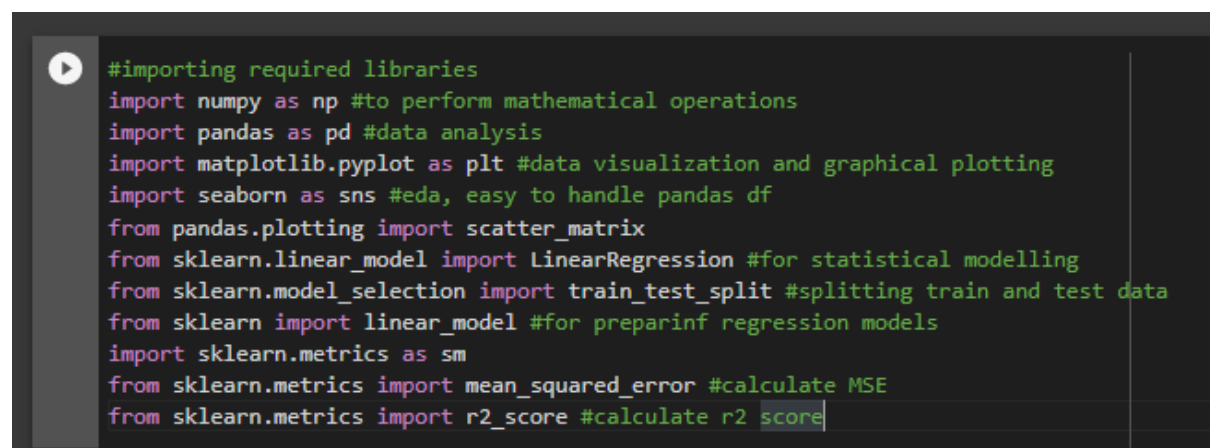
You'll choose the modelling technique you'll use when you're in the Modelling stage. Although you may have already chosen a tool during the business knowledge step, you will now choose a modelling technique, such as decision-tree building using a specific library, etc at this stage. If you're using numerous approaches, do this task for each one separately.

During the next step, which is Evaluation, you'll assess how well the model fulfils your company objectives and try to figure out if there's a business reason why it's not working. In addition to analysing any other data mining results you've generated, the assessment phase entails assessing any other data mining outcomes you've generated.

In the final deployment stage, you'll take the outcomes of your evaluation and devise a strategy for implementing them. It makes sense to think about deployment options at the business understanding phase as well, because deployment is critical to the project's success. This is where predictive analytics may truly assist you enhance the company's operations.

ANALYSIS IN PYTHON

Now we will be getting into the analysis part that has been done using PYTHON with respect to the problem statement. Google Colab is used to write and execute the program. The libraries which are used in the for the analysis are mentioned in the below image.

A screenshot of a Google Colab code cell. On the left, there is a play button icon. The code cell contains a series of Python import statements for various libraries used in data analysis and machine learning. The code is as follows:

```
#importing required libraries
import numpy as np #to perform mathematical operations
import pandas as pd #data analysis
import matplotlib.pyplot as plt #data visualization and graphical plotting
import seaborn as sns #eda, easy to handle pandas df
from pandas.plotting import scatter_matrix
from sklearn.linear_model import LinearRegression #for statistical modelling
from sklearn.model_selection import train_test_split #splitting train and test data
from sklearn import linear_model #for preparing regression models
import sklearn.metrics as sm
from sklearn.metrics import mean_squared_error #calculate MSE
from sklearn.metrics import r2_score #calculate r2 score
```

MOUNTING THE LOCATION

```
✓ [2] #connecting to cloud
29s from google.colab import drive
drive.mount("/content/gdrive")

Mounted at /content/gdrive

✓ [3] %cd /content/gdrive/My Drive/Colab Notebooks/
0s

/content/gdrive/My Drive/Colab Notebooks
```

After mounting the location to Google drive, the file path in which the dataset is available is loaded. From the above screenshot, we can see that the dataset file is available in Colab Notebooks folder.

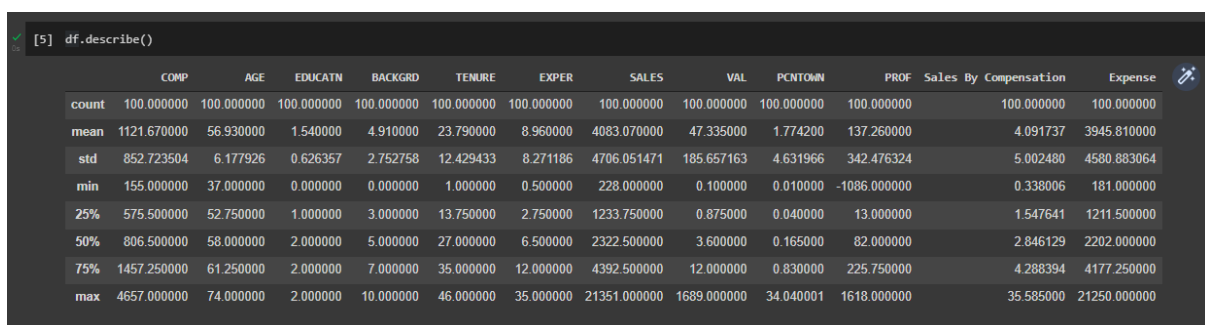
READING THE DATA

```
[4] #data exploration
df = pd.read_csv("CeoCompensationupdated.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   COMP                  100 non-null   int64
1   AGE                   100 non-null   int64
2   EDUCATN              100 non-null   int64
3   BACKGRD              100 non-null   int64
4   TENURE               100 non-null   int64
5   EXPER                100 non-null   float64
6   SALES                 100 non-null   int64
7   VAL                  100 non-null   float64
8   PCNTOWN              100 non-null   float64
9   PROF                 100 non-null   int64
10  COMPANY              100 non-null   object
11  BIRTH                99 non-null    object
12  Sales By Compensation 100 non-null   float64
13  Expense              100 non-null   int64
dtypes: float64(4), int64(8), object(2)
memory usage: 11.1+ KB
```

The dataset name is “CeoCompensationupdated.csv”, which is named as df, for easy reference. Now the information of the dataset is seen by the function df.info(). We can see that there are 14 variables. In the given dataset, there were only 12 variables. Variable number 12 and 13 has been added in the process of data manipulation to get the desired R2 score. Sales by compensation is the percentage of sales compared to compensation and the Expense is basically found out using Profitability of the CEO.

SUMMARY OF THE DATA



```
[5] df.describe()
```

	COMP	AGE	EDUCATN	BACKGRD	TENURE	EXPER	SALES	VAL	PCNTOWN	PROF	Sales By Compensation	Expense
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
mean	1121.670000	56.930000	1.540000	4.910000	23.790000	8.960000	4083.070000	47.335000	1.774200	137.260000	4.091737	3945.810000
std	852.723504	6.177926	0.626357	2.752758	12.429433	8.271186	4706.051471	185.657163	4.631966	342.476324	5.002480	4580.883064
min	155.000000	37.000000	0.000000	0.000000	1.000000	0.500000	228.000000	0.100000	0.010000	-1086.000000	0.338006	181.000000
25%	575.500000	52.750000	1.000000	3.000000	13.750000	2.750000	1233.750000	0.875000	0.040000	13.000000	1.547641	1211.500000
50%	806.500000	58.000000	2.000000	5.000000	27.000000	6.500000	2322.500000	3.600000	0.165000	82.000000	2.846129	2202.000000
75%	1457.250000	61.250000	2.000000	7.000000	35.000000	12.000000	4392.500000	12.000000	0.830000	225.750000	4.288394	4177.250000
max	4657.000000	74.000000	2.000000	10.000000	46.000000	35.000000	21351.000000	1689.000000	34.040001	1618.000000	35.585000	21250.000000

The describe() function is used to get the summary of the dataset, which is similar to summary() function in R. This is one of the processes done as a part of Exploratory Data Analysis. From the above screenshot, we can find out the Mean, Standard Deviation, Minimum and Maximum value in a variable, Quadrant average of each variable. This is analysed basically to understand the dataset which we are working on.

CORRELATION

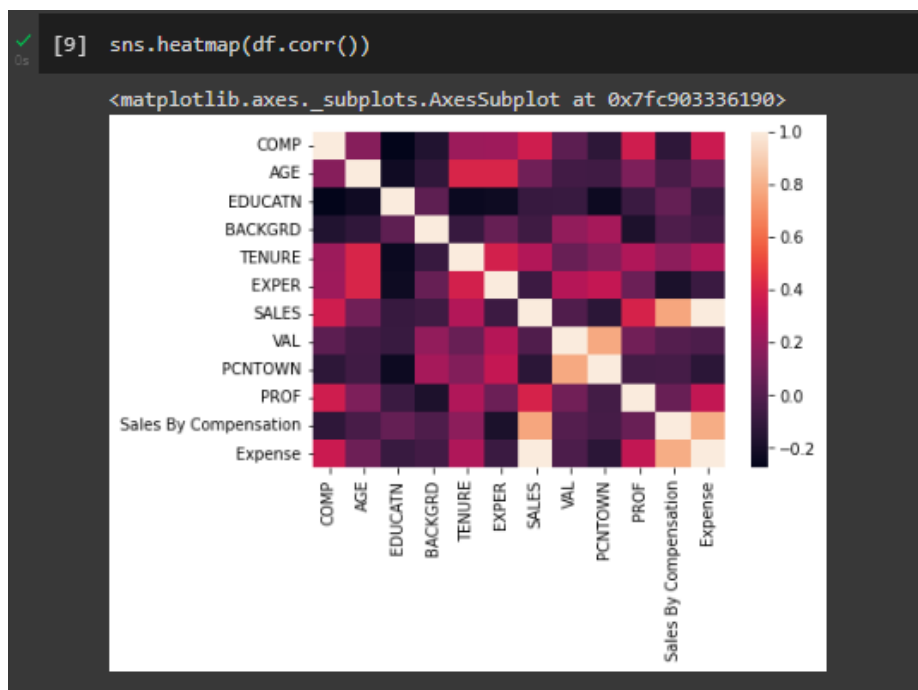
```
[6] df.corr()
```

	COMP	AGE	EDUCATN	BACKGRD	TENURE	EXPER	SALES	VAL	PCNTOWN	PROF	Sales By Compensation	Expense
COMP	1.000000	0.152339	-0.276684	-0.163288	0.219522	0.225994	0.375510	0.030027	-0.124731	0.375028	-0.119938	0.357732
AGE	0.152339	1.000000	-0.222455	-0.115008	0.408119	0.407652	0.088300	-0.055871	-0.059319	0.130514	-0.036061	0.080955
EDUCATN	-0.276684	-0.222455	1.000000	0.034330	-0.238290	-0.229756	-0.088677	-0.087105	-0.234718	-0.079675	0.056487	-0.085144
BACKGRD	-0.163288	-0.115008	0.034330	1.000000	-0.091781	0.059510	-0.066015	0.189340	0.249127	-0.180608	-0.013229	-0.054317
TENURE	0.219522	0.408119	-0.238290	-0.091781	1.000000	0.390031	0.286167	0.066009	0.142358	0.281460	0.174238	0.272944
EXPER	0.225994	0.407652	-0.229756	0.059510	0.390031	1.000000	-0.072703	0.296672	0.338686	0.076148	-0.184995	-0.080383
SALES	0.375510	0.088300	-0.088677	-0.066015	0.286167	-0.072703	1.000000	-0.011674	-0.129357	0.397007	0.774920	0.997643
VAL	0.030027	-0.055871	-0.087105	0.189340	0.066009	0.296672	-0.011674	1.000000	0.782427	0.094723	0.006057	-0.019075
PCNTOWN	-0.124731	-0.059319	-0.234718	0.249127	0.142358	0.338686	-0.129357	0.782427	1.000000	-0.047284	-0.047169	-0.129357
PROF	0.375028	0.130514	-0.079675	-0.180608	0.281460	0.076148	0.397007	0.094723	-0.047284	1.000000	0.064641	0.333093
Sales By Compensation	-0.119938	-0.036061	0.056487	-0.013229	0.174238	-0.184995	0.774920	0.006057	-0.047169	0.064641	1.000000	0.791261
Expense	0.357732	0.080955	-0.085144	-0.054317	0.272944	-0.080383	0.997643	-0.019075	-0.129357	0.333093	0.791261	1.000000

Through `corr()` function, we will be knowing the correlation between two variables. Correlation is a statistical measure that expresses the extent to which two variables are linearly related (meaning they change together at a constant rate). It's a common tool for describing simple relationships without making a statement about cause and effect.

From the result, it is seen that Sales and Expense is highly correlated with the value of 0.997643. Education has the greatest number of negatively correlated values from which we can say that as the Educations' value increases, other variables' value decreases.

HEATMAP



In the above heatmap, the darker shades of the chart represent higher values than the lighter shades of the chart. An effect scores closer to 0 translates to there being no relationship. A score closer to 1 or -1 is a positive or negative relationship. A perfect score of 1 is a direct correlation. Additionally, taking action using these values without testing the normalcy/distribution of your data is not recommended.

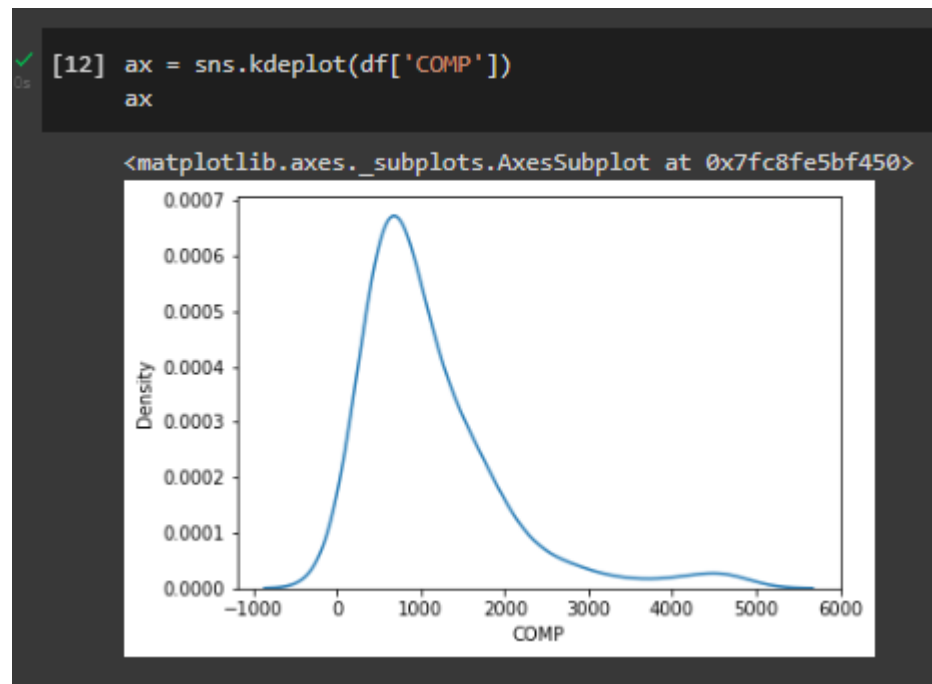
SCATTER MATRIX



Scatter Matrices are a great way to roughly determine if you have a linear correlation between multiple variables. This is particularly helpful in pinpointing specific variables that might have similar correlations to your genomic or proteomic data.

The variables are chosen on the basis of correlation values. The datapoints are very much dispersed between comp and sales which indicates that the relation is weak. The relationship is strong between Sales by comp and comp.

KERNEL DENSITY ESTIMATION

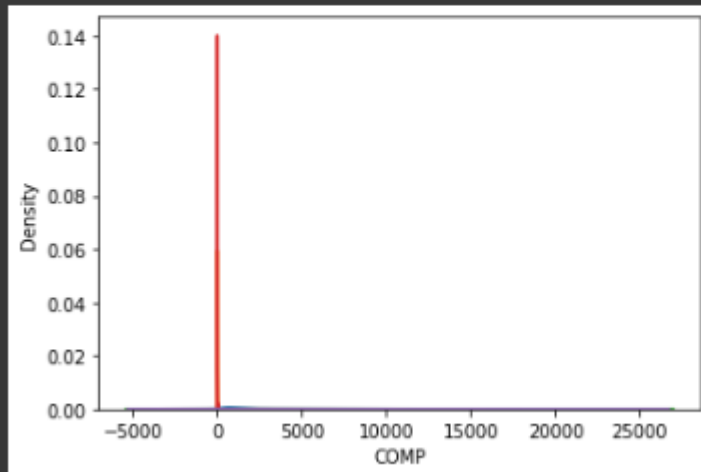


Kernel density estimation is a technique for estimation of probability density function that is a must-have enabling the user to better analyse the studied probability distribution. The Kernel Density Estimation works by plotting out the data and beginning to create a curve of the distribution. The curve is calculated by weighing the distance of all the points in each specific location along the distribution.

The dependent variable COMP is left skewed, which indicated that the mean is greater than the median.


```
[14] comp = ['COMP', 'EXPER', 'SALES', 'Sales By Compensation', 'Expense']
```

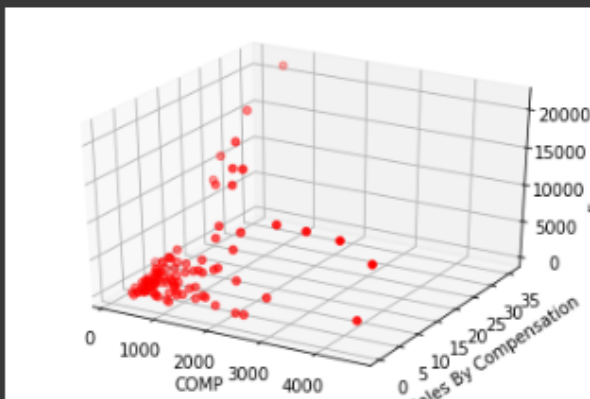
```
for col in comp:  
    ax_comp = sns.kdeplot(df[col])  
    ax_comp
```



KDE for few of the variables based on the correlation values are plotted. The density is highest for the dependent variable whereas other variables have low density.

3D – Graph

```
[ ] data = df[["COMP","Sales By Compensation","Expense"]]  
  
[ ] fig=plt.figure()  
    ax=fig.add_subplot(111,projection='3d')  
    n=100  
    ax.scatter(data["COMP"],data["Sales By Compensation"],data["Expense"],color="red")  
    ax.set_xlabel("COMP")  
    ax.set_ylabel("Sales By Compensation")  
    ax.set_zlabel("Expense")  
    plt.show()
```



The 3d Graph is used to find the dispersion among 3 variables for better understanding of those variables. The variables used are the dependent variable, which is COMP and two other variables which are added in the process of data manipulation.

From the graph, we can infer that as the sales percentage grows, Compensation given to the CEO is also increased. Only in few exceptions, Compensation is not influenced by the Sales percentage. In the same way, the more expense spent by a CEO has less compensation.

UNIVARIATE LINEAR REGRESSION

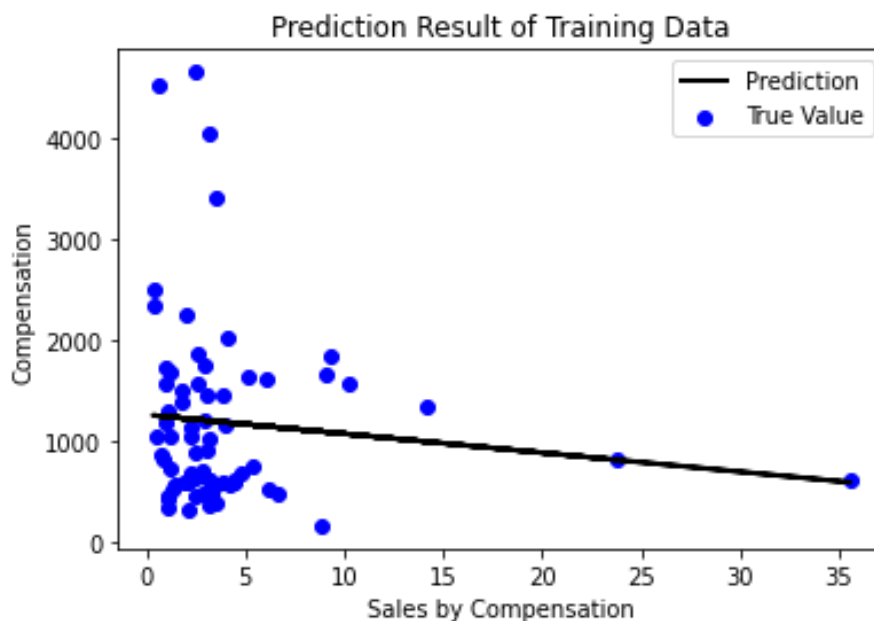
```
[24] X = df['Sales By Compensation']
      Y = df['COMP']

[25] X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.3, random_state=42)
      X_train = np.array(X_train).reshape((len(X_train),1))
      Y_train = np.array(Y_train).reshape((len(Y_train),1))
      X_test = np.array(X_test).reshape(len(X_test), 1)
      Y_test = np.array(Y_test).reshape(len(Y_test), 1)
      model = linear_model.LinearRegression()
      model.fit(X_train, Y_train)

      Y_train_pred = model.predict(X_train)

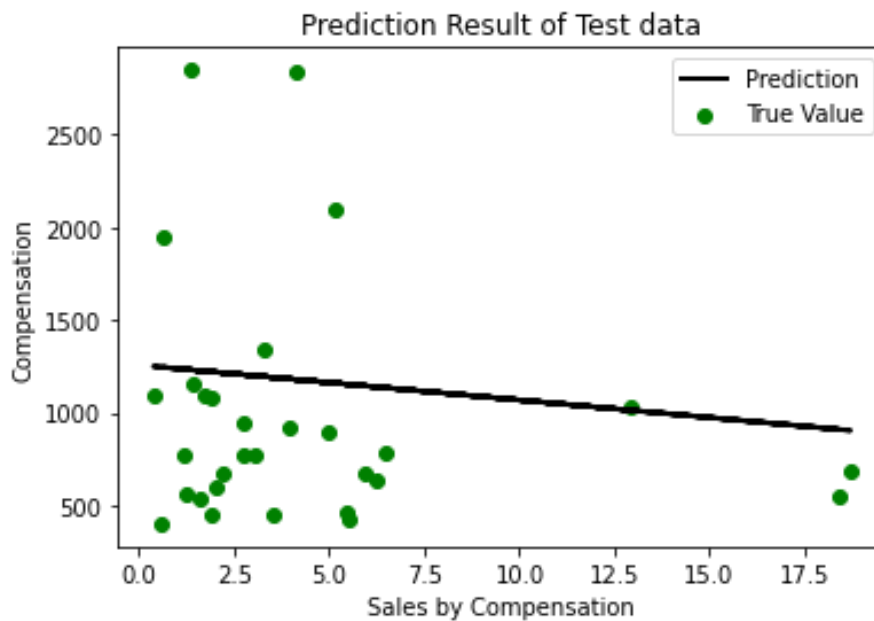
      plt.figure()
      plt.scatter(X_train, Y_train, color='blue', label="True Value")
      plt.plot(X_train, Y_train_pred, color='black', linewidth=2, label="Prediction")
      plt.xlabel("Sales by Compensation")
      plt.ylabel("Compensation")
      plt.title('Prediction Result of Training Data')
      plt.legend()
      plt.show()
```

30 % of the dataset is considered as test data and simple univariate linear regression is performed. At this stage, we have trained a linear model and we first use it to predict the compensation on our training set to see how well it fit on the data. Two chosen variables are Sales by Comp and COMP. Using the linear model, we are going to predict the Compensation based on the training set.



Using the Matplotlib, a plot to visualize the predicted results is created. The “true values” are plotted as the blue dots on the chart and the predicted values are plotted as a black colour straight line. In general, the linear model fits well on the training data as the datapoints are closer to the prediction line and there are only very few datapoints which are uncommon. This shows a linear relationship between the Compensation and Sales by Compensation.

Now, we need to check if the linear model can perform well on our test set (unknown data).



Here, the Matplotlib is used to plot and visualize the predicted results. The “true values” are plotted as the green dots on the chart and the predicted values are plotted as a black colour straight line. The graph shows that our linear model can fit quite well on the test set. We can observe a linear pattern of how the amount of CEO is increased by the Sales by Compensation.

MULTIVARIATE LINEAR REGRESSION

In the previous simple linear regression, we used a graphical approach to evaluate the performance of our linear model which can sometimes be quite subjective in judgment. Here, in Multivariate Linear Regression, we will use some quantitative methods to obtain a more precise performance evaluation of our linear model.

We will use three types of quantitative metrics:

- Mean Square Error — The average of the squares of the difference between the true values and the predicted values. The lower the difference the better the performance of the model. This is a common metric used for regression analysis.
- Intercept and Coefficients
- R2 Score — A measurement to examine how well our model can predict values based on the test set (unknown samples). The perfect score is 1.0.

```
#generating training and testing data from our data:
# We are using 30% data for training.
train = df[:int((len(df)*0.3))]
test = df[int((len(df)*0.3):)]

# Modeling:
# Using sklearn package to model data :
regr = linear_model.LinearRegression()
train_x = np.array(train[["Sales By Compensation", "Expense", "AGE", "TENURE", "EXPER", "SALES", "VAL", "PROF", "EDUCATN", "BACKGRD", "PCNTOWN"]])
train_y = np.array(train[["COMP"]])
regr.fit(train_x, train_y)

# Predicting values:
# Function for predicting future values
def get_regression_predictions(input_features, intercept, slope):
    predicted_values = input_features*slope + intercept
    return predicted_values

# Checking various accuracy:
from sklearn.metrics import r2_score
test_x = np.array(test[["Sales By Compensation", "Expense", "AGE", "TENURE", "EXPER", "SALES", "VAL", "PROF", "EDUCATN", "BACKGRD", "PCNTOWN"]])
test_y = np.array(test[["COMP"]])
test_y_ = regr.predict(test_x)
print("Mean sum of squares (MSE): %.2f" % np.mean((test_y_ - test_y)** 2))
print("coefficients: %.2f" % np.mean(np.absolute(test_y_ - test_y)))
y_pred=regr.predict(x)
print("R2-score: %.2f" % r2_score(test_y_ , test_y ) )
```

Training and testing the data using the train and test variables (respectively). Only 30% of the data is used for training. We use the sklearn package to model data that returns multivariable regression values. We are getting the values of MSE, Coefficients, Intercept and R2 score.

RESULT

```
MSE : 264793.030684971
coefficients : [[-1.45318730e+01  3.99563642e+00  1.60242798e+01  6.58331424e-02
 1.08470567e+00 -6.79981035e-02 -2.03934322e+02 -3.10928088e+01
 -5.40339206e+01 -1.63970385e+02  1.33831246e-01]]
Intercept : [2104.96594183]
R2 score : 0.6321634439182421
```

The R-squared value obtained from this model is 0.6322 which means that the above model explains a 63.2% of the variation in the Compensation with respect to other variables.

Also, few of the variables had p-value greater than 0.05, which can be optimized in the model by removing the insignificant variables.

CONCLUSION

We have managed to build a multivariate linear model to predict compensation based on few independent variables. Based on our linear model, we can conclude that Compensation is grown with years of working experience, Sales percentage and profit the most, as there is a linear relationship between them. We can use this linear model to predict the CEO compensation by giving input of the above-mentioned variables.

Although it is a little unwise to assume that compensation is solely based on years of experience, sales percentage, and profit, this is an excellent example of how to create a multivariate linear regression model to illustrate a link between many variables. In reality, most real-life occurrences are difficult to describe using a linear model. Knowing how to develop a linear model, on the other hand, is the foundation for understanding how to build a complicated model.