Case Study_Human Resourse Dataset

- **Human_Resources.csv Analysis**
- **Apply K mean Clustering**
- **Apply PCA**
- **Apply Autoencoder**

Task 1:Import your libraries (Lab 2)

```
#Import the libraries here
```

```
#Attach the Human_Resource.csv file and view the first five records
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	Relation
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	

5 rows × 35 columns

```
# show all the file data types
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

```
# Show the following basic statistics
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	1470.000000	1
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156	2.729932	
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428	0.711561	
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000	1.000000	
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000	2.000000	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	3.000000	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000	3.000000	
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000	4.000000	

8 rows × 26 columns

Task 2: Visualize Dataset (Lab 2)

```
# Replace 'Attritition','Overtime' and 'Over18' columns with integers before performing any visualizations
```

```
# display the current first four records
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	Relation
0	41	1	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	0	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	1	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	0	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	

4 rows × 35 columns

```
# Drop EmployeeNumber',EmployeeCount' ,'Standardhours' and 'Over18' since they do not change from one employee to the other
```

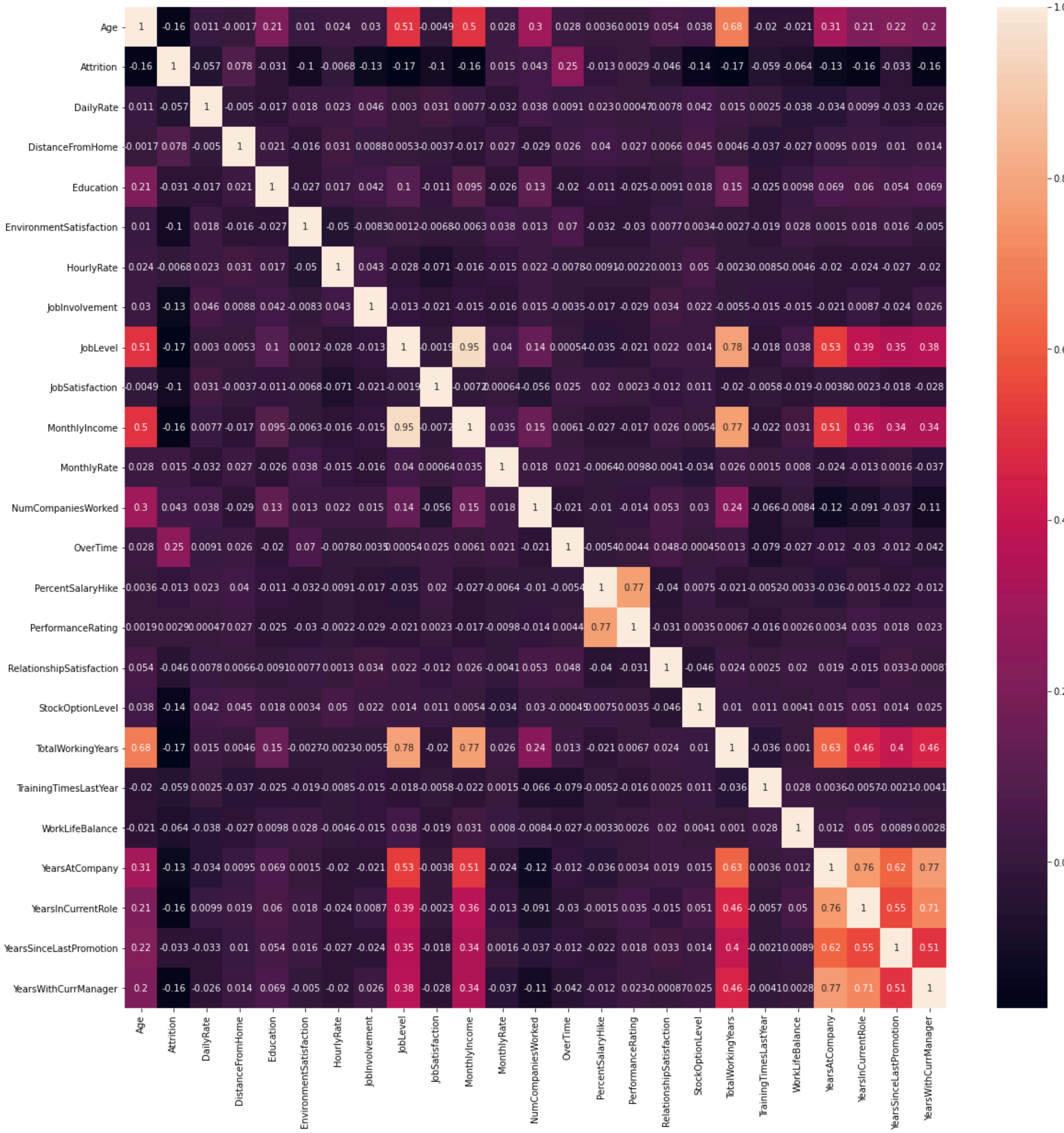
```
# Let's see how many employees left the company!
left_df        = employee_df[employee_df['Attrition'] == 1]
stayed_df      = employee_df[employee_df['Attrition'] == 0]
```

```
# Count the number of employees who stayed and left
# It seems that we are dealing with an imbalanced dataset
```

```
Total = 1470
Number of employees who left the company = 237
Percentage of employees who left the company = 16.122448979591837 %
Number of employees who did not leave the company (stayed) = 1233
Percentage of employees who did not leave the company (stayed) = 83.87755102040816 %
```

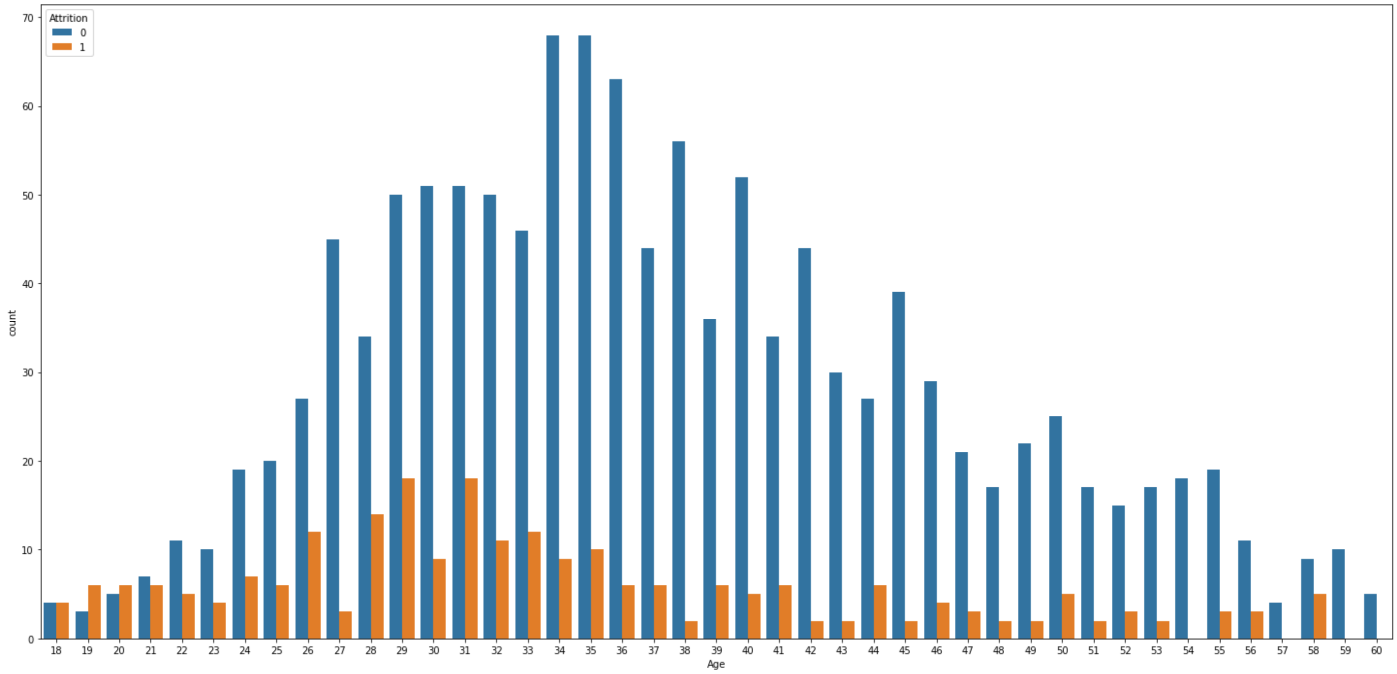
```
# show the correlation heat map as below
```

```
<AxesSubplot:>
```



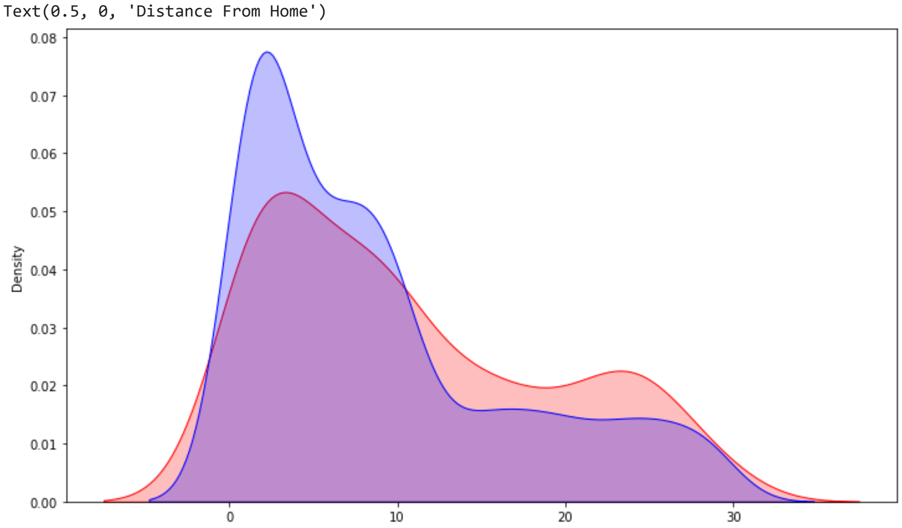

```
# Display the below visualization with hue as Attrition
```

<AxesSubplot:xlabel='Age', ylabel='count'>



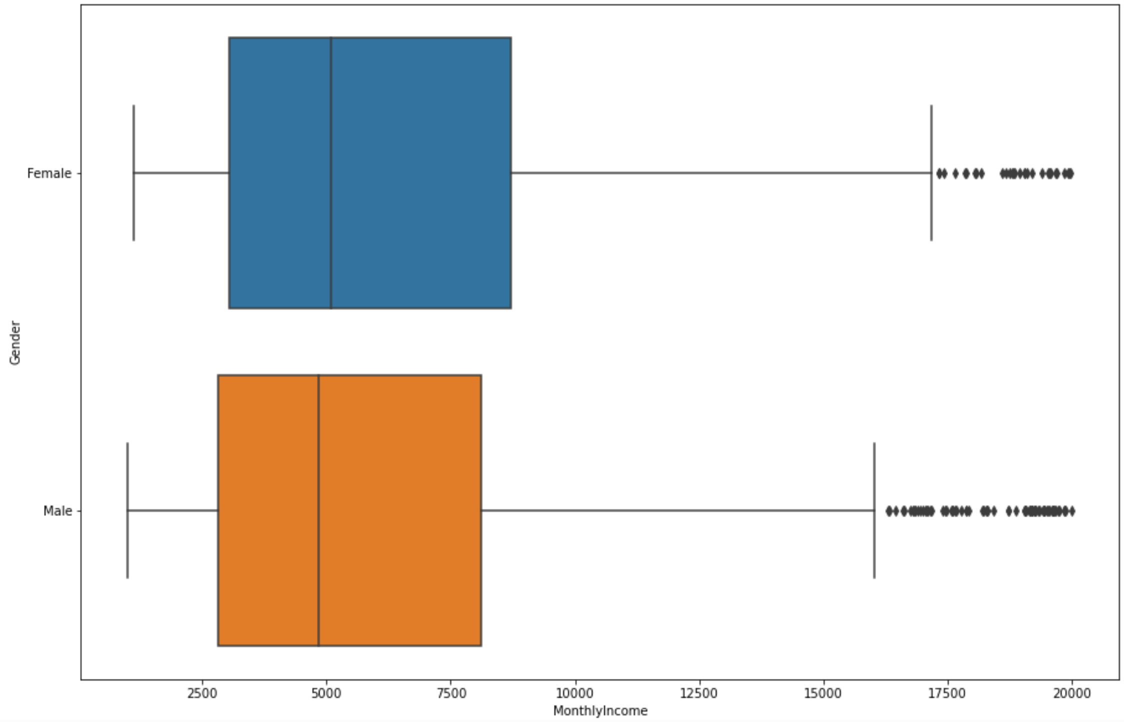

```
# create a Kernel Density Estimate comparing 'Employees who left' and 'Employees who Stayed' using 'Distance From Home'
plt.figure(figsize=(12,7))
```

Text(0.5, 0, 'Distance From Home')




```
# Let's see the Gender vs. Monthly Income using box plots
```

<AxesSubplot:xlabel='MonthlyIncome', ylabel='Gender'>



Task 3: Create Testing and Training Dataset & Perform Data Cleaning (Lab 2)

```
# Convert the categorical fields into numerics using OneHotEncoder
```

```
# select your features here i.e. drop the target 'Atrittion'
```

	0	1	2	3	4	5	6	7	8	9	...	PerformanceRating	RelationshipSatisfaction	StockOptionLevel	TotalWorkingYears	TrainingTim
0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	...	3	1	0	8	
1	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	...	4	4	1	10	
2	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	3	2	0	7	
3	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	...	3	3	0	8	
4	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	3	4	1	6	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1465	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	3	3	1	17	
1466	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	3	1	1	9	
1467	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	...	4	2	1	6	
1468	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	...	3	4	0	17	
1469	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	3	1	0	6	

1470 rows × 50 columns

```
# scale your features data assigning it variable X
```

```
X
```

```
array([[0.        , 0.        , 1.        , ..., 0.22222222, 0.        ,
        0.29411765],
       [0.        , 1.        , 0.        , ..., 0.38888889, 0.06666667,
        0.41176471],
       [0.        , 0.        , 1.        , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.        , 0.        , 1.        , ..., 0.11111111, 0.        ,
        0.17647059],
       [0.        , 1.        , 0.        , ..., 0.33333333, 0.        ,
        0.47058824],
```

```
       [0.        , 0.        , 1.        , ..., 0.16666667, 0.06666667,
        0.11764706]])
```

```
# select your dependent, target or response data as "Attrition" using variable y
```

```
y
```

```
0       1
1       0
2       1
3       0
4       0
       ..
1465    0
1466    0
1467    0
1468    0
1469    0
Name: Attrition, Length: 1470, dtype: int64
```

Task 4: Find the Optimal Number of Clusters using Elblow Method (Lab 2)

```
# Compute 'within cluster sum of squares' or WCSS metric for a range of k clusters
```

```
# Create a visualization for Finding the right number of clusters - Elbow method'
```

Task 5: Apply K-Means Clustering (Lab 2)

```
# Check size of each cluster - Are they all representative ?
```

Task 6: Apply PCA and Visualize Results (Lab 3)

```
# Obtain the principal components
```

```
# All samples projected on the two principal components
```

```
# Create a dataframe with the two components
```

```
# Concatenate the clusters labels to the dataframe
```

```
# Create a scatterplot visual of Projection of the dataset on the 2 PCA dimensions'
```

```
# show the % of the total variance explained by each principal component. Overall close to 48% explained by these two.
```

Task 7: Perform Dimensionality Reduction using Autoencoders (Lab 3)

```
#import the autoencoder libraries
```

```
# create your autoencoder with all the features showing Encoder, bottleneck, decoder, autoencoder
# compile the autoencoder using optimizer='adam', loss='mean_squared_error'
```

```
# show the autoencoder summary
```

```
## Train autoencoder using input = output
```

```
# Use Autoencoder to reduce the number of features / dimensions and show the dimensions
```

Task 8: Apply KMEANS to encoded dataset (Lab 3)

```
# Apply KMEANS to encoded dataset here
```

```
# create a line plot to show the " Pick optimal number of clusters using Elbow method" of the unreduced and reduced dimension Kmeans features
```

```
## Apply the resulting optimal k to find new centroids
```

```
## Show the centroids shape
```

```
# show the clusters shape
```

```
# concatenate the clusters to the data
```

```
# show the 'Number of samples" in your current consolidated
```