This Jupyter Notebook is pepared by Brantely Deines

1 - Load Data and Perform Basic EDA

I - Imopri Libraries

```
import numpy as np
import pandas as pd
import matplotlib as mpt
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msn
from sklearn.preprocessing import StandardScaler
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
    Drive already mounted at /content/drive; to attempt to forcibly remount, call
```

II - Import Data and Show Count of Rows and Rolumns

```
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/hrdata3.csv', index_col
data.shape
```

```
    (12977, 7)
```

III - Show First and Last 5 rows

```
data.head()
```

	enrollee_id	city_development_index	experience	company_size	last_new_jo
1	29725	0.776	15	2	5
4	666	0.767	21	2	4
6	28806	0.920	5	2	1

✓ 1s completed at 6:24 PM ● ×

8	27107	0.920	7	2	1

```
data.tail()
```

	enrollee_id	city_development_index	experience	company_size	last_ne
19149	251	0.920	9	2	
19150	32313	0.920	10	3	
19152	29754	0.920	7	1	
19155	24576	0.920	21	2	
19156	5756	0.802	0	4	

IV - Show Count of Null Values

```
data.isnull().sum().sort_values(ascending = False)
```

```
enrollee_id               0
city_development_index    0
experience                0
company_size              0
last_new_job              0
training_hours            0
target                    0
dtype: int64
```

V - Ensure All Columns are Numeric

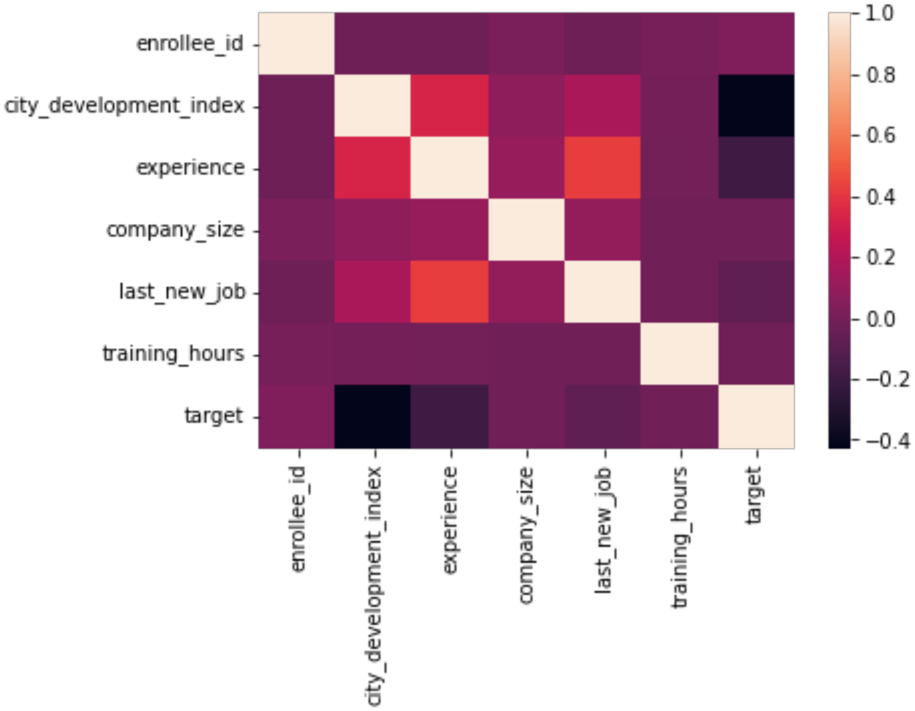
```
data.dtypes
```

```
enrollee_id                 int64
city_development_index    float64
experience                  int64
company_size                int64
last_new_job                int64
training_hours              int64
target                    float64
dtype: object
```

VI - Plot Heatmap

```
correlation = data.corr()
sns.heatmap(correlation)
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7f63d1396ad0>
```



2 - Feature Selection and Pre-Processing

I - Put All Data Except 'enrollee_id' and 'target' into X

[ ] ↳ *1 cell hidden*

II - Scale X with StandardScaler and Show Sample Data

[ ] ↳ *1 cell hidden*

3 - KMeans Clustering

I - Import Related Libraries and Perform KMeans on X W/ random_state = 47

[ ] ↳ *2 cells hidden*

II - Show Cluster Centers as is Then Inverse and Show Again.

Explain The Centers as Related to the Columns of the Data Set

```
modelScale.cluster_centers_
```

```
    array([[-0.31364817, -0.63941844, -0.15207764, -0.55076921,  0.01170319],
           [ 0.44177356,  0.90062111,  0.21420141,  0.77575864, -0.01648395]])
```

```
scaler.inverse_transform(modelScale.cluster_centers_)
```

```
    array([[ 0.7987756 ,  6.62300698,  2.9252866 ,  1.32309922, 66.4026881 ],
           [ 0.88943708, 16.75426875,  3.72717149,  3.49907201, 64.69413512]])
```

Double-click (or enter) to edit

III - Show Distance Matrix

[ ] ↳ *1 cell hidden*

IV - Show Labels

[ ] ↳ *1 cell hidden*

V - Add 'cluster_label' Column to the DataFrame and Assign it the KMean Label

```
scaledX['cluster_label'] = pd.Series(modelScale.labels_, index=data.index)
X['cluster_label'] = pd.Series(modelScale.labels_, index=data.index)
```

VI - Add 'target' Column as a Floating-Point Value

Map the 'target' Column to a 'target_int' Column

```
scaledX['target'] = data['target'].astype(float)
scaledX['target_int'] = scaledX['target'].astype(int)

X['target'] = data['target'].astype(float)
X['target_int'] = X['target'].astype(int)
```

VII - Show Top 5 Rows of DataFrame

```
scaledX.head(10)
```

	city_development_index	experience	company_size	last_new_job	training_
1	-0.503422	0.633957	-0.574723	1.690762	-0.30
4	-0.578413	1.546009	-0.574723	1.081137	-0.95
6	0.696434	-0.886130	-0.574723	-0.747739	-0.68
7	-0.620075	0.329940	-1.488268	1.690762	-0.78
8	0.696434	-0.582112	-0.574723	-0.747739	-0.32
9	0.696434	0.937974	1.709140	1.690762	0.94
11	0.696434	-0.886130	1.252367	-0.747739	0.69
12	0.638108	1.546009	0.795595	0.471512	-0.70
15	0.746428	0.785966	-1.031496	1.690762	-0.78
16	0.696434	-1.494164	-0.574723	-1.357364	0.66

VIII - Print Confusion Matrix Comparing 'target_int' and 'cluster_label' Show classification report and Total Misclasifications

```
confusion_matrix(scaledX['target_int'], scaledX['cluster_label'])
```

```
    array([[5835, 4860],
           [1747,  535]])
```

IX - Discuss Numbers From 3 - VIII

↳ 1 cell hidden

X - Show Inertial of the Cluster

[ ] *↳ 1 cell hidden*

XI - What is the Elbow Method and What is it's Purpose is KMeans

↳ 1 cell hidden

XII - Plot the Inertias for KMeans With Clusters of 2 - 20

[ ] *↳ 2 cells hidden*

XIII - Show Scatter Plot with 'training_hours' Vs 'experience' with the points colored based on 'cluster_label'. Write thoughts on the plot

[ ] *↳ 3 cells hidden*

XIV - Show Scatter Plot with Any Other 2 Attributes Similar to 3 - XIII. Write Thoughts on the Plot

[ ] *↳ 2 cells hidden*

4 - AgglomerativeClustering

```
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering
```

I - Plot Dendogram

[ ] ↳ *1 cell hidden*

II - Perform AgglrmerativeClustering with 2 Clusters Using Euclidean Distarge fro Affinity and Linkage = 'ward'

[ ] ↳ *1 cell hidden*

III - Plot 'training_hours' Vs 'experience'. Write Toughts on the Plot

[ ] ↳ *2 cells hidden*

IV - Increase Clusters to 4 or 5 and Build Clusters Again. Plot Them to See Difference

[ ] ↳ *3 cells hidden*