Factors Effecting Employee Work Life Balance

Experiential Learning Assignment

Advanced Statistical Methods (PDBAZG53)

Submitted by:

Group 38

Group Members

Sr No	BITS ID	Name
1	2023PD21505	Isha Somani
2	2023PD21506	Niharyka Singh
3	2023PD21522	Sudeep B Raj
4	2023PD21524	Adithyan Kv

Table of Contents

PHASE 1

Step 1: Data Collection and Preparation

Problem Statement

This research aims to investigate the multifaceted influences of various factors such as *job role, industry type, years of experience, income level,* and other pertinent variables on employee *work-life balance*. By analyzing these variables, the study seeks to uncover patterns and correlations that shed light on how different aspects of employment impact the ability of individuals to maintain a harmonious equilibrium between their professional responsibilities and personal life. Through a comprehensive examination of these factors, the research endeavors to provide valuable insights.

Structured questionnaire

We have prepared a structured questionnaire to collect the responses from people and collected 125 responses. Questionnaire can be access using below link.

https://forms.gle/wMatwUXS3JR7x9216

Sl Nbr	Demographic Variables	Variable Type
1	Gender	Nominal
2	Working City	Nominal
3	Marriage Status/ kids	Nominal
4	Annual Salary	Ordinal
5	Age	Scale - Continuous

Data Collection

We were able to collect 125 responses from diverse group of working professionals, ranging from different age group and geographies. The below link contains the unaltered data collected from responses.

https://docs.google.com/spreadsheets/d/1lksrlmRS-qFSTqLrKjuNZUNzDQuVMPz6/edit?usp=sharing&ouid=113963560297463239511&rtpof=true&sd=true

Data Preprocessing

Since there were some data inconsistencies observed in the responses, below steps were taken for cleaning the data.

- Removed observations where City was out of India or had garbage value
- Where ever Avg time to commute per day one way was more than 3 hours, we divided it by number of working days, assuming people had put it for the week

- Where ever frequency was less for the values present in the Working City, Type of Industry, Job Profile, it was tagged as Others to reduce number of levels.
- Some people had entered actual and designated working hours in days instead of week, It was taken care of by multiplying the fields with working days when <15 hours
- Calculated average working hours per week by dividing actual hours by designated hours

The below link contains data post handling data inconsistencies. It also contains the descriptive statistics for the cleaned data.

https://docs.google.com/spreadsheets/d/1_eN3wFv4o1-hPQrW7jUAXo_oQihUjBO9/edit?usp=sharing&ouid=113963560297463239511&rtpof=true&sd=true

ASM_Assignment1_Responses.xlsx

Step 2: Descriptive Analysis

For Continuous Variables

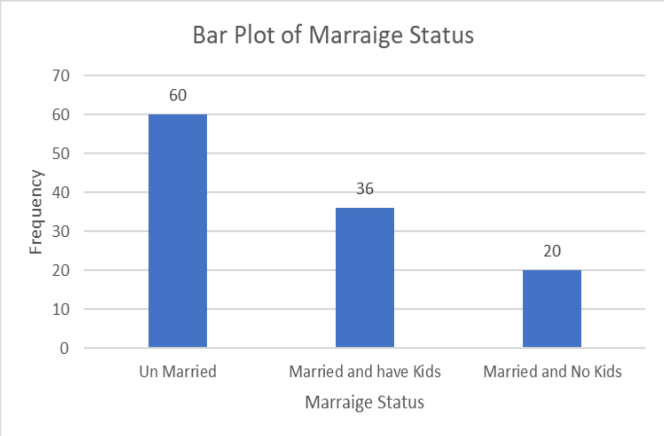
There are 9 Continuous Variables.

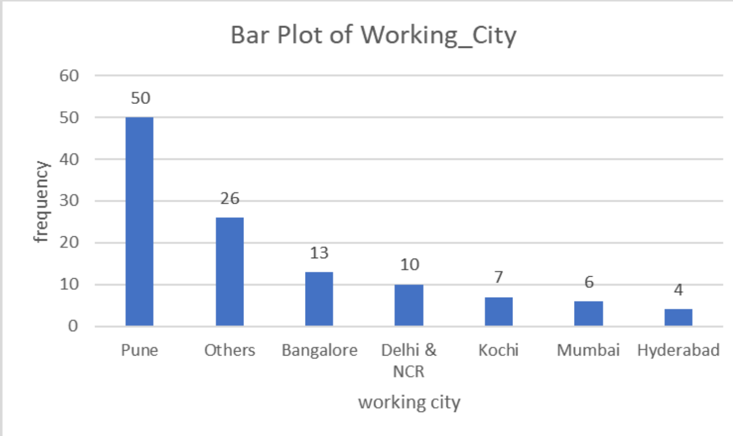
Variables	N_uniqu	mean	std	min	25%	50%	75%	max	Range	IQR	skew	kurt	Skewness	Kurtosis
Please Mention your age in y	23	30.44	5.83	24	27	28.5	33	53	29	6	1.63	3.00	High	mesokurtic
Work Experience (in years)	23	8.56	6.11	1	4	7	11	29	28	7	1.40	1.95	High	platykurtic
clnd_Average Time to Comm	23	43.48	33.67	0	20	30	60	180	180	40	1.17	1.70	High	platykurtic
Number of total paid leave (I	38	27.77	11.92	0	21	29	33.25	60	60	12.25	-0.01	0.78	Symmetric	platykurtic
day_hrs_per_week	3	5.85	0.34	5	6	6	6	6	1	0	-2.02	2.27	High	platykurtic
clnd_Actual Work hours per	25	50.23	14.86	15	44	48	54.25	168	153	10.25	4.55	34.17	High	leptokurtic
Clnd_Designated work hours	18	45.37	7.78	15	40	45	48	72	57	8	0.96	5.65	moderatel	leptokurtic
Average working hours	35	1.32	0.85	0.69	1	1.05	1.25	5	4.31	0.25	3.64	12.76	High	leptokurtic
GTE_Fair_Worklife_balance	2	0.86	0.35	0	1	1	1	1	1	0	-2.13	2.57	High	leptokurtic

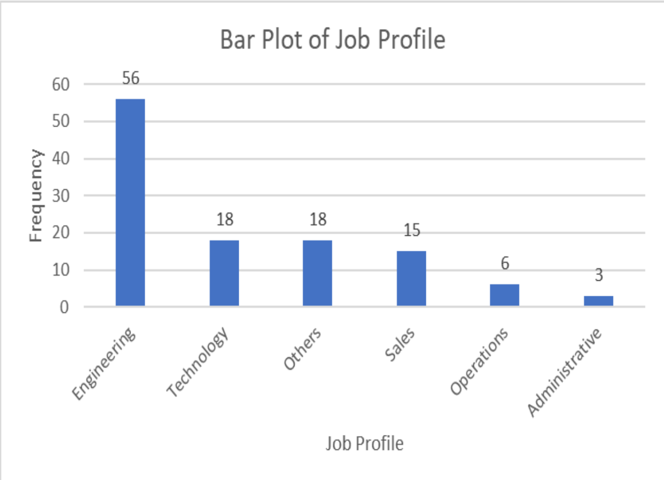
For Categorical Variables

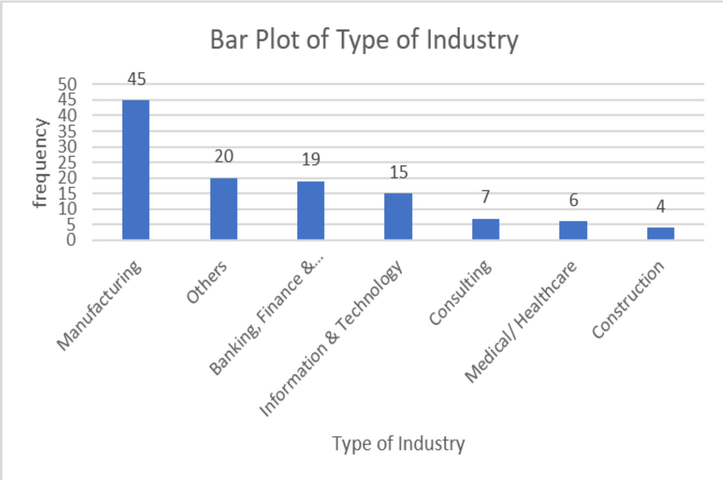
Variables	N_unique	Mode	freq _of_Mode
Gender	2	Male	95
clnd_Working_City	7	Pune	50
Marriage status/ Kids	3	Un Married	60
Annual Salary (in Indian Rupees)	8	5Lacs to 10Lacs	34
clnd_Type of Industry	7	Manufacturing	45
clnd_Job Profile	6	Engineering	56
Work Mode	3	Work From Office (WFO)	66
Employment Type	4	Permanent - Company Payroll	102
Flexibility in office Timings	3	Fixed working hours	51
Working days in Week	3	Mon to Fri or 5 days	72
How is your Work-Life Balance	5	Good	52

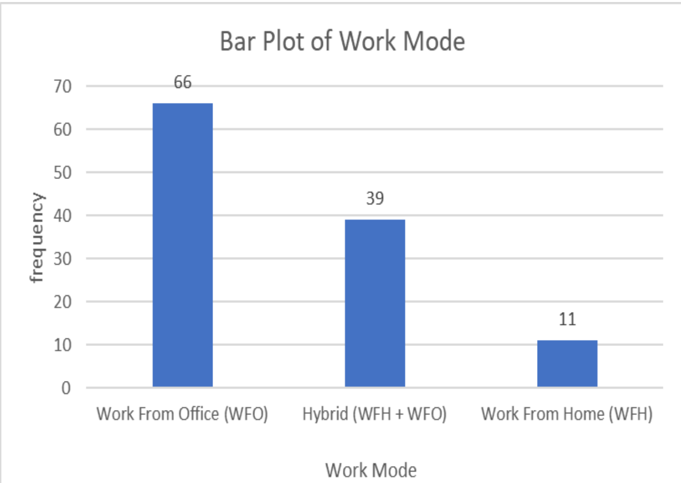
Bar Plots corresponding to Categorical Variables

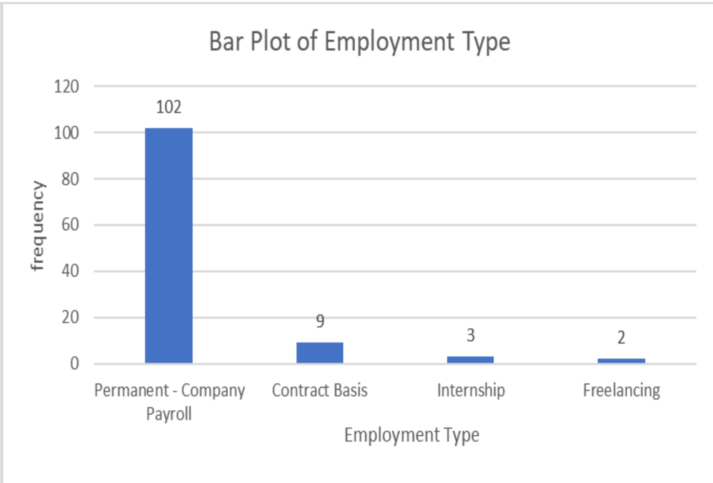


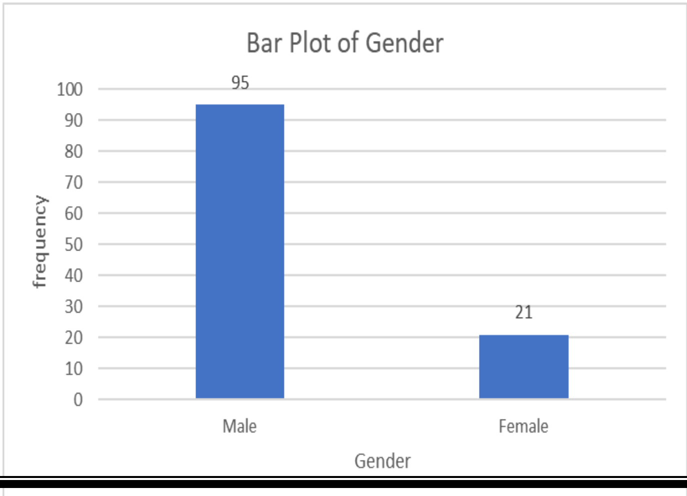


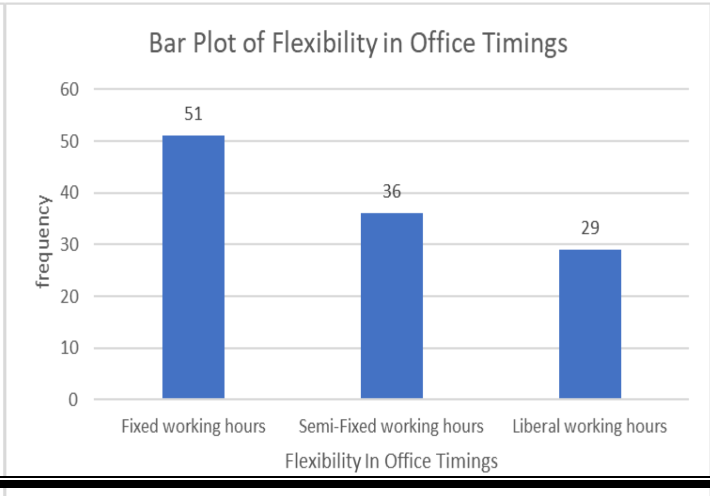




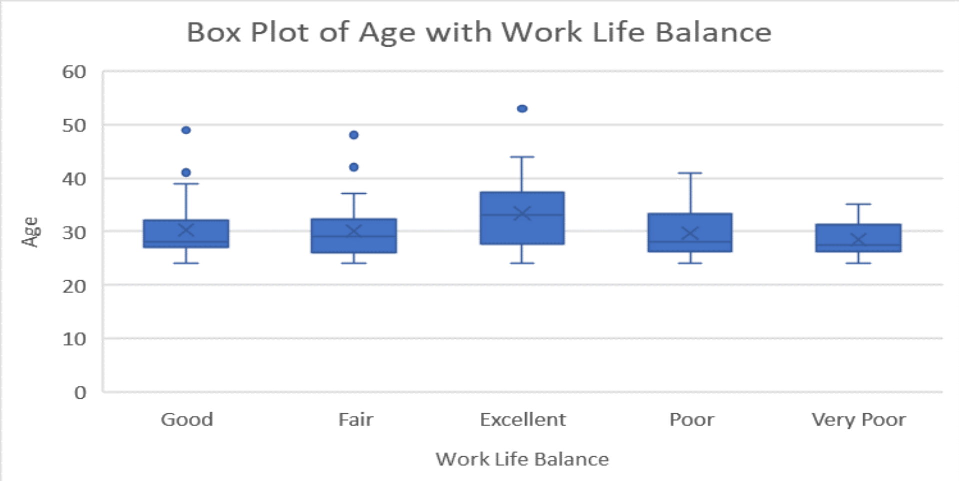




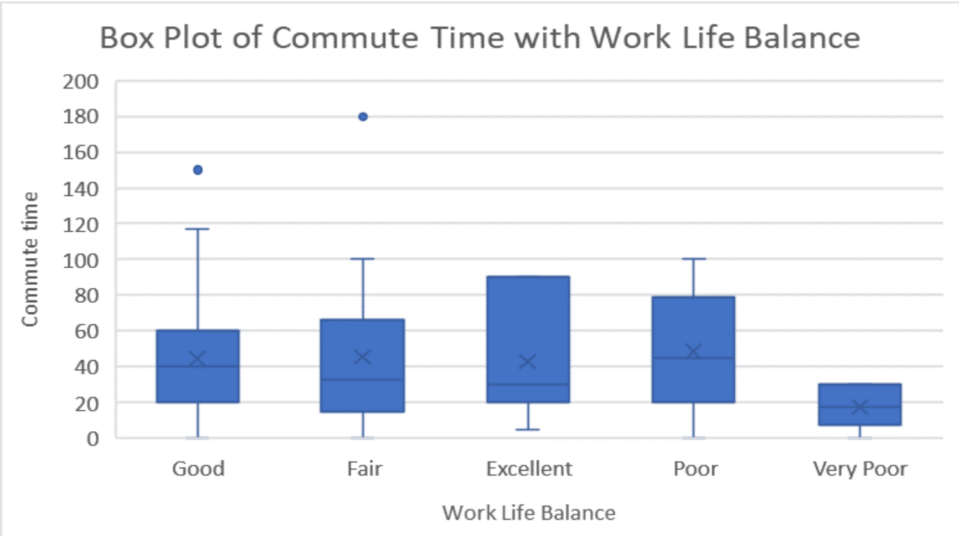




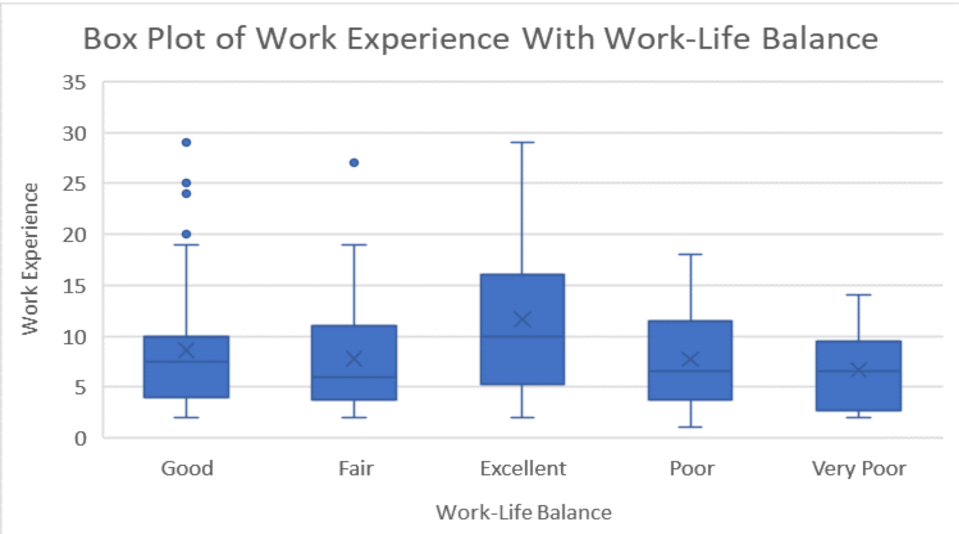
Box Plot of continuous Variables with Work Life Balance



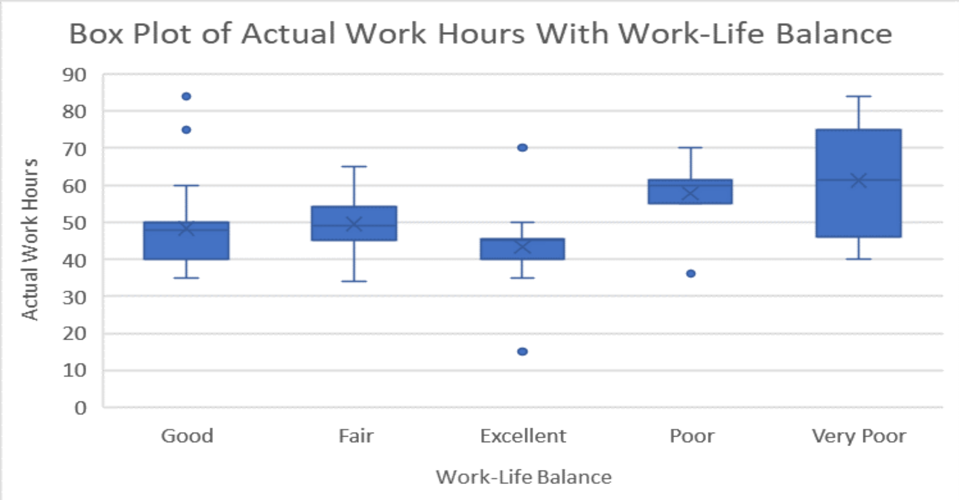
Age means are not differing much across each level of work life balance



Commute time means are quite different across each level of work life balance

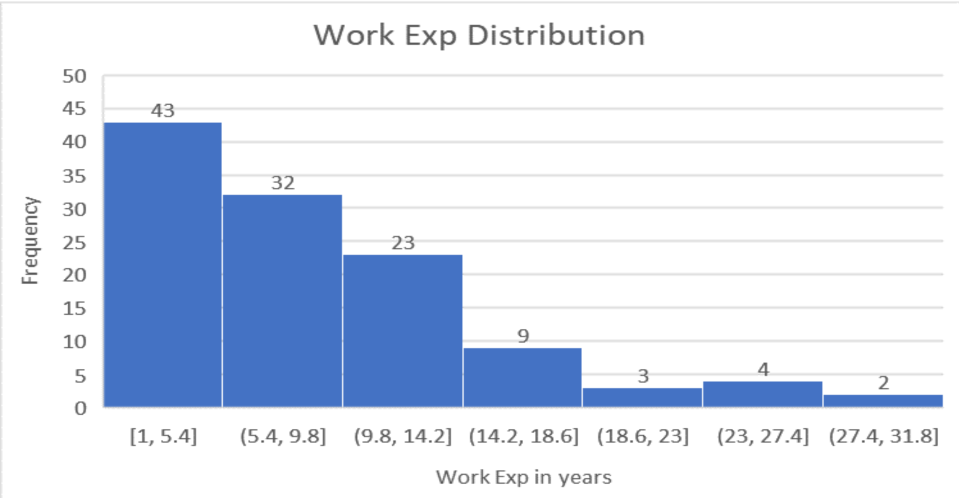


Work Experience means are quite different across each level of work life balance

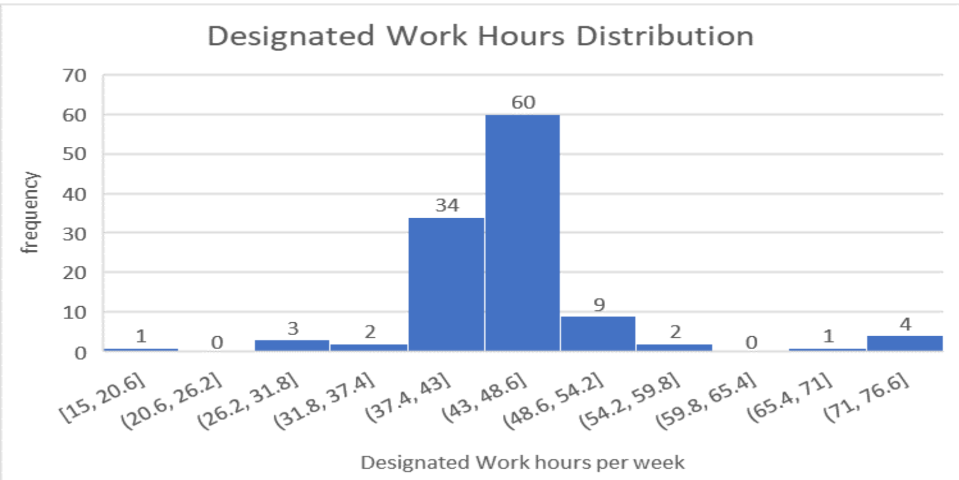


Actual Works Hours means are quite different across each level of work life balance.

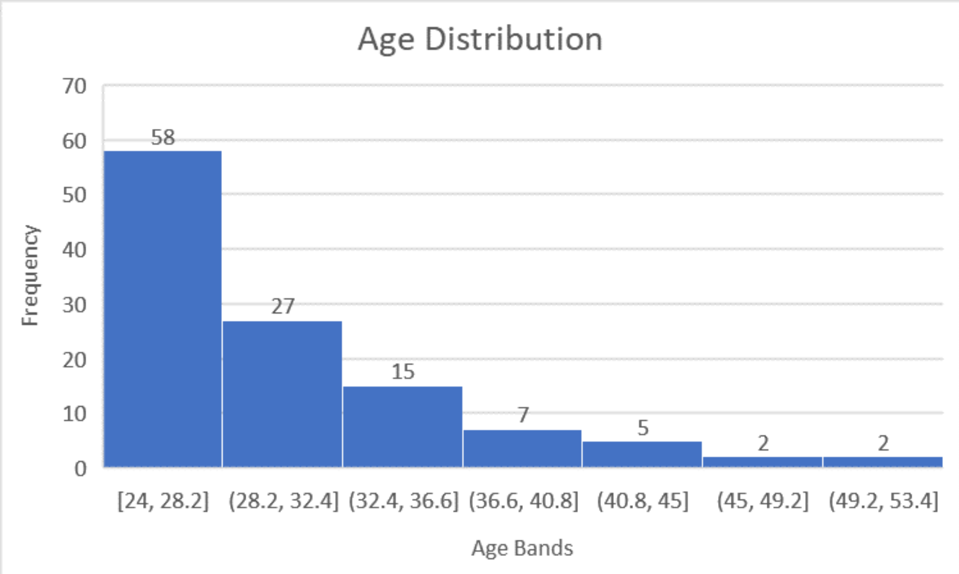
Checking Distribution of Continuous Variables using Histogram



Work Experience in Year is Right Skewed



Designated Work Hour Distribution is Symmetrically distributed.



Age is right skewed distributed.

Hypothesis Testing

Below table contains type of hypotheses and the conclusion of the hypothesis testing.

Variables	Hypotheses	Test Results	Conclusion
Gender (**Nominal**) and work life balance (**Nominal**)	H0: Gender variable is **not** correlated with work life balance. Ha: Gender variable is correlated with work life balance.	CHI Square Test : chi2: 2.3054003895755337 p-value: 0.6797860000954742	The p value > 0.05 hence we fail to reject Null Hypothesis. Gender variable has no correlation with work life balance.
Annual Salary (**Ordinal**) and work life balance (**Nominal**)	H0: Annual Salary variable is **not** correlated with work life balance. Ha: Annual Salary variable is correlated with work life balance.	CHI Square Test : chi2: 28.95897962258727 p-value: 0.41461078807129464	The p value > 0.05 hence we fail to reject Null Hypothesis. Annual Salary variable has no correlation with work life balance.
Age (**Scale**) and Actual Work hours per week (**Scale**)	H0: Age variable is **not** correlated with Actual Work hours per week Ha: Age variable is correlated with Actual Work hours per week.	Pearson Correlation Coefficient = 0.168 P value: 0.7382904764	The p value > 0.05 hence we fail to reject Null Hypothesis. Age variable has no correlation with Actual Work hours per week.

Age (**Scale**) and Work Experience (**Scale**)	H0: Age variable is **not** correlated with Work Experience. Ha: Age variable is correlated with Work Experience.	Pearson Correlation Coefficient = 0.94493288 P value: 0.017628261	The p value < 0.05 hence we fail to reject Null Hypothesis. Age variable is highly correlated with Work Experience.

PHASE 2

Cleaned data of phase 1 is used in phase2.

Step 1: Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction in data analysis. It identifies patterns and correlations within high-dimensional datasets by transforming variables into a new set of orthogonal components, allowing for simplified visualization and analysis while retaining the most important information.

Scree Plot

A scree plot is a graphical tool used in principal component analysis (PCA) to visualize the eigenvalues of components. It helps identify the number of meaningful components to retain in analysis. **6 Principal Components (PCs)** were created to explain the variability in the data.

Variance Explained by 6 PCs

42% of variance is being explained by **6** Principal Components.

```
[28]: var_exp = pd.DataFrame(data=["PC_" + str(i+1) for i in range(n_pca)],columns=["PC"])
      var_exp["variance_explained"] = pca.explained_variance_ratio_
      var_exp["cumm_variance_explained"] = np.cumsum(pca.explained_variance_ratio_)
      var_exp
```

[28]:

	PC	variance_explained	cumm_variance_explained
0	PC_1	0.113779	0.113779
1	PC_2	0.078153	0.191932
2	PC_3	0.064890	0.256821
3	PC_4	0.061743	0.318564
4	PC_5	0.055180	0.373744
5	PC_6	0.045044	0.418787

Factor Loadings

Factor loadings represent the strength of the relationship between observed variables and latent factors in a factor analysis, indicating how much each variable contributes to the underlying constructs. Variable wise factor loadings are present below for each of 6 PCs. **Age and Work Experience** have higher factor loadings.

```
loadings = pd.DataFrame(pca.components_.T * np.sqrt(pca.explained_variance_),columns= ["PC_" + str(i+1) for i in range(n_pca)])
loadings.index = pca_cols
loadings
```

	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6
Please Mention your age in years	-0.774536	0.070142	-0.029179	0.168792	0.153964	0.372815
Work Experience (in years)	-0.802546	-0.034199	-0.058589	0.179494	0.180358	0.327340
clnd_Average Time to Commute to office one way (in minutes)	-0.233754	0.101935	-0.070049	-0.264204	-0.265520	0.229723
Number of total paid leave (Paid + Sick)	-0.440710	0.355155	0.230595	0.065180	0.066627	-0.154948
day_hrs_per_week	-0.057278	-0.537487	0.053791	0.171831	0.250949	0.178965
clnd_Actual Work hours per week (in hours)	-0.259375	-0.255780	0.245169	-0.428111	0.365630	0.002530
Clnd_Designated work hours per week that a person is expected to work (In hours)	-0.192778	-0.279057	0.282811	-0.397469	0.239570	0.044846
Actual_work_hours_per_desogmated_work_hours	0.081589	0.019496	-0.185803	0.027717	0.513712	-0.034392
Good_Worklife	-0.074914	0.083400	0.277759	0.167569	-0.471702	0.062484
enc_salary	-0.411803	0.529387	-0.117286	-0.017671	0.266859	0.195151
Gender_Female	0.544748	0.166406	0.290716	0.537841	0.055887	-0.075928
Gender_Male	-0.544748	-0.166406	-0.290716	-0.537841	-0.055887	0.075928

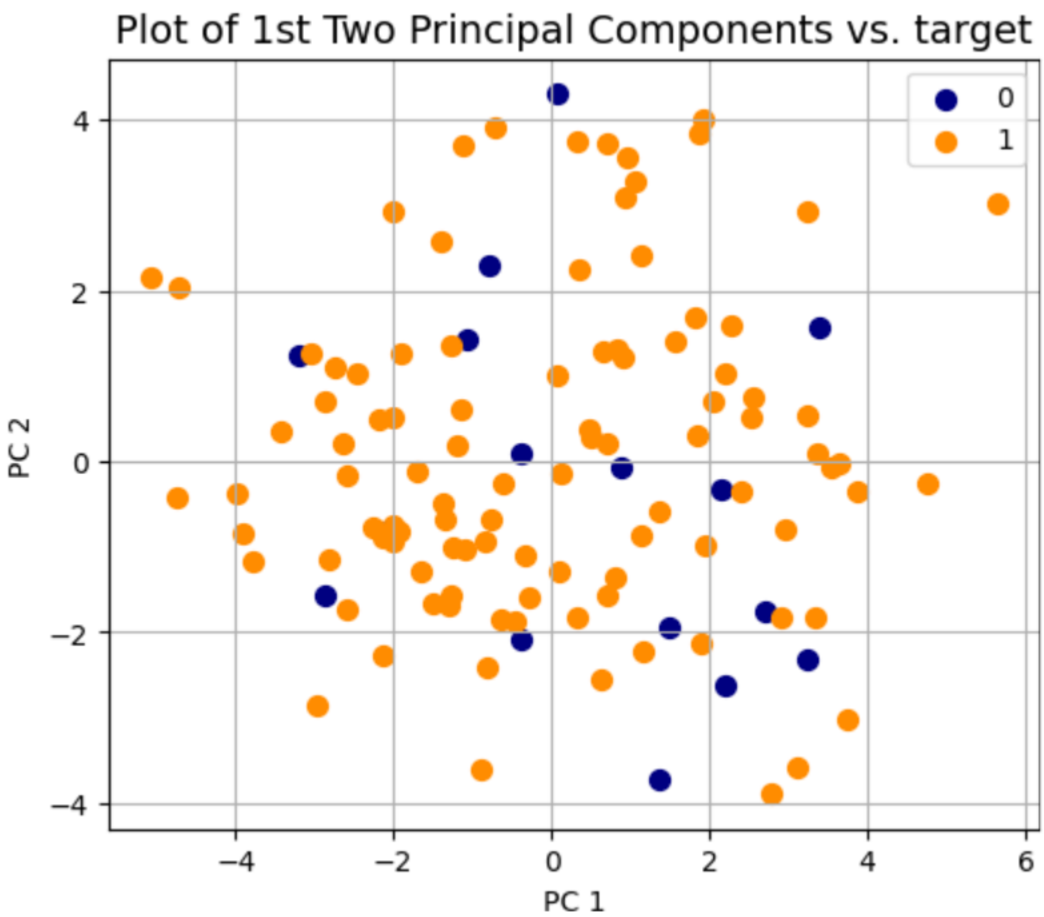
Contributions of The Variable %

Variable wise contribution for each of the PCs are as follows, only top few variables are shown. **Age and Work Experience** are top 2 variables contributing in Principal Component 1.

```
dff = np.round(100*abs(loadings)/abs(loadings).sum(),2)
dff.head(12)
```

	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6
Please Mention your age in years	6.44	0.73	0.29	1.80	1.84	4.61
Work Experience (in years)	6.67	0.36	0.58	1.92	2.15	4.05
clnd_Average Time to Commute to office one way (in minutes)	1.94	1.06	0.69	2.82	3.17	2.84
Number of total paid leave (Paid + Sick)	3.66	3.69	2.28	0.70	0.80	1.92
day_hrs_per_week	0.48	5.59	0.53	1.84	2.99	2.21
clnd_Actual Work hours per week (in hours)	2.16	2.66	2.43	4.57	4.36	0.03
Clnd_Designated work hours per week that a person is expected to work (In hours)	1.60	2.90	2.80	4.25	2.86	0.55
Actual_work_hours_per_desogmated_work_hours	0.68	0.20	1.84	0.30	6.13	0.43
Good_Worklife	0.62	0.87	2.75	1.79	5.63	0.77

Plot of PC1 vs PC2



Step 2: Regression Analysis

Data Summary:

The data post cleaning in phase-1 had 116 rows and 21 columns
The data has 8 categorical columns and 10 numerical columns, details are present in phase one
Target Definition: Quality of Work Life Balance
Levels present in Target: “Excellent”, “Good”, “Fair”, “Poor”, “Very Poor”

Frequency table of Target	
Work-Life Balance	**Frequency**
Good	52
Fair	34
Excellent	14
Poor	10
Very Poor	6

```
df["How is your Work-Life Balance"].value_counts()
#gives count of each values in a column
```

```
How is your Work-Life Balance
Good         52
Fair         34
Excellent    14
Poor         10
Very Poor     6
Name: count, dtype: int64
```

To convert above multi class problem to binary class, following adjustment has been done
Target = **Good_Worklife = 1** if Work-life balance in [Excellent, Good, Fair]
Target = **Good_Worklife = 0** if Work-life balance in [Poor, Very Poor]

```
df["Good_Worklife"] = np.where(df["How is your Work-Life Balance"].isin(['Excellent','Good','Fair']),1,0)
#  when worklife is good(excellent,good,fair) it is 1 and in very,poor its 0
df["Good_Worklife"].value_counts()
```

```
Good_Worklife
1    100
0     16
Name: count, dtype: int64
```

Feature Summary

```
# for checking data summary
def data_info(df):
    df_info = pd.DataFrame(df.isna().sum(),columns = ['Null_count'])
    df_info['N_unique'] = df_info.index.map(df.nunique())
    df_info['D_types'] = df_info.index.map(df.dtypes)
    df_info['Blank_count'] = df_info.index.map((df=='').sum())
    return df_info
```

```
data_info(df) # We get null cnt,unique, blanks in a single table.
data_info(df).to_csv("smry.csv")
```

Variable Treatment

Attributes	Null_count	N_unique	D_types	Blank_ count	Treatment
Gender	0	2	Nominal	0	One Hot Encoding (OHE)
clnd_Working_City	0	7	Nominal	0	One Hot Encoding (OHE)
Marriage status/ Kids	0	3	Nominal	0	One Hot Encoding (OHE)
Please Mention your age in years	0	23	int64	0	NA
Work Experience (in years)	0	23	int64	0	NA
clnd_Average Time to Commute to office one way (in minutes)	0	23	float64	0	NA
clnd_Type of Industry	0	7	Nominal	0	One Hot Encoding (OHE)
clnd_Job Profile	0	6	Nominal	0	One Hot Encoding (OHE)
Work Mode	0	3	Nominal	0	One Hot Encoding (OHE)
Employment Type	0	4	Nominal	0	One Hot Encoding (OHE)
Flexibility in office Timings	0	3	Nominal	0	One Hot Encoding (OHE)
Number of total paid leave (Paid + Sick)	0	38	int64	0	NA
day_hrs_per_week	0	3	float64	0	NA
clnd_Actual Work hours per week (in hours)	0	24	float64	0	NA
Clnd_Designated work hours per week that a person is expected to work (In hours)	0	18	float64	0	NA
Actual_work_hours_per_desogmated_work_hours	0	35	float64	0	NA
Good_Worklife	0	2	int32	0	NA
Annual Salary (in Indian Rupees)	0	8	Ordinal	0	Label Encoding

What is Label Encoding?

Label encoding is a process in machine learning where categorical data is converted into numerical labels. Each category is assigned a unique integer. It's useful for algorithms that require numerical input and where data is ordinal. Only Salary variable was present as ordinal categorical variable, this variable was numerically coded into numbers

```
#df["Annual Salary (in Indian Rupees)"].unique()
df['enc_salary'] = np.select([
        df["Annual Salary (in Indian Rupees)"] == '0 to 5 Lacs',
        df["Annual Salary (in Indian Rupees)"] == '5Lacs to 10Lacs',
        df["Annual Salary (in Indian Rupees)"] == '10Lacs to 15Lacs',
        df["Annual Salary (in Indian Rupees)"] == '15Lacs to 20Lacs',
        df["Annual Salary (in Indian Rupees)"] == '20Lacs to 25Lacs',
        df["Annual Salary (in Indian Rupees)"] == '25Lacs to 30Lacs',
        df["Annual Salary (in Indian Rupees)"] == '30Lacs to 35Lacs',
        df["Annual Salary (in Indian Rupees)"] == '35Lacs +' ],
        [ 1,2,3,4,5,6,7,8],
        default= -999 )
df.groupby(["enc_salary"])["enc_salary"].count()
```

```
enc_salary
1    23
2    34
3    22
4    15
5    10
6     4
7     2
8     6
Name: enc_salary, dtype: int64
```

What is One Hot Encoding?

One-hot encoding is a technique used in machine learning to represent categorical data numerically. Each category is assigned a unique binary value, with all other values set to zero. This creates a sparse matrix where each column corresponds to a category and only one element per row is set to one.

```
target = "Good_Worklife"
char_cols = list(df.select_dtypes("object").columns)
print(char_cols)
```

```
['Gender', 'clnd_Working_City', 'Marriage status/ Kids', 'clnd_Type of Industry ', 'clnd_Job Profile', 'Work Mode', 'Employment Type', 'Flexibility in of
fice Timings']
```

```
#df[features]
df_ohe = pd.get_dummies(df, columns= char_cols)
df_ohe.shape
df_ohe.head()
```

```
(116, 45)
```

ıc_salary	...	Work Mode_Hybrid (WFH + WFO)	Work Mode_Work From Home (WFH)	Work Mode_Work From Office (WFO)	Employment Type_Contract Basis	Employment Type_Freelancing	Employment Type_Internship	Employment Type_Permanent - Company Payroll	Flexibility in office Timings_Fixed working hours	Flexibility in office Timings_Liberal working hours	Flexibility in office Timings_Semi-Fixed working hours
8	...	False	True	False	False	False	False	True	False	False	True
4	...	False	False	True	False	False	False	True	True	False	False
5	...	False	False	True	False	False	False	True	False	False	True

Final Feature List:

After creating dummy variable using One Hot Encoding, total numbers of variable become 44.

```
features = list(df_ohe.columns)
#features.remove(target)
features = [i for i in features if i != target]
print("Nbr of features after OHE - ", len(features))
print(features)
```

```
Nbr of features after OHE -  44
['Please Mention your age in years', 'Work Experience (in years)', 'clnd_Average Time to Commute to office one way (in minutes)', 'Number of total paid l
eave (Paid + Sick)', 'day_hrs_per_week', 'clnd_Actual Work hours per week (in hours)', 'Clnd_Designated work hours per week that a person is expected to
work (In hours)', 'Actual_work_hours_per_desogmated_work_hours', 'enc_salary', 'Gender_Female', 'Gender_Male', 'clnd_Working_City_Bangalore', 'clnd_Worki
ng_City_Delhi & NCR', 'clnd_Working_City_Hyderabad', 'clnd_Working_City_Kochi', 'clnd_Working_City_Mumbai', 'clnd_Working_City_Others', 'clnd_Working_Cit
y_Pune', 'Marriage status/ Kids_Married and No Kids', 'Marriage status/ Kids_Married and have Kids', 'Marriage status/ Kids_Un Married', 'clnd_Type of In
dustry _Banking, Finance & Insurance', 'clnd_Type of Industry _Construction', 'clnd_Type of Industry _Consulting', 'clnd_Type of Industry _Information &
Technology', 'clnd_Type of Industry _Manufacturing', 'clnd_Type of Industry _Medical/ Healthcare', 'clnd_Type of Industry _Others', 'clnd_Job Profile_Adm
inistrative', 'clnd_Job Profile_Engineering', 'clnd_Job Profile_Operations', 'clnd_Job Profile_Others', 'clnd_Job Profile_Sales', 'clnd_Job Profile_Techn
ology', 'Work Mode_Hybrid (WFH + WFO)', 'Work Mode_Work From Home (WFH)', 'Work Mode_Work From Office (WFO)', 'Employment Type_Contract Basis', 'Employme
nt Type_Freelancing', 'Employment Type_Internship', 'Employment Type_Permanent - Company Payroll', 'Flexibility in office Timings_Fixed working hours',
'Flexibility in office Timings_Liberal working hours', 'Flexibility in office Timings_Semi-Fixed working hours']
```

Final Event Rates

Out of every 100 observations, 14 have target as 0

Event Rate of the Overall data, Train & Test Set ¶

- Since Stratified Sampling is done, Event rate of train and test data will be similar to that of overall data

```
#np.unique (y_train, return_counts=True)
#np.unique (y_train, return_counts=True)
print("Event Rate of full data\n", round(100*df_ohe[target].value_counts(normalize = True),2),"\n")
print("Event Rate of train\n",  round(100*pd.Series(y_train).value_counts(normalize = True),2),"\n")
print("Event Rate of test\n",  round(100*pd.Series(y_train).value_counts(normalize = True),2),"\n")
```

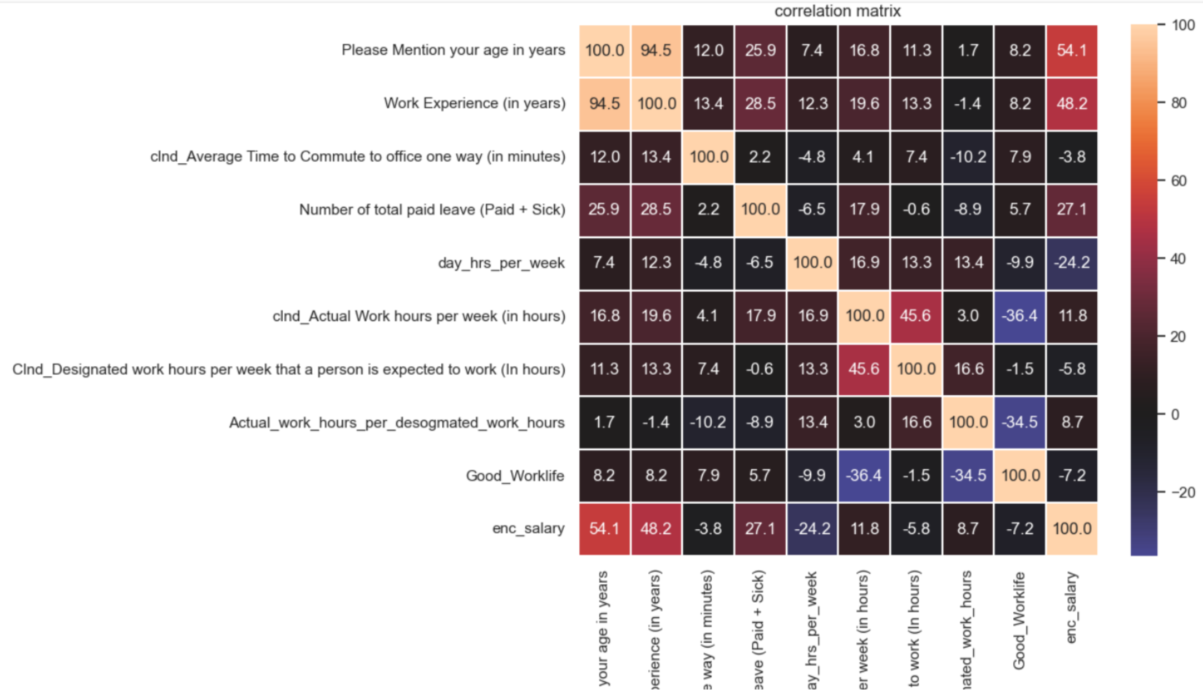
```
Event Rate of full data
 Good_Worklife
1    86.21
0    13.79
Name: proportion, dtype: float64

Event Rate of train
 Good_Worklife
1    86.42
0    13.58
Name: proportion, dtype: float64

Event Rate of test
 Good_Worklife
1    86.42
0    13.58
Name: proportion, dtype: float64
```

Correlation Plot

A correlation plot visually represents the correlation between variables in a dataset, typically using colours or symbols to indicate the strength and direction of relationships. It helps identify patterns and dependencies among variables.

Work Experience and Age are highly correlated variable considering threshold of 70%, rest variables are will within threshold.

```python
cor_mat = df_ohe[df.select_dtypes(np.number).columns].corr()
#https://seaborn.pydata.org/generated/seaborn.heatmap.html
fig, ax = plt.subplots(figsize=(8.5,7)) ;
x = sns.heatmap(100*cor_mat, annot=True,fmt='.1f',center=0,linewidths = 0.25).set_title('correlation matrix')
plt.show()
```



Fitting Logistic Model

Scaling is crucial in logistic regression to ensure variables are on a similar scale, preventing dominance by variables with larger ranges. It enhances model stability, convergence, and interpretation, as coefficients represent the impact of predictors on the outcome consistently, regardless of their original units or magnitudes.

Scaling data helps in identifying feature importance of the data as it brings all the features on same scale, So data is scaled before fitting the model.

[27]:
```python
scaler = StandardScaler()
logreg = LogisticRegression(solver='saga',l1_ratio = 0.75,penalty = "elasticnet")
```

[28]:
```python
#creating and fitting pipe
from sklearn.pipeline import make_pipeline,Pipeline
pipe = Pipeline([('sc', scaler), ('lr', logreg)])
#pipe[0] #pipe[1]
pipe.fit(X_train, y_train)
```

[28]:
```
▸ Pipeline
  ▸ StandardScaler
  ▸ LogisticRegression
```

Classification Report

The model performance is coming to be good even with Regression model as ROC **AUC = 80%.**

Other performance metrics like F1 Score, precision, recall and accuracy are also high.

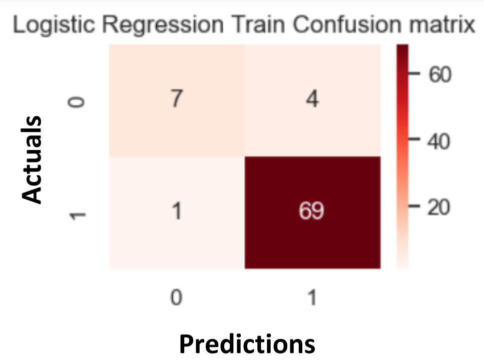
```
estimator= pipe_lr
pred_bin_tr = estimator.predict(X_train[features_n])
pred_bin_te = estimator.predict(X_test[features_n])
#confusion_matrix(y_train,pred_bin_tr)
f'train_rocauc : {round(roc_auc_score(y_train,estimator.predict_proba(X_train[features_n])[:,1]),4)} ; \
test_rocaauc : {round(roc_auc_score(y_test,estimator.predict_proba(X_test[features_n])[:,1]),4)}'
f'train_f1_score : {round(f1_score(y_train,pred_bin_tr),4)} ; test_f1_score : {round(f1_score(y_test,pred_bin_te),4)}'
f'train_precision : {round(precision_score(y_train,pred_bin_tr),4)} ; test_precision : {round(precision_score(y_test,pred_bin_te),4)}'
f'train_recall : {round(recall_score(y_train,pred_bin_tr),4)} ; test_recall : {round(recall_score(y_test,pred_bin_te),4)}'
f'train_accuracy : {round(accuracy_score(y_train,pred_bin_tr),4)} ; test_accuracy : {round(accuracy_score(y_test,pred_bin_te),4)}'
#f'train_conf : {confusion_matrix(y_train,pred_bin_tr)} ; test_conf : {confusion_matrix(y_test,pred_bin_te)}'
```

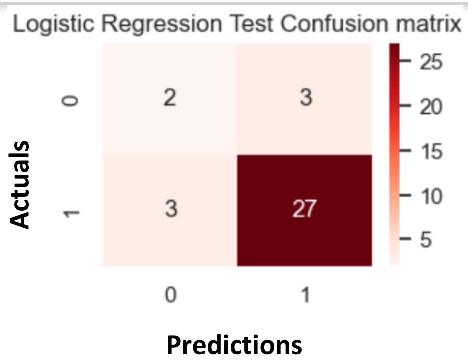
```
'train_rocauc : 0.987 ; test_rocaauc : 0.8067'

'train_f1_score : 0.965 ; test_f1_score : 0.9'

'train_precision : 0.9452 ; test_precision : 0.9'

'train_recall : 0.9857 ; test_recall : 0.9'

'train accuracy : 0.9383 ; test accuracy : 0.8286'
```

Confusion Matrix

Confusion Matrix on Train and Test data is shown below,

```
cf_tr = confusion_matrix(y_train,pred_bin_tr);#cf_tr
cf_te = confusion_matrix(y_test,pred_bin_te);#cf_te
fig, ax = plt.subplots(figsize=(3,2)) ;
x = sns.heatmap(cf_tr, annot=True, cmap='Reds',fmt='.0f',).set_title('Logistic Regression Train Confusion matrix')
plt.show()
fig, ax = plt.subplots(figsize=(3,2)) ;
x = sns.heatmap(cf_te, annot=True, cmap='Reds',fmt='.0f',).set_title('Logistic Regression Confusion matrix')
plt.show()
```





Feature Importance

Below are important features list on scaled variables., Higher the modulus of coefficient, important the feature is. **Top variables are Actual Work Hours per week, Marriage Status,Working City, Salary, age, Hours per week.**

[39]:
```
f'intercept is {pipe[1].intercept_}'
coff = pd.DataFrame({'variable' : df_ohe[features_n].columns, 'coefficient' : pipe_lr[1].coef_.transpose().flatten()})
coff = coff.sort_values(by='coefficient', key=abs,ascending = False).reset_index(drop = True)
coff
```

[39]: 'intercept is [3.0880758]'

[39]:

	variable	coefficient
0	clnd_Actual Work hours per week (in hours)	-1.747912
1	Actual_work_hours_per_desogmated_work_hours	-1.300760
2	Clnd_Designated work hours per week that a per...	1.082415
3	Marriage status/ Kids_Married and No Kids	0.815396
4	clnd_Working_City_Others	0.516046
5	day_hrs_per_week	-0.429300
6	clnd_Working_City_Bangalore	-0.360062
7	enc_salary	0.317954
8	Please Mention your age in years	0.183181

Step 3: Cluster Analysis

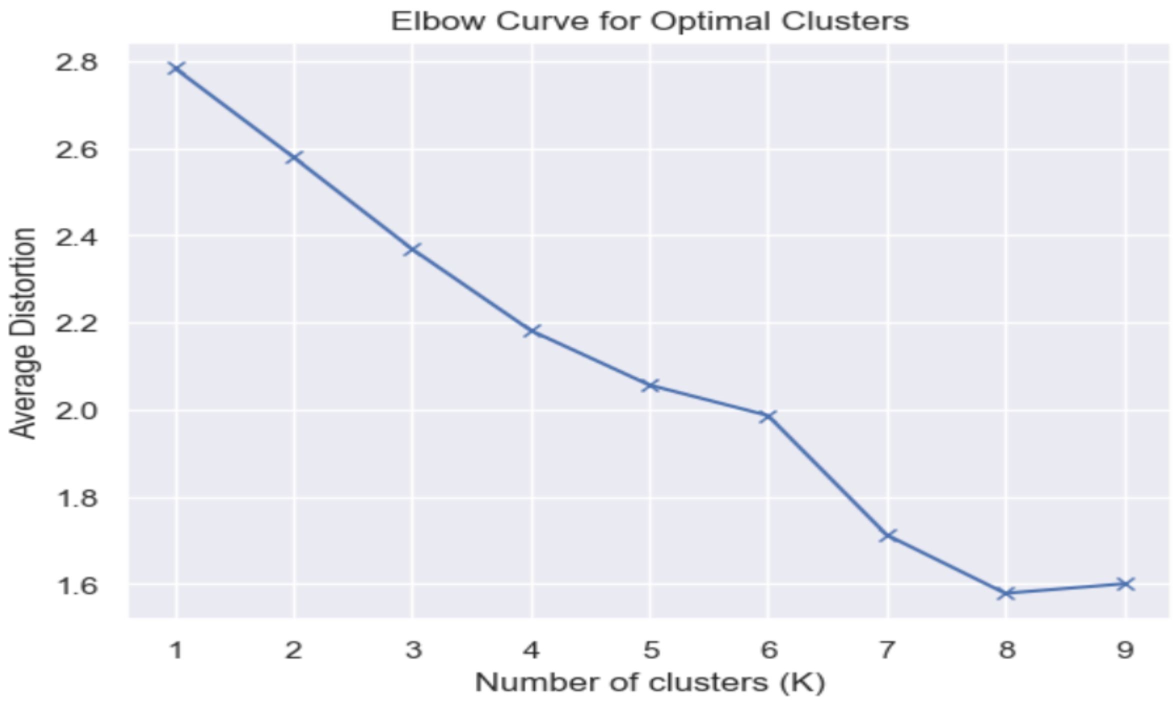
KMeans Clustering:

K-means clustering is a machine learning algorithm that partitions data into 'k' clusters based on similarity, minimizing the within-cluster sum of squares.

Elbow Curve

The elbow curve, in statistics, represents the point where the rate of decrease in variance slows significantly, helping to identify the optimal number of clusters in a k-means clustering algorithm.

From Below plot, number of optimum clusters **K = 6**



Fitting KMeans Model

Fitting KMeans Model with K = 6

```
# it look like 6 and 7 are bends in chart, explore cluster based on this
kmeans_mdl = KMeans(n_clusters=6,random_state = 51)
kmeans_mdl.fit(X_train_scl)
trn_pred = kmeans_mdl.predict(X_train_scl)
tst_pred = kmeans_mdl.predict(X_test_scl)
X_train['kmn_cluster'] = trn_pred
X_test['kmn_cluster'] = tst_pred
```

```
KMeans
KMeans(n_clusters=6, random_state=51)
```

```
X_train['kmn_cluster'] = trn_pred
X_test['kmn_cluster'] = tst_pred
```

Cluster Wise Summary

Various Statistic of Actual Work Hours for each of the 6 clusters show that each clusters have different behavior. Similarly, statistics were calculated for each of the variables, please refer phase 2 HTML file for the same. The statistics for all the variables were showing different trend for the clusters.

```
[39]: for i in features_n:
          print(f"Cluster summary for {i}")
          X_train.groupby(["kmn_cluster"])[i].describe()
```

```
Cluster summary for clnd_Actual Work hours per week (in hours)
```

[39]:

kmn_cluster	count	mean	std	min	25%	50%	75%	max
0	11.0	47.681818	9.675414	35.0	41.0	45.0	52.5	70.0
1	13.0	52.076923	11.235817	40.0	45.0	50.0	54.0	84.0
2	7.0	49.714286	3.545621	44.0	49.0	50.0	50.0	56.0
3	15.0	49.666667	15.262778	34.0	40.0	45.0	57.0	84.0
4	11.0	47.636364	8.090398	40.0	40.0	45.0	54.5	60.0
5	24.0	49.791667	9.784545	40.0	45.0	48.0	51.0	84.0

```
Cluster summary for Actual_work_hours_per_desogmated_work_hours
```

[39]:

kmn_cluster	count	mean	std	min	25%	50%	75%	max
0	11.0	1.157197	0.413300	0.875000	1.000000	1.000000	1.128472	2.333333
1	13.0	1.451068	1.131644	0.833333	1.000000	1.041667	1.155556	5.000000
2	7.0	1.097222	0.091779	1.000000	1.020833	1.111111	1.138889	1.250000
3	15.0	1.581049	1.305476	0.809524	1.000000	1.125000	1.400327	5.000000
4	11.0	1.426263	1.194972	0.833333	1.000000	1.000000	1.211111	5.000000

Cluster Wise Box Plots

Box Plots for each of the clusters were created for all variables to check distribution of the variable. For Actual Work Hours variable, it is clear that the behavior of each cluster is different. Please refer attached phase 2 HTML file for rest of the variables' box plots.

```
[40]: for i in features_n:
          pfig = plt.figure();
          img = sns.boxplot(x="kmn_cluster",y=i,data=X_train,palette='rainbow').set_title(
          fig = img.get_figure();
          #fig.savefig(i);
          #fig.clf(); # run this for not printing figure
```

Hierarchical Clustering :

Hierarchical Clustering is a method of grouping similar data points into nested clusters. It builds a hierarchy of clusters by iteratively merging or splitting them based on similarity.

Fitting Average Linkage Model

Hierarchical clustering with average linkage merges clusters based on average distances between their members, gradually forming a hierarchy of clusters, allowing for a step-by-step exploration of relationships within data.

```
from sklearn.cluster import AgglomerativeClustering
agg_clstr = AgglomerativeClustering(n_clusters=3, metric='euclidean',  linkage='average')
agg_clstr.fit(X_train_scl)
```

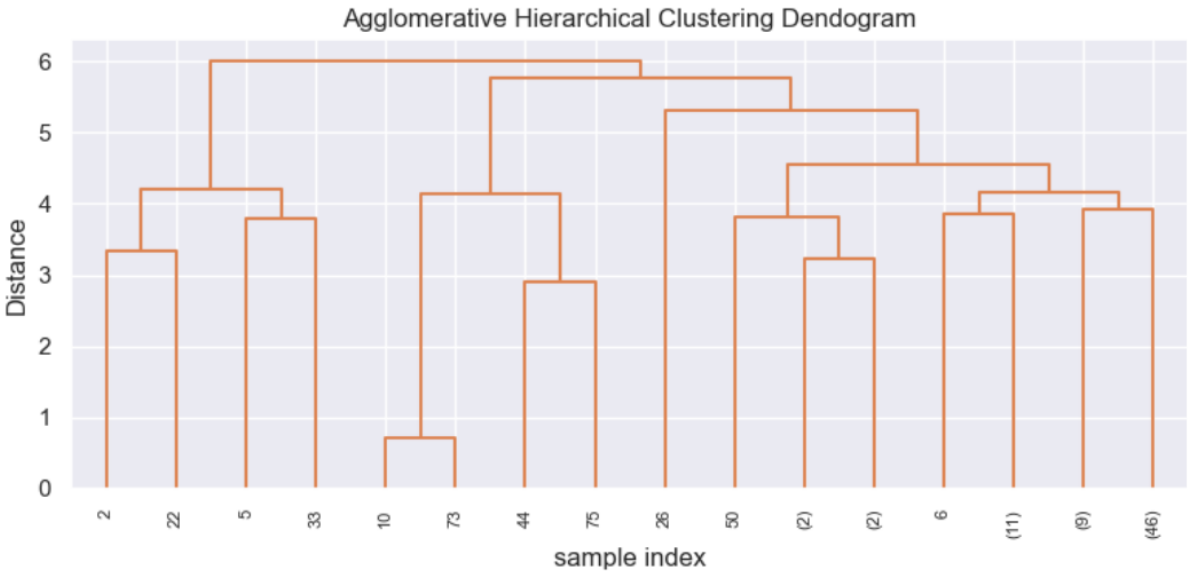
```
AgglomerativeClustering
AgglomerativeClustering(linkage='average', n_clusters=3)
```

```
trn_pred = agg_clstr.fit_predict(X_train_scl)
X_train['agg_cluster'] = trn_pred
```

Plotting Dendogram

Hierarchical clustering dendrogram is a tree-like diagram showing the hierarchical relationship between data points, illustrating their clustering based on similarity, with branches merging as clusters form.

```
z = linkage(X_train_scl, metric='euclidean', method='average')#z[:2]
plt.figure(figsize=(8, 4));
plt.title('Agglomerative Hierarchical Clustering Dendogram');
plt.xlabel('sample index');
plt.ylabel('Distance');
dendrogram(z, leaf_rotation=90.,color_threshold = 40, leaf_font_size=8.,truncate_mode="level", p=5 );
plt.tight_layout();
```



Step 4: Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction technique. It seeks to find linear combinations of features that best separate multiple classes in a dataset. By maximizing the between-class variance and minimizing within-class variance, LDA identifies the most discriminative features for classification tasks.

Difference between LDA & PCA

Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are both dimensionality reduction techniques, but they serve different purposes. LDA aims to find the linear combinations of features that best separate multiple classes in a dataset, optimizing class discrimination. It does this by maximizing the between-class variance while minimizing within-class variance. In contrast, PCA focuses on capturing the maximum variance in the dataset by identifying orthogonal components. While PCA is unsupervised and ignores class labels, LDA is supervised and considers class information. Thus, LDA is ideal for classification tasks, whereas PCA is primarily used for data exploration and visualization.

Fitting LDA Model

Scaling is required before fitting LDA model

•[46]:
```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
lda = LinearDiscriminantAnalysis()
```

[47]:
```
pipe_lda = Pipeline([('sc', scaler), ('lda', lda)])
pipe_lda.fit(X_train[features_n], y_train)
```

[47]:
```
▸ Pipeline
    ▸ StandardScaler
    ▸ LinearDiscriminantAnalysis
```

[48]:
```
X_train_lda = pipe_lda.transform(X_train[features_n])
y_train_pred = pipe_lda.predict(X_train[features_n])
X_test_lda = pipe_lda.transform(X_test[features_n])
y_test_pred = pipe_lda.predict(X_test[features_n])
```

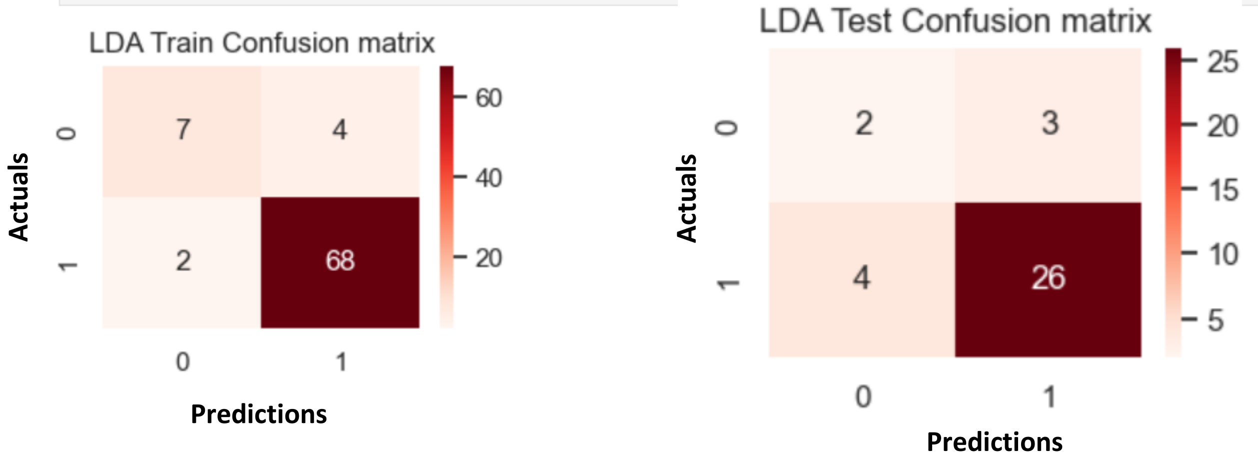
LDA Classification Report

```
estimator= pipe_lda
pred_bin_tr = estimator.predict(X_train[features_n])
pred_bin_te = estimator.predict(X_test[features_n])
#confusion_matrix(y_train,pred_bin_tr)
f'train_rocauc : {round(roc_auc_score(y_train,estimator.predict_proba(X_train[features_n])[:,1]),4)} ; \
test_rocaauc : {round(roc_auc_score(y_test,estimator.predict_proba(X_test[features_n])[:,1]),4)}'
f'train_f1_score : {round(f1_score(y_train,pred_bin_tr),4)} ; test_f1_score : {round(f1_score(y_test,pred_bin_te),4)}'
f'train_precision : {round(precision_score(y_train,pred_bin_tr),4)} ; test_precision : {round(precision_score(y_test,pred_bin_te),4)}'
f'train_recall : {round(recall_score(y_train,pred_bin_tr),4)} ; test_recall : {round(recall_score(y_test,pred_bin_te),4)}'
f'train_accuracy : {round(accuracy_score(y_train,pred_bin_tr),4)} ; test_accuracy : {round(accuracy_score(y_test,pred_bin_te),4)}'
#f'train_conf : {confusion_matrix(y_train,pred_bin_tr)} ; test_conf : {confusion_matrix(y_test,pred_bin_te)}'
```

```
'train_rocauc : 0.9792 ; test_rocaauc : 0.8267'

'train_f1_score : 0.9577 ; test_f1_score : 0.8814'

'train_precision : 0.9444 ; test_precision : 0.8966'

'train_recall : 0.9714 ; test_recall : 0.8667'

'train_accuracy : 0.9259 ; test_accuracy : 0.8'
```

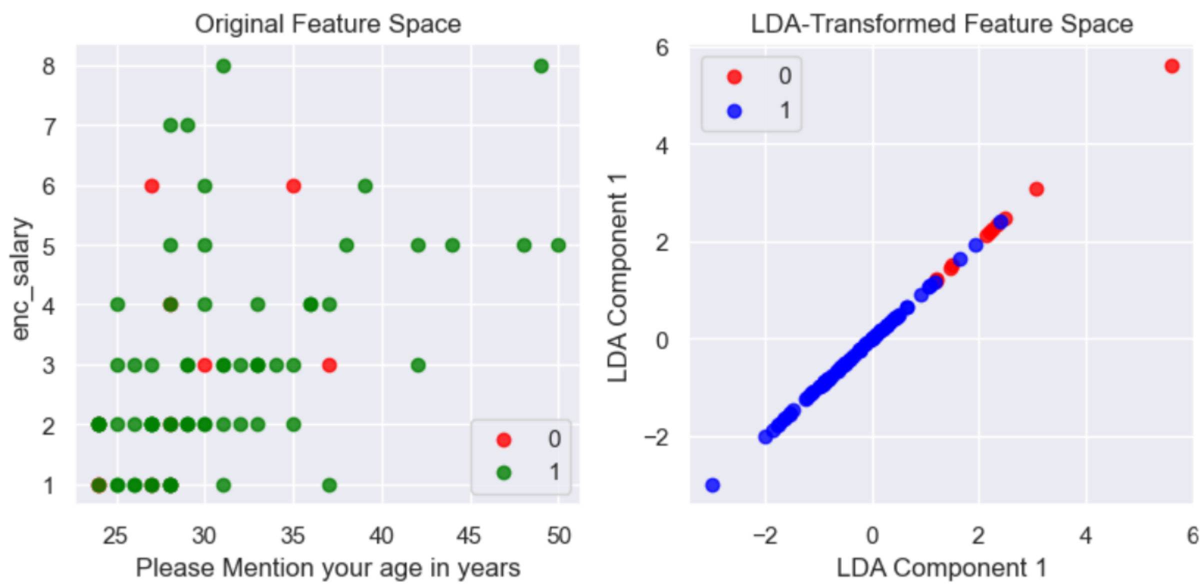
LDA Confusion Matrix

```
cf_tr = confusion_matrix(y_train,pred_bin_tr);#cf_tr
cf_te = confusion_matrix(y_test,pred_bin_te);#cf_te
fig, ax = plt.subplots(figsize=(3,2)) ;
x = sns.heatmap(cf_tr, annot=True, cmap='Reds',fmt='.0f',).set_title('LDA Train Confusion matrix')
plt.show()
fig, ax = plt.subplots(figsize=(3,2)) ;
x = sns.heatmap(cf_te, annot=True, cmap='Reds',fmt='.0f',).set_title('LDA Test Confusion matrix')
plt.show()
```



LDA Component Plot

Since there were only two levels in the target variable i.e. 0 and 1, So there will be only 1 component in LDA. Left plot shows scatter plot of top 2 variables of data and right plot shows the LDA transformed plot.



Comparison of LDA and Logistic Regression Model:

The logistic Regression model has ROC auc value of 80% whereas that of LDA is 82%. So, it can be concluded that LDA model is able to outperform logistic Regression model

PHASE 3

Cleaned data of phase 1 is used in phase3 as well.

Step 1: MANOVA

MANOVA analysis was carried out with X as all the numerical variables and Y as the “Good Worklife” variable.

```
# Fit MANOVA model
maov = MANOVA.from_formula('''Please_Mention_your_age_in_years + Work_Experience_in_years + clnd_Average_Time_to_Commute_to_office_one_way_in_minutes +
Number_of_total_paid_leave_Paid_Sick + day_hrs_per_week + clnd_Actual_Work_hours_per_week_in_hours +
Clnd_Designated_work_hours_per_week_that_a_person_is_expected_to_work_In_hours + Actual_work_hours_per_desogmated_work_hours + enc_salary ~
Good_Worklife''',
       data=df1)
result = maov.mv_test()

# Print the MANOVA results
print(result.summary())
```

```
                   Multivariate linear model
================================================================

----------------------------------------------------------------
       Intercept         Value  Num DF  Den DF   F Value  Pr > F
----------------------------------------------------------------
          Wilks' lambda  0.0123 9.0000 106.0000 944.8306 0.0000
         Pillai's trace  0.9877 9.0000 106.0000 944.8306 0.0000
 Hotelling-Lawley trace 80.2215 9.0000 106.0000 944.8306 0.0000
    Roy's greatest root 80.2215 9.0000 106.0000 944.8306 0.0000
----------------------------------------------------------------

----------------------------------------------------------------
       Good_Worklife      Value  Num DF  Den DF  F Value Pr > F
----------------------------------------------------------------
            Wilks' lambda 0.6705 9.0000 106.0000  5.7887 0.0000
           Pillai's trace 0.3295 9.0000 106.0000  5.7887 0.0000
   Hotelling-Lawley trace 0.4915 9.0000 106.0000  5.7887 0.0000
      Roy's greatest root 0.4915 9.0000 106.0000  5.7887 0.0000
================================================================
```

Conclusion: The p Value of the MANOVA is very small, which means that there is a significant difference in the mean values between the 2 groups of the target variable.

Step 2: MANCOVA

MANOVA analysis was carried out with X as all the numerical variables and Y as the “Good Worklife” variable along with other latent variables

```
# Assuming df is your DataFrame
#manova_data = df[['Please Mention your age in years', 'write', 'math', 'prog']]

# Fit MANOVA model
macov = MANOVA.from_formula('''Please_Mention_your_age_in_years + Work_Experience_in_years + clnd_Average_Time_to_Commute_to_office_one_way_in_minutes +
Number_of_total_paid_leave_Paid_Sick + day_hrs_per_week + clnd_Actual_Work_hours_per_week_in_hours +
Clnd_Designated_work_hours_per_week_that_a_person_is_expected_to_work_In_hours + Actual_work_hours_per_desogmated_work_hours + enc_salary ~
Good_Worklife + Gender + Marriage_status_Kids + Work_Mode + Employment_Type''',
       data=df1)
result2 = macov.mv_test()

# Print the MANOVA results
print(result2.summary())
```

```
                  Multivariate linear model
==============================================================

--------------------------------------------------------------
       Intercept         Value  Num DF  Den DF  F Value  Pr > F
--------------------------------------------------------------
          Wilks' lambda  0.0521 9.0000 98.0000 198.3017 0.0000
         Pillai's trace  0.9479 9.0000 98.0000 198.3017 0.0000
 Hotelling-Lawley trace 18.2114 9.0000 98.0000 198.3017 0.0000
    Roy's greatest root 18.2114 9.0000 98.0000 198.3017 0.0000
--------------------------------------------------------------

--------------------------------------------------------------
           Gender         Value  Num DF  Den DF F Value Pr > F
--------------------------------------------------------------
            Wilks' lambda 0.8713 9.0000 98.0000  1.6085 0.1232
           Pillai's trace 0.1287 9.0000 98.0000  1.6085 0.1232
   Hotelling-Lawley trace 0.1477 9.0000 98.0000  1.6085 0.1232
      Roy's greatest root 0.1477 9.0000 98.0000  1.6085 0.1232
--------------------------------------------------------------

--------------------------------------------------------------
  Marriage_status_Kids  Value   Num DF  Den DF   F Value Pr > F
--------------------------------------------------------------
          Wilks' lambda 0.4606 18.0000 196.0000  5.1556 0.0000
         Pillai's trace 0.5729 18.0000 198.0000  4.4155 0.0000
 Hotelling-Lawley trace 1.0985 18.0000 159.5973  5.9330 0.0000
    Roy's greatest root 1.0278  9.0000  99.0000 11.3061 0.0000
--------------------------------------------------------------
```

Conclusion : The p Value of the MANCOVA is very small for **Marriage_status_kids**, which means that there is a significant difference in the mean values between the 2 groups of the target variable.

The p Value of the MANOVA is high for **Gender**, which means that there is not significant difference in the mean values between the 2 groups of the target variable.

```
----------------------------------------------------------------
       Work_Mode        Value   Num DF  Den DF   F Value Pr > F
----------------------------------------------------------------
          Wilks' lambda 0.5528 18.0000 196.0000  3.7566 0.0000
         Pillai's trace 0.5035 18.0000 198.0000  3.7011 0.0000
 Hotelling-Lawley trace 0.7072 18.0000 159.5973  3.8194 0.0000
    Roy's greatest root 0.5058  9.0000  99.0000  5.5641 0.0000
----------------------------------------------------------------

----------------------------------------------------------------
    Employment_Type     Value   Num DF  Den DF   F Value Pr > F
----------------------------------------------------------------
          Wilks' lambda 0.6291 27.0000 286.8529  1.8273 0.0088
         Pillai's trace 0.4146 27.0000 300.0000  1.7819 0.0114
 Hotelling-Lawley trace 0.5222 27.0000 211.0650  1.8746 0.0077
    Roy's greatest root 0.3532  9.0000 100.0000  3.9244 0.0003
----------------------------------------------------------------

----------------------------------------------------------------
       Good_Worklife      Value  Num DF  Den DF F Value Pr > F
----------------------------------------------------------------
            Wilks' lambda 0.6475 9.0000 98.0000  5.9282 0.0000
           Pillai's trace 0.3525 9.0000 98.0000  5.9282 0.0000
   Hotelling-Lawley trace 0.5444 9.0000 98.0000  5.9282 0.0000
      Roy's greatest root 0.5444 9.0000 98.0000  5.9282 0.0000
================================================================
```

Multivariate linear model
==

The p Value of the MANCOVA is very less for **Work_mode**, which means that there is significant difference in the mean values between the 2 groups of the target variable.

The p Value of the MANCOVA is high for **Employment_Type**, which means that there is not significant difference in the mean values between the 2 groups of the target variable.

The p Value of the MANCOVA is very less for **Good_Worklife**, which means that there is significant difference in the mean values between the 2 groups of the target variable.

Step 3: Structural Equation Modeling (SEM)

Structural Equation Modelling (SEM) is a statistical method used to test and estimate complex relationships among variables. It incorporates both observed and latent variables to evaluate causal relationships and model complex theoretical constructs. SEM allows researchers to assess the direct and indirect effects of variables on each other and to evaluate the fit of the hypothesized models to the observed data. It is widely used in social sciences, psychology, economics, and other fields to analyze complex systems and test theoretical models, offering a comprehensive approach to understanding relationships between variables in multivariate data.

Multi-Variate Regression:

Multivariate Regression model as a part of SEM was developed with target variables as Good Worklife, Gender, Marriage status and numerical variables as independent variable.

```
fmla1 = "Good_Worklife,Gender_Male, Marriage_status_Kids_Married_and_have_Kids ~ Please_Mention_your_age_in_years + Work_Experience_in_years + clnd_Avera
print(fmla1)
```

```
Good_Worklife,Gender_Male, Marriage_status_Kids_Married_and_have_Kids ~ Please_Mention_your_age_in_years + Work_Experience_in_years + clnd_Average_Time_t
o_Commute_to_office_one_way_in_minutes + Number_of_total_paid_leave_Paid_Sick + day_hrs_per_week + clnd_Actual_Work_hours_per_week_in_hours + Clnd_Design
ated_work_hours_per_week_that_a_person_is_expected_to_work_In_hours + Actual_work_hours_per_desogmated_work_hours + enc_salary
```

```
import semopy
model1 = semopy.Model(fmla1)
result1 = model1.fit(df1_ohe)
print(result1)
```

```
Name of objective: MLW
Optimization method: SLSQP
Optimization successful.
Optimization terminated successfully
Objective value: 0.053
Number of iterations: 35
Params: 0.010 0.000 0.000 0.003 -0.028 -0.017 0.011 -0.144 -0.017 -0.012 0.025 0.002 -0.003 -0.086 0.005 0.006 -0.034 0.020 0.016 0.040 0.001 0.001 0.010
-0.005 -0.007 0.029 -0.014 0.122 0.080 0.111
```

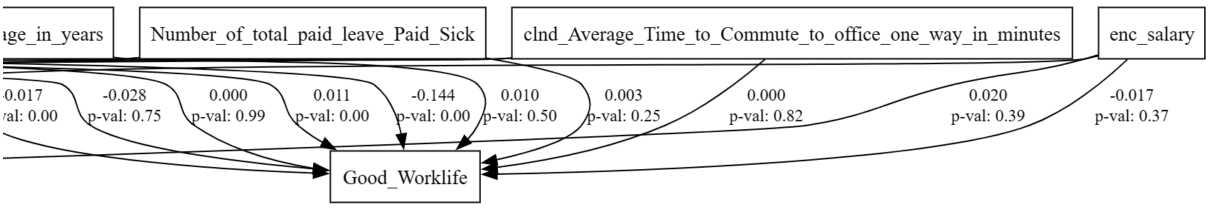
Inspecting the fitted SEM Model:

Smaller P-Value is showing the higher significance of the independent variable e.g. **Actual Work life balance, Average time to commute.**

```
ins1 = model1.inspect()
ins1
```

	lval	op	rval	Estimate	Std. Err	z-value	p-value
0	Good_Worklife	~	Please_Mention_your_age_in_years	0.009759	0.014573	0.669626	5.030964e-01
1	Good_Worklife	~	Work_Experience_in_years	0.000233	0.013662	0.017082	9.863711e-01
2	Good_Worklife	~	clnd_Average_Time_to_Commute_to_office_one_way...	0.000181	0.000804	0.225528	8.215683e-01
3	Good_Worklife	~	Number_of_total_paid_leave_Paid_Sick	0.002771	0.002385	1.161842	2.452996e-01
4	Good_Worklife	~	day_hrs_per_week	-0.027884	0.085895	-0.324633	7.454589e-01
5	Good_Worklife	~	clnd_Actual_Work_hours_per_week_in_hours	-0.016501	0.002963	-5.570026	2.547007e-08
6	Good_Worklife	~	Clnd_Designated_work_hours_per_week_that_a_per...	0.011117	0.003959	2.808449	4.978076e-03
7	Good_Worklife	~	Actual_work_hours_per_desogmated_work_hours	-0.144054	0.032741	-4.399839	1.083310e-05
8	Good_Worklife	~	enc_salary	-0.016858	0.018812	-0.896156	3.701696e-01
9	Gender_Male	~	Please_Mention_your_age_in_years	-0.011700	0.018046	-0.648361	5.167517e-01
10	Gender_Male	~	Work_Experience_in_years	0.025113	0.016917	1.484504	1.376753e-01
11	Gender_Male	~	clnd_Average_Time_to_Commute_to_office_one_way...	0.001781	0.000995	1.789582	7.352120e-02
12	Gender_Male	~	Number_of_total_paid_leave_Paid_Sick	-0.003283	0.002953	-1.111880	2.661898e-01
13	Gender_Male	~	day_hrs_per_week	-0.086469	0.106360	-0.812979	4.162300e-01
14	Gender_Male	~	clnd_Actual_Work_hours_per_week_in_hours	0.004910	0.003668	1.338510	1.807303e-01

Plotting the SEM Model: (please refer HTML phase 3 file for full plot)



SEM Model with Measurement part, Structural part and additional Covariances

Creating Formula for complex SEM Model with Measurement Model, Regression model and Residual Correlations

```
fmla2 = '''# measurement model
sem1 =~ Please_Mention_your_age_in_years + Work_Experience_in_years + clnd_Average_Time_to_Commute_to_office_one_way_in_minutes + Number_of_total_paid_leave_P
sem2 =~ Good_Worklife +  Gender_Female + Gender_Male + Marriage_status_Kids_Married_and_No_Kids + Marriage_status_Kids_Married_and_have_Kids + Marriage_status
sem3 =~ Work_Mode_Hybrid_WFH_WFO + Work_Mode_Work_From_Home_WFH + Work_Mode_Work_From_Office_WFO + Employment_Type_Contract_Basis + Employment_Type_Freelancin
# regressions
sem2 ~ sem1
sem3 ~ sem1 + sem2
# residual correlations
Good_Worklife ~~ Work_Mode_Hybrid_WFH_WFO + Marriage_status_Kids_Married_and_have_Kids
Gender_Male ~~ Work_Mode_Work_From_Home_WFH + Employment_Type_Permanent__Company_Payroll
Good_Worklife ~~ Marriage_status_Kids_Un_Married
Work_Mode_Hybrid_WFH_WFO ~~ Employment_Type_Internship
'''
```

Fitting Complex SEM model

```
model2 = semopy.Model(fmla2)
model2.fit(df1_ohe)
ins2 = model2.inspect()
ins2
```

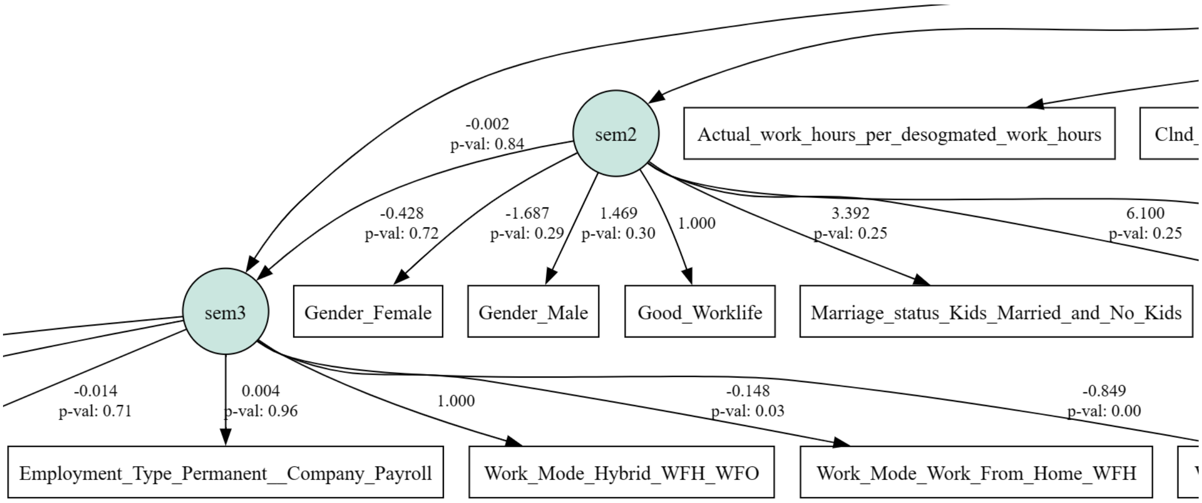
```
WARNING:root:Sample covariance matrix is not PD. It may indicate that data is bad. Also, it arises often when p
py now will run nearPD subroutines.
```

```
 SolverResult(fun=16.221867023407437, success=True, n_it=222, x=array([ 1.05481466e+00,  7.76463871e-01,  6.035
        3.51782114e-01,  1.74924088e-01,  3.65060581e-04,  1.71891760e-01,
       -1.68728337e+00,  1.46948768e+00,  3.39190587e+00,  6.10032246e+00,
       -9.58793551e+00, -1.48083743e-01, -8.49169386e-01,  5.79545354e-04,
        1.31913151e-02, -1.43888578e-02,  3.51751037e-03,  6.06887853e-03,
       -2.16292538e-03, -4.28308169e-01,  2.29786706e-03, -1.65400438e-02,
        1.93391348e-03,  1.16378768e-01, -1.77162916e-02,  6.95984907e-03,
        1.40746753e-01,  3.04978796e-03,  1.03385924e-15,  7.11636883e-01,
        5.91305677e+01,  7.15683561e-02,  1.69048590e-02,  2.51724349e-02,
        1.06123986e-01,  1.40522101e-01,  1.11444899e-01,  1.12746477e-01,
        0.00000000e+00,  1.29646428e+02,  2.05536410e+00,  1.81508087e+00,
        8.14434613e-02,  8.42127115e-02,  1.04835022e+02,  1.10015004e+03,
        1.15337997e-01,  2.50288636e+00,  3.16714784e+01,  1.55122906e-03,
        2.22303839e-01]), message='Optimization terminated successfully', name_method='SLSQP', name_obj='MLW')
```

Smaller P-Value is showing the higher significance of the independent variable e.g. Actual Work life balance, Average time to commute.

	lval	op	rval	Estimate	Std. Err	z-value	p-value
0	sem2	~	sem1	6.068879e-03	0.005135	1.181767	0.237298
1	sem3	~	sem1	-2.162925e-03	0.010545	-0.205106	0.83749
2	sem3	~	sem2	-4.283082e-01	1.182463	-0.362217	0.71719
3	Please_Mention_your_age_in_years	~	sem1	1.000000e+00	-	-	-
4	Work_Experience_in_years	~	sem1	1.054815e+00	0.044466	23.722046	0.0
5	clnd_Average_Time_to_Commute_to_office_one_way...	~	sem1	7.764639e-01	0.55515	1.398655	0.161916
6	Number_of_total_paid_leave_Paid_Sick	~	sem1	6.035113e-01	0.191149	3.157289	0.001592
7	day_hrs_per_week	~	sem1	5.947070e-03	0.005682	1.046579	0.295294
8	clnd_Actual_Work_hours_per_week_in_hours	~	sem1	3.517821e-01	0.171516	2.05102	0.040265
9	Clnd_Designated_work_hours_per_week_that_a_per...	~	sem1	1.749241e-01	0.128698	1.359179	0.17409
10	Actual_work_hours_per_desogmated_work_hours	~	sem1	3.650606e-04	0.014109	0.025874	0.979358
11	enc_salary	~	sem1	1.718918e-01	0.026874	6.396255	0.0
12	Good_Worklife	~	sem2	1.000000e+00	-	-	-
13	Gender_Female	~	sem2	-1.687283e+00	1.582506	-1.06621	0.286329
14	Gender_Male	~	sem2	1.469488e+00	1.411969	1.040737	0.297998
15	Marriage_status_Kids_Married_and_No_Kids	~	sem2	3.391906e+00	2.944972	1.151761	0.249419
16	Marriage_status_Kids_Married_and_have_Kids	~	sem2	6.100322e+00	5.27864	1.155662	0.24782

Plotting the SEM Model: (please refer HTML phase 3 file for full plot)



Final Conclusion of Phase 1,2&3 :

From the analysis carried out in all the 3 phases it is evident that the *Work life balance* is depending on attributes like *"Actual Work Hours per week", "Marriage status", "Day hours per week", "Salary" "Age" "Designated work hours per week".*

<u>THE END</u>