Mini Project on Factor analysis

In today’s world, data can be seen everywhere.Business owners are trying hard to get the insights from their data.

In fact, 90% of the data in the world has been generated in the last 3-4 years! The numbers are truly mind boggling. Below are just some of the examples of the kind of data being collected:

Need of Dimensionality Reduction :

Here i have listed few benefits of using dimensionality reduction techniques.

- Storage of data is reduced as the dimensions are reduced.

- It is easy to do the computation with less dimensions.

- It takes care of multicollinearity by removing redundant features. For example, you have two variables – Date of Birth and age. These variables are highly correlated.you can easily compute the one from another.Hence, there is no point in storing both as just one of them does what you require.

- It helps in visualizing data. As discussed earlier, it is very difficult to visualize data in higher dimensions so reducing our space to 2D or 3D may allow us to plot and observe patterns more clearly

Among many dimension reduction technique,Factor Analysis is the prominent one.

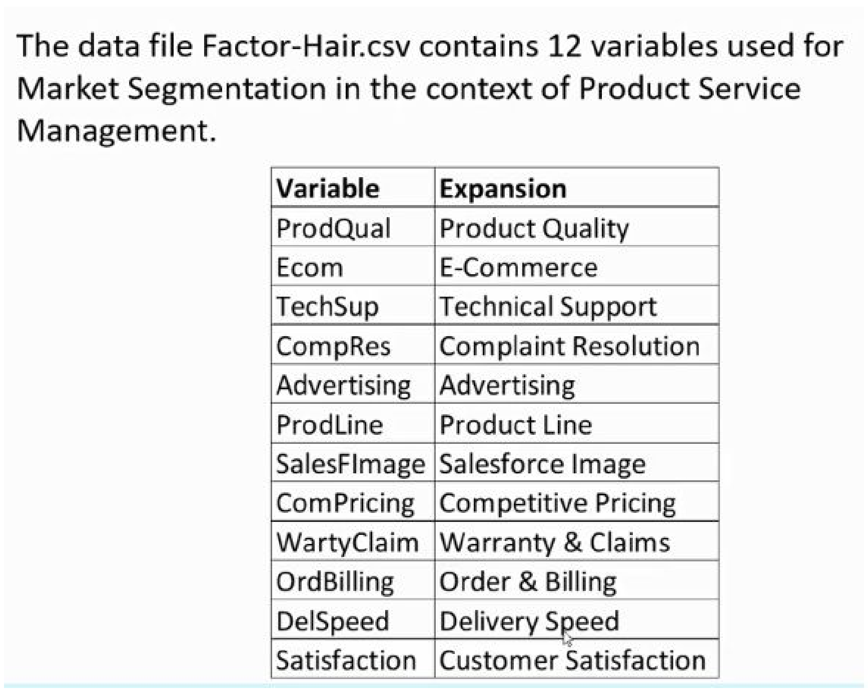
In the Factor Analysis technique, variables are grouped by their correlations, i.e., all variables in a particular group will have a high correlation among themselves, but a low correlation with variables of other group(s). Here, each group is known as a factor. These factors are small in number as compared to the original dimensions of the data. However, these factors are difficult to observe.

And Thanks to python libraries such as **Numpy**,**Pandas**,**Scipy**,**Scikit** which made our job easier.

Here for the given csv file **Factor-Hair-Revised.csv**,Factor analysis should be applied and the dimensions should be reduced.

List of variables

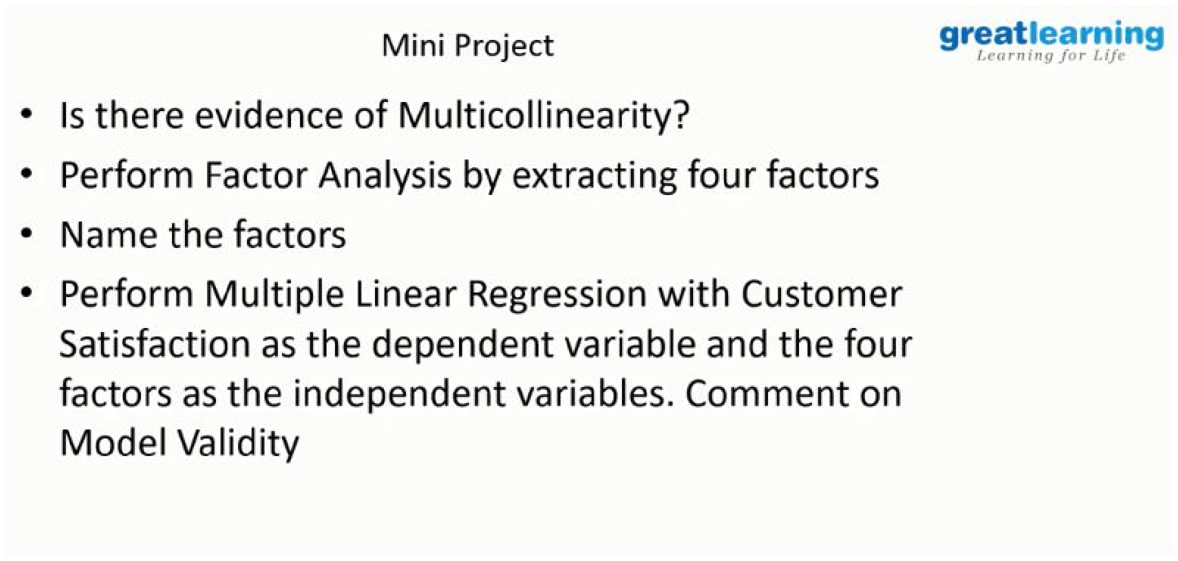
These variables used for market segmentation in the context of product service management.



The data file Factor-Hair.csv contains 12 variables used for Market Segmentation in the context of Product Service Management.

Variable	Expansion
ProdQual	Product Quality
Ecom	E-Commerce
TechSup	Technical Support
CompRes	Complaint Resolution
Advertising	Advertising
ProdLine	Product Line
SalesFImage	Salesforce Image
ComPricing	Competitive Pricing
WartyClaim	Warranty & Claims
OrdBilling	Order & Billing
DelSpeed	Delivery Speed
Satisfaction	Customer Satisfaction

And the questions to focus on



Mini Project

greatlearning
Learning for Life

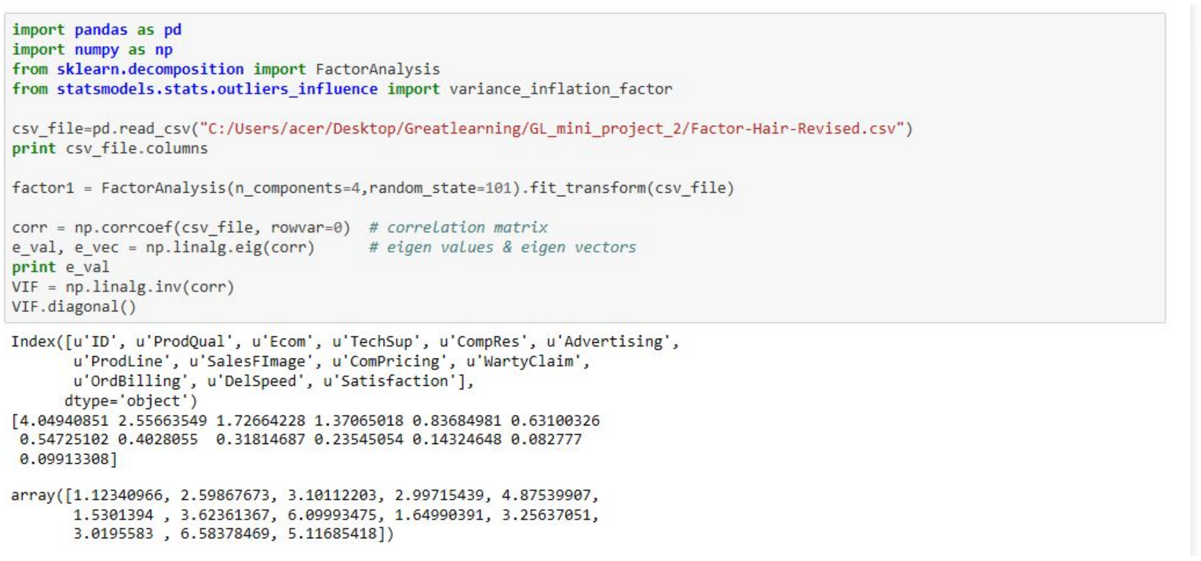
- Is there evidence of Multicollinearity?
- Perform Factor Analysis by extracting four factors
- Name the factors
- Perform Multiple Linear Regression with Customer Satisfaction as the dependent variable and the four factors as the independent variables. Comment on Model Validity

Let us dive into the project and get the needed output.

Q1.Is there a evidence of multicollinearity?

The presence of Multicollinearity reduces the precision of the estimate coefficients, which weakens the statistical power of your regression model. Multicollinearity can be detected using Variance inflation Factor(VIF)

If the VIF is equal to 1 there is no multicollinearity among factors, but if the VIF is greater than 1, the predictors may be moderately correlated. The factors above 1.5, which indicates some correlation, but not enough to be overly concerned about. A VIF between 5 and 10 indicates high correlation that may be problematic. And if the VIF goes above 10, you can assume that the regression coefficients are poorly estimated due to multicollinearity.



```python
import pandas as pd
import numpy as np
from sklearn.decomposition import FactorAnalysis
from statsmodels.stats.outliers_influence import variance_inflation_factor

csv_file=pd.read_csv("C:/Users/acer/Desktop/Greatlearning/GL_mini_project_2/Factor-Hair-Revised.csv")
print csv_file.columns

factor1 = FactorAnalysis(n_components=4,random_state=101).fit_transform(csv_file)

corr = np.corrcoef(csv_file, rowvar=0)  # correlation matrix
e_val, e_vec = np.linalg.eig(corr)      # eigen values & eigen vectors
print e_val
VIF = np.linalg.inv(corr)
VIF.diagonal()
```

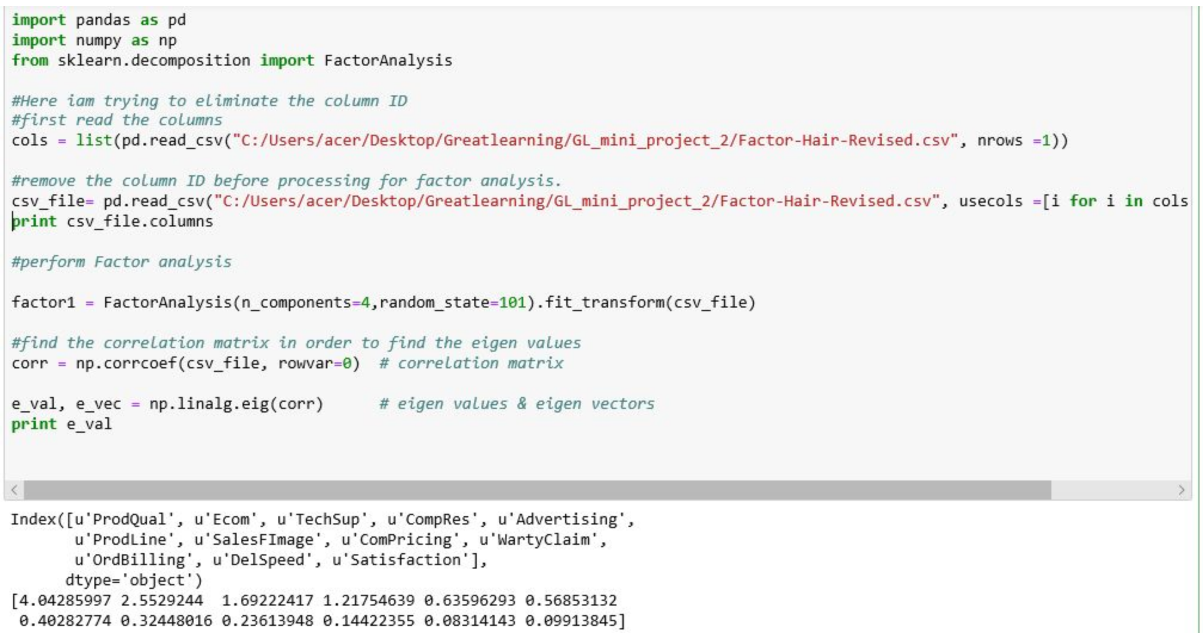
```
Index([u'ID', u'ProdQual', u'Ecom', u'TechSup', u'CompRes', u'Advertising',
       u'ProdLine', u'SalesFImage', u'ComPricing', u'WartyClaim',
       u'OrdBilling', u'DelSpeed', u'Satisfaction'],
      dtype='object')
[4.04940851 2.55663549 1.72664228 1.37065018 0.83684981 0.63100326
 0.54725102 0.4028055  0.31814687 0.23545054 0.14324648 0.082777
 0.09913308]

array([1.12340966, 2.59867673, 3.10112203, 2.99715439, 4.87539907,
       1.5301394 , 3.62361367, 6.09993475, 1.64990391, 3.25637051,
       3.0195583 , 6.58378469, 5.11685418])
```

- **No Collinearity**:VIF=1   None
- **Moderate Collinearity**:VIF>1   Advertising and Competitive pricing are >1.5.Product Quality,Ecommerce,Technical Support,Complaint resolution,Production Line,Warrenty Claim,Order and billing are 2.5 to 5
- **High Collinearity**:VIF >5 and <10 : Delivery speed,Satisfaction and Sales Force Image has VIF >5,hence these three are highly correlated.so these can be removed or ignored.

Q2 . Perform Factor analysis by extracting four factors

In every factor analysis, there are the same number of factors as there are variables. Each factor captures a certain amount of the overall variance in the observed variables, and the factors are always listed in order of how much variation they explain.

The eigenvalue is a measure of how much of the variance of the observed variables a factor explains. Any factor with an eigenvalue ≥1 explains more variance than a single observed variable.



```
import pandas as pd
import numpy as np
from sklearn.decomposition import FactorAnalysis

#Here iam trying to eliminate the column ID
#first read the columns
cols = list(pd.read_csv("C:/Users/acer/Desktop/Greatlearning/GL_mini_project_2/Factor-Hair-Revised.csv", nrows =1))

#remove the column ID before processing for factor analysis.
csv_file= pd.read_csv("C:/Users/acer/Desktop/Greatlearning/GL_mini_project_2/Factor-Hair-Revised.csv", usecols =[i for i in cols
print csv_file.columns

#perform Factor analysis

factor1 = FactorAnalysis(n_components=4,random_state=101).fit_transform(csv_file)

#find the correlation matrix in order to find the eigen values
corr = np.corrcoef(csv_file, rowvar=0)  # correlation matrix

e_val, e_vec = np.linalg.eig(corr)      # eigen values & eigen vectors
print e_val
```

```
Index([u'ProdQual', u'Ecom', u'TechSup', u'CompRes', u'Advertising',
       u'ProdLine', u'SalesFImage', u'ComPricing', u'WartyClaim',
       u'OrdBilling', u'DelSpeed', u'Satisfaction'],
      dtype='object')
[4.04285997 2.5529244  1.69222417 1.21754639 0.63596293 0.56853132
 0.40282774 0.32448016 0.23613948 0.14422355 0.08314143 0.09913845]
```

Form the above output,it is very clear that the first four factors are holding eigen values >1.

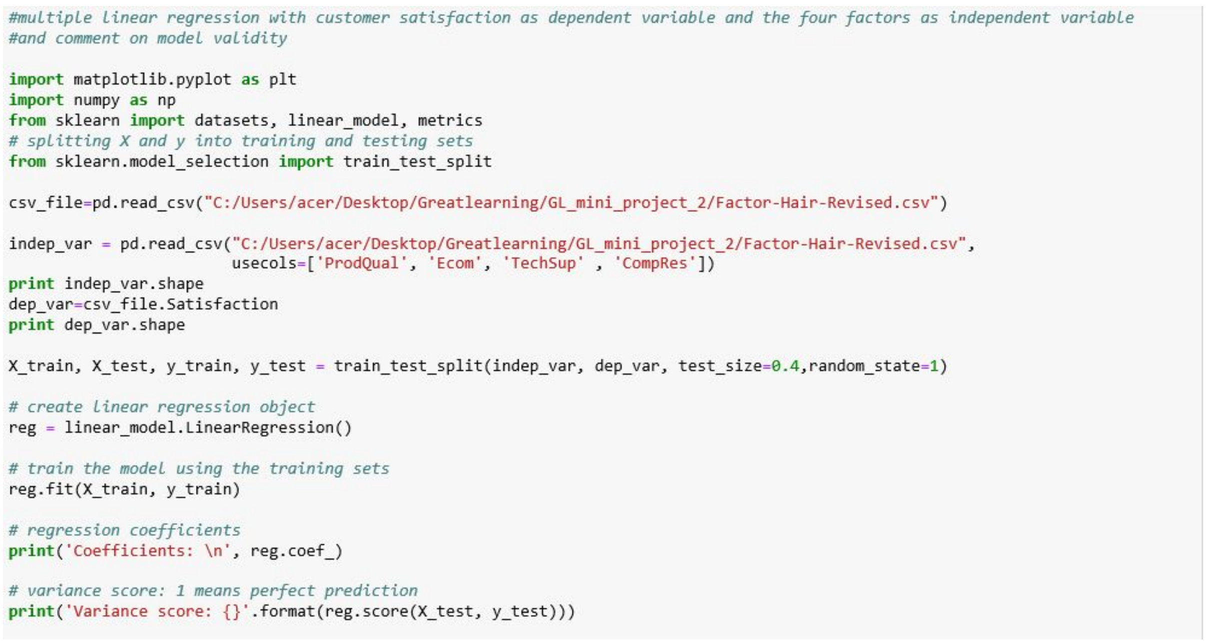
Product Quality,Ecommerce,Technical Support,Complaint resolution are the major four factors to be considered for further analysis

Q3.Name the four factors.

Product Quality,Ecommerce,Technical Support,Complaint resolution are the major four factors to be considered for further analysis

Q4.Perform multiple linear regression with customer satisfaction as dependent variable and the four factors as independent variable and comment on model validity.

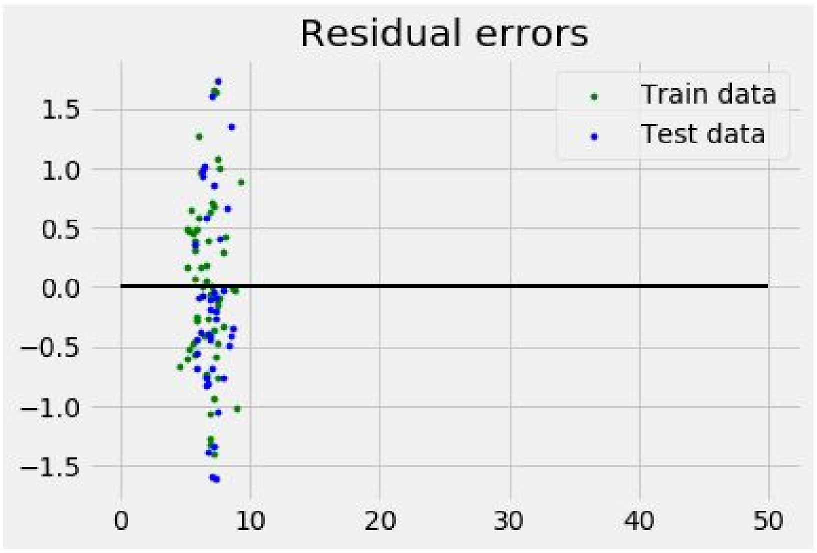
Multiple linear regression attempts to model the relationship between **two or more features** and a response by fitting a linear equation to observed data.



```
#multiple linear regression with customer satisfaction as dependent variable and the four factors as independent variable
#and comment on model validity

import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model, metrics
# splitting X and y into training and testing sets
from sklearn.model_selection import train_test_split

csv_file=pd.read_csv("C:/Users/acer/Desktop/Greatlearning/GL_mini_project_2/Factor-Hair-Revised.csv")

indep_var = pd.read_csv("C:/Users/acer/Desktop/Greatlearning/GL_mini_project_2/Factor-Hair-Revised.csv",
                        usecols=['ProdQual', 'Ecom', 'TechSup' , 'CompRes'])
print indep_var.shape
dep_var=csv_file.Satisfaction
print dep_var.shape

X_train, X_test, y_train, y_test = train_test_split(indep_var, dep_var, test_size=0.4,random_state=1)

# create linear regression object
reg = linear_model.LinearRegression()

# train the model using the training sets
reg.fit(X_train, y_train)

# regression coefficients
print('Coefficients: \n', reg.coef_)

# variance score: 1 means perfect prediction
print('Variance score: {}'.format(reg.score(X_test, y_test)))
```

```
# plot for residual error

## setting plot style
plt.style.use('fivethirtyeight')

## plotting residual errors in training data
plt.scatter(reg.predict(X_train), reg.predict(X_train) - y_train,color = "green", s = 10, label = 'Train data')

## plotting residual errors in test data
plt.scatter(reg.predict(X_test), reg.predict(X_test) - y_test,color = "blue", s = 10, label = 'Test data')

## plotting line for zero residual error
plt.hlines(y = 0, xmin = 0, xmax = 50, linewidth = 2)

## plotting legend
plt.legend(loc = 'upper right')

## plot title
plt.title("Residual errors")

## function to show plot
plt.show()
```

```
(100, 4)
(100,)
('Coefficients: \n', array([0.3585701 , 0.5627802 , 0.00892802, 0.50939761]))
Variance score: 0.422960456991
```



From the graph output,the relationship between the variables are Non-Linear. It is assumed that there is little multicollinearity in the data. Multicollinearity occurs when the features (or independent variables) are not independent from each other

Model validity:

Model validity is the process of validating the model with respect to real system.Here are the steps involved in cross validation:

1. You *reserve* a sample data set
2. Train the model using the remaining part of the dataset
3. Use the reserve sample of the test (validation) set.If your model delivers a positive result on validation data, go ahead with the current model.

The most popular methods of validations are **K-Folds Cross Validation** and **Leave One Out Cross Validation** (LOOCV).