Employee Salary Prediction

CS - 5805 Machine Learning - 1

Venkata Chaitanya Kanakamedala

Contents

- Problem statement
  - About the Dataset
- Phase-1 (Feature Engineering & EDA)
- Phase-2 (Regression Analysis)
- Phase-3 (Classification Analysis)
- Phase -4 (Clustering & Association)

Dataset

- Source
  - https://www.kaggle.com/datasets/mukeshmanral/employ-earnings-data/data
- 10,00,000 data points
- Total Attributes - 23
- Important Attributes :19
- Target class : High Salary

Attributes in Original Dataset

Categorical_features

['jobType', 'degree', 'major', ‘industry’, 'high_salary']

Numerical_features

['yearsExperience', 'milesFromMetropolis', 'salary']

EDA

Data preprocessing:
- Data Cleaning
- Insignificant attributes are removed

Duplicate Checking & Removal
Aggregation
DownSampling
Discretization & Binarization : One hot Encoding
Variable Transformation : Standardization
Anomaly Detection
Covariance Matrix
Pearson correlation matrix
Balanced / Imbalanced Data

Dimensionality Reduction Techniques

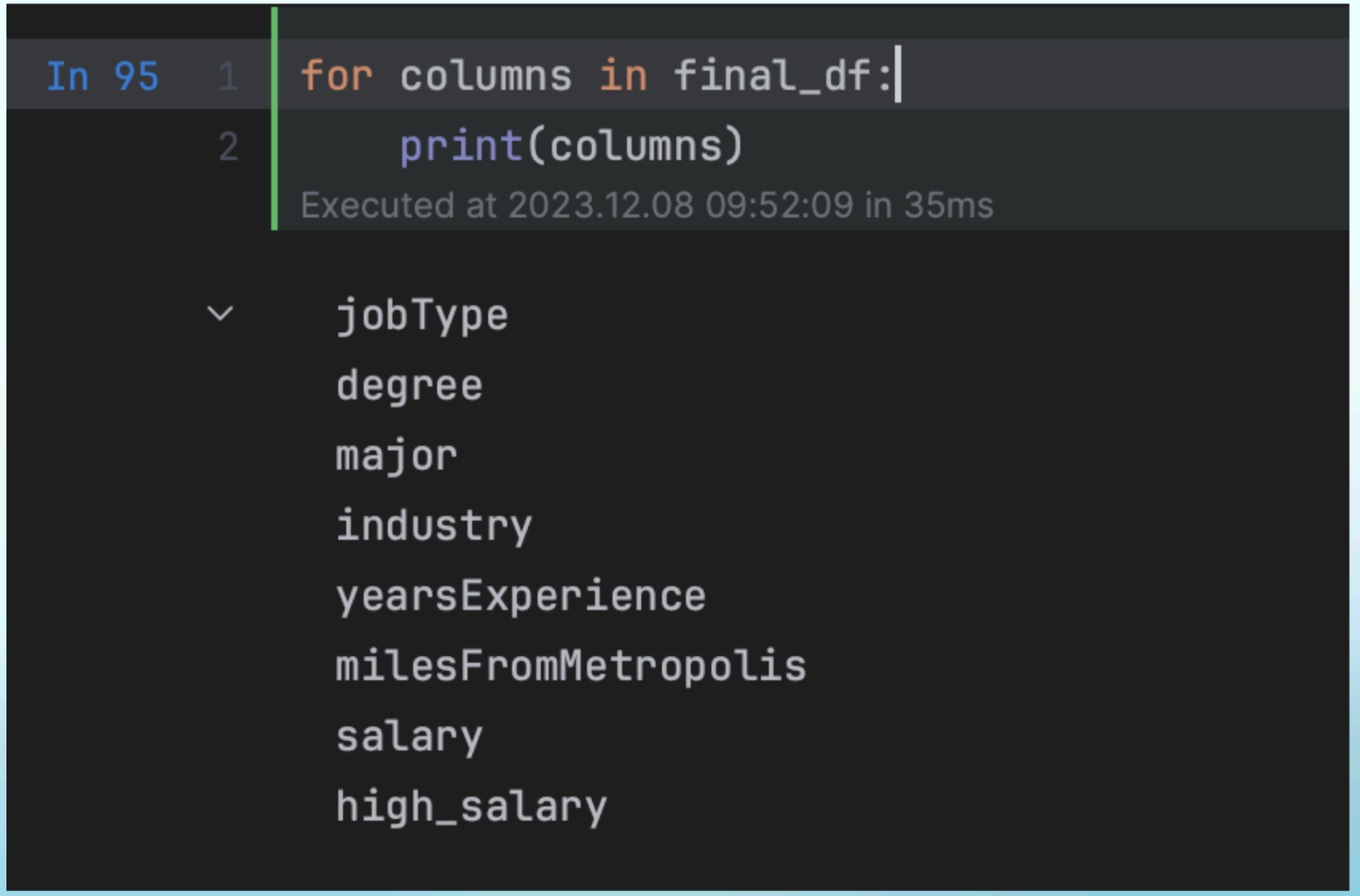
Random Forest Analysis

Principal Component Analysis & Condition number

Singular value decomposition analysis

VIF

Attributes in Dataset



Phase 1

Feature Engineering & Exploratory Data Analysis

Phase -1 : Exploratory Data Analysis

Data Pre processing : The missing Nan values were removed from the dataset and also “NONE” values have also been removed. t is mainly because our dataset represents individual employee salaries and its details and hence replacing with mean and mode would result in alteration of data.

Duplicates Check : In the data frame duplicates were really low and had been removed.

Aggregation (If applicable) : There was not a necessity in my dataset to aggregate the features

Down Sampling : Downsampling has been implemented in-order to balance the ‘high salary’ attribute .

Discretization & Binarization : One hot encoding was implemented on the categorical columns because the original data lacked inherent relationships among these categorical features. Additionally, this encoding technique helped circumvent the issue of the dummy variable trap

Variable Transformation : Standardization has been applied on the numerical columns from the dataset and made sure not to include one-hot encoded variables.

Data Pre Processing

The missing Nan values were removed from the dataset and also “NONE” values have also been removed. t is mainly because our dataset represents individual employee salaries and its details and hence replacing with mean and mode would result in alteration of data.

Also in this we have ensured to drop insignificant feature like ‘CompanyID’ from the dataframe.

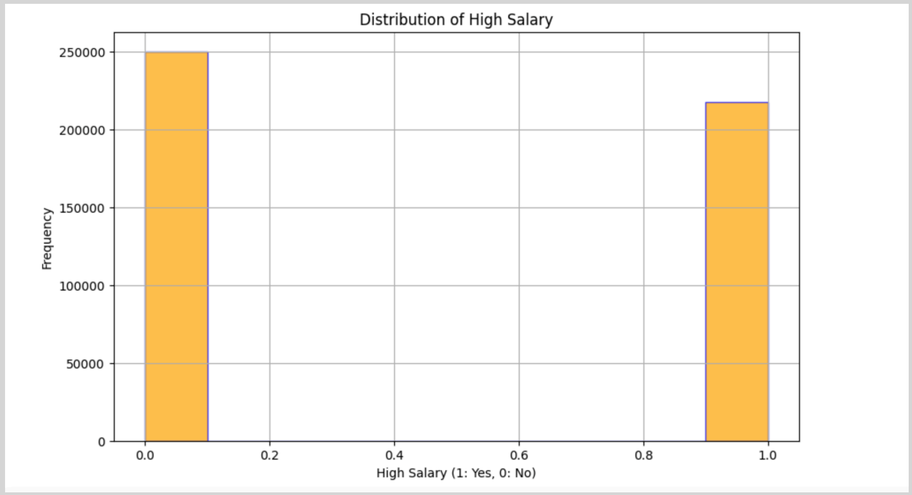
Our final data frame looks like this :

5 rows | 5 rows × 8 columns pd.DataFrame | CSV

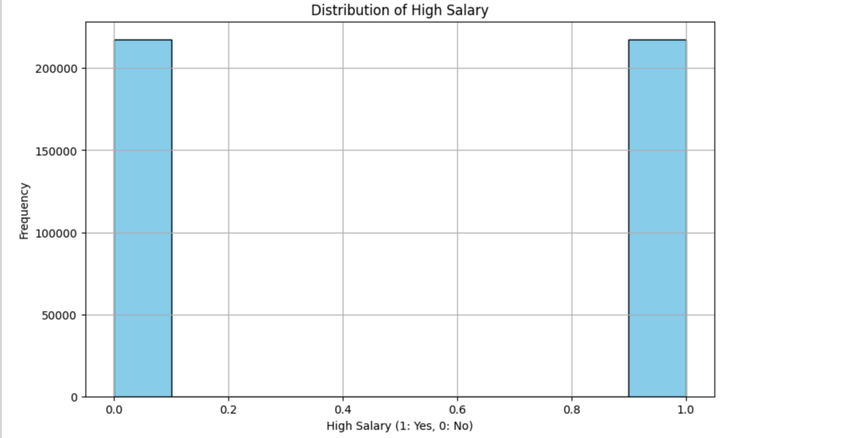
jobType	degree	major	industry	yearsExperience	milesFromMetropolis	salary	high_salary
VICE_PRESIDENT	MASTERS	ENGINEERING	HEALTH	8	74	109	0
CEO	DOCTORAL	MATH	OIL	2	97	130	0
JUNIOR	BACHELORS	MATH	OIL	2	73	75	0
CTO	MASTERS	ENGINEERING	SERVICE	11	28	114	0
JUNIOR	BACHELORS	LITERATURE	AUTO	22	71	110	0

Down Sampling

Downsampling has been done on the High Salary column and ensured to balance the dataset.



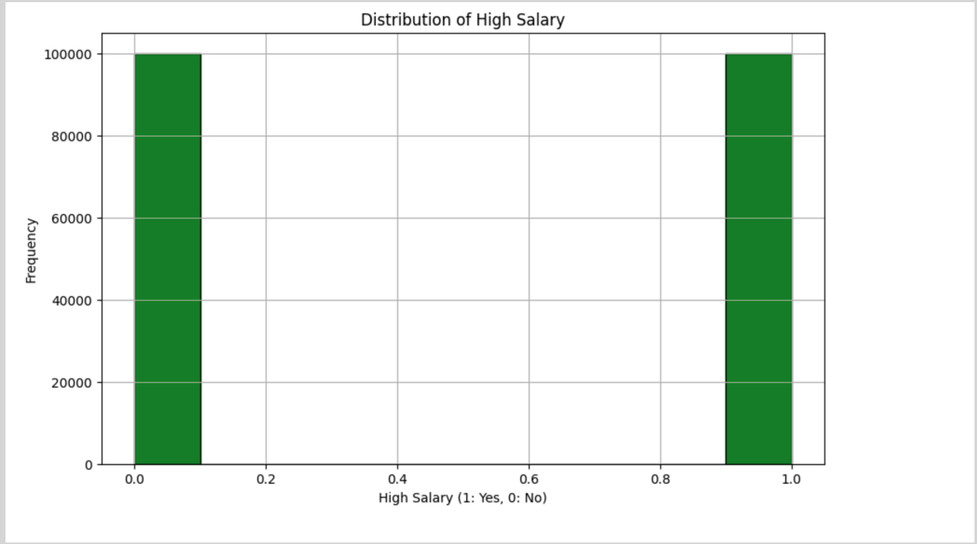

```
Class distribution after undersampling:
0    217587
1    217587
Name: high_salary, dtype: int64
```



From above plots we can infer that High_Salary column is balanced.

Dataset Selection

Data Selection: Due to large size of data and expensive computation only 2,00,000 observations have been considered for our dataset.



Note : Also Skewness in the target variable was comparatively low in the salary column and hence balancing the column wasn't necessary.

```
Skewness Summary:
0.38905457562646345
```

One hot Encoding

One hot encoding has been performed on the categorical columns to convert them into binary features and ensured to avoid dummy trap.

```
[ ]  # Perform one-hot encoding
     df_encoded = pd.get_dummies(final_df, columns=categorical_features, drop_first=True)
     df_encoded = df_encoded.astype(int)
```

```
[ ]  df_encoded.columns
```

```
Index(['yearsExperience', 'milesFromMetropolis', 'salary', 'high_salary',
       'jobType_CFO', 'jobType_CTO', 'jobType_JUNIOR', 'jobType_MANAGER',
       'jobType_SENIOR', 'jobType_VICE_PRESIDENT', 'degree_DOCTORAL',
       'degree_MASTERS', 'major_BUSINESS', 'major_CHEMISTRY', 'major_COMPSCI',
       'major_ENGINEERING', 'major_LITERATURE', 'major_MATH', 'major_PHYSICS',
       'industry_EDUCATION', 'industry_FINANCE', 'industry_HEALTH',
       'industry_OIL', 'industry_SERVICE', 'industry_WEB'],
      dtype='object')
```

Standardization

Splitting the dataset & Standardization :

We ensured to split the dataset in 80:20 set alongside with a random state, to ensure the same splits will be used even for any further use. Also the data has been split before standardization so test data can remain intact for classification and for regression have been standardized for better performance.

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5805)
```

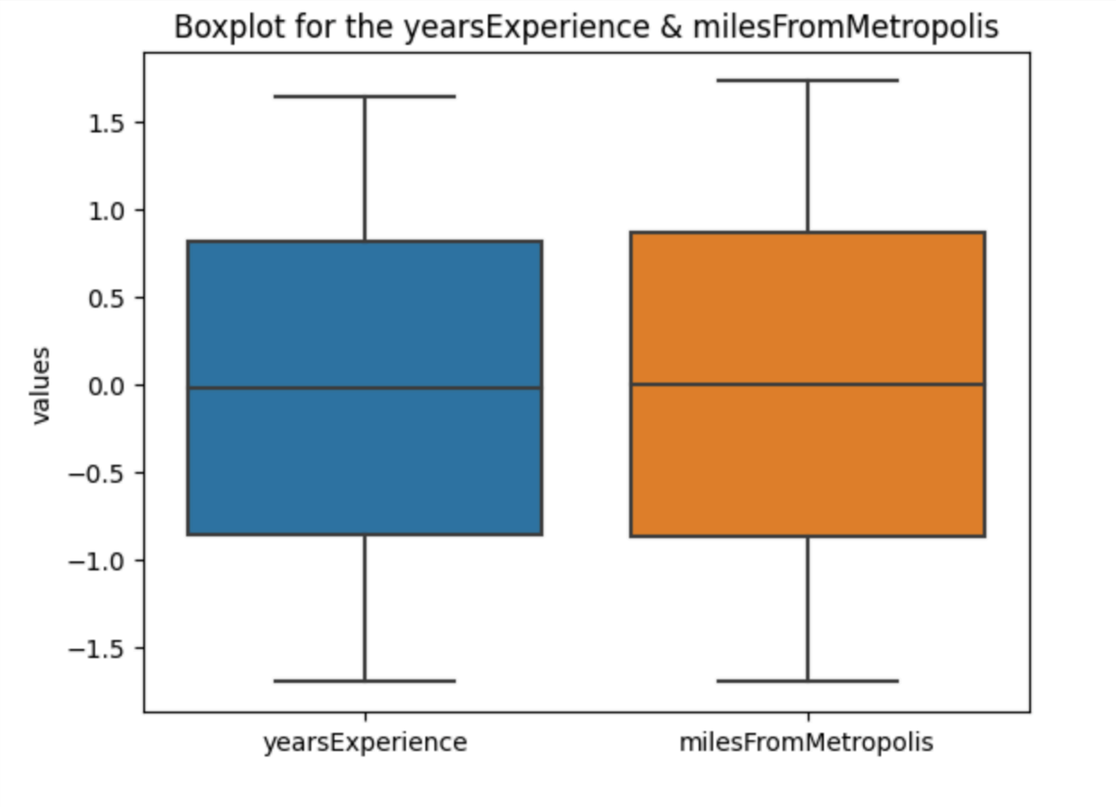
Standardization for Regression

```
[ ]
    x_scaler = StandardScaler()
    X_train[['yearsExperience', 'milesFromMetropolis']] = x_scaler.fit_transform(X_train[['yearsExperience', 'milesFromMetropolis']])
    X_test[['yearsExperience', 'milesFromMetropolis']] = x_scaler.transform(X_test[['yearsExperience', 'milesFromMetropolis']])
    X_train_std = X_train
    X_test_std = X_test

    # Target standardization
    y_scaler = StandardScaler()
    y_train_std = y_scaler.fit_transform(y_train.values.reshape(-1, 1))
    y_test_std = y_scaler.transform(y_test.values.reshape(-1, 1))
```

Anomaly Detection

Outlier Analysis:

We have plotted box plots to analyze any outliers in the ‘years of experience’ and ‘miles from metropolis’ columns and have interestingly found no outliers.



Years Experience

```
Number of outliers: 0
```

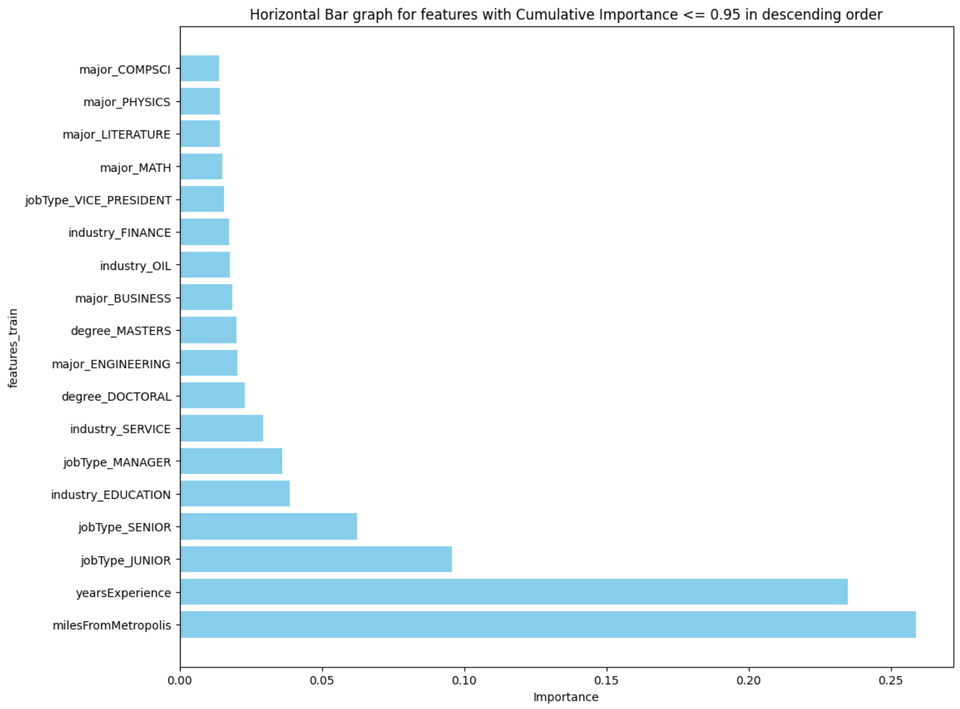
milesFromMetropolis

```
Number of outliers: 0
```

Random Forest Analysis

Random Forest Analysis:

Random forest analysis helps with the importance of each of each attribute in the data frame.

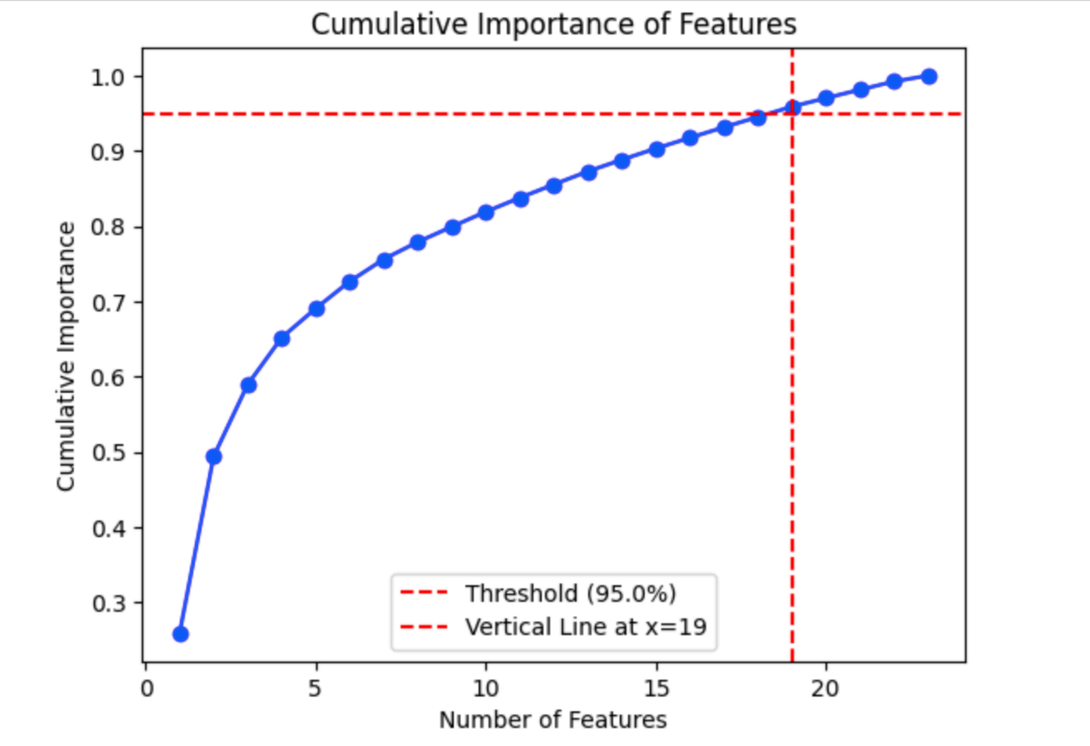



```
Selected Features with Cumulative Importance <= 0.95:
milesFromMetropolis: 0.2589078498351973
yearsExperience: 0.23488915255848236
jobType_JUNIOR: 0.09558179373032452
jobType_SENIOR: 0.06235194744743013
industry_EDUCATION: 0.03876553179430325
jobType_MANAGER: 0.03602052520760728
industry_SERVICE: 0.029418699999393488
degree_DOCTORAL: 0.022980668988518548
major_ENGINEERING: 0.020207572694088058
degree_MASTERS: 0.019914600366749507
major_BUSINESS: 0.018593544413996622
industry_OIL: 0.017655406383662572
industry_FINANCE: 0.017419345183413672
jobType_VICE_PRESIDENT: 0.015512568506980953
major_MATH: 0.014926250375535393
major_LITERATURE: 0.01422137260640727
major_PHYSICS: 0.014023798398943278
major_COMPSCI: 0.013707353024132517
```

Random Forest Analysis

Random Forest Analysis:

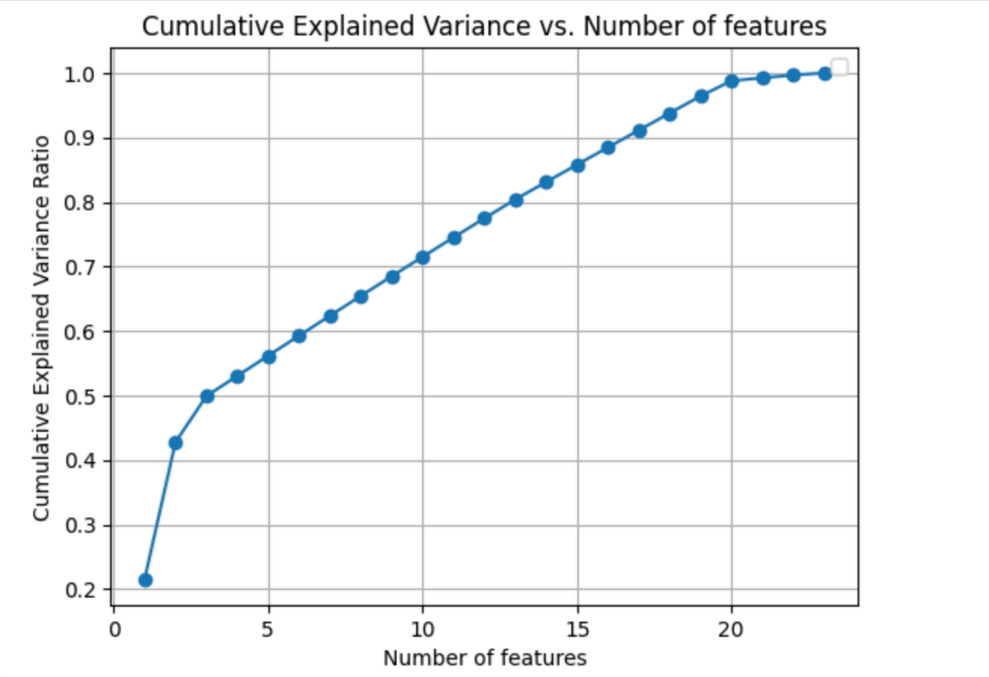
The cumulative importance plot suggests that 19 features within the data frame are adequate for describing the entire data frame, meeting a threshold of 95%.



Principal Component Analysis

Principal Component Analysis:

PCA informs us about the number of important features rather than their exact nature. Additionally, it allows us to observe a reduction in the condition number.. PCA suggested that 96% of variance is explained by 19 components.



Condition number for X_train_std

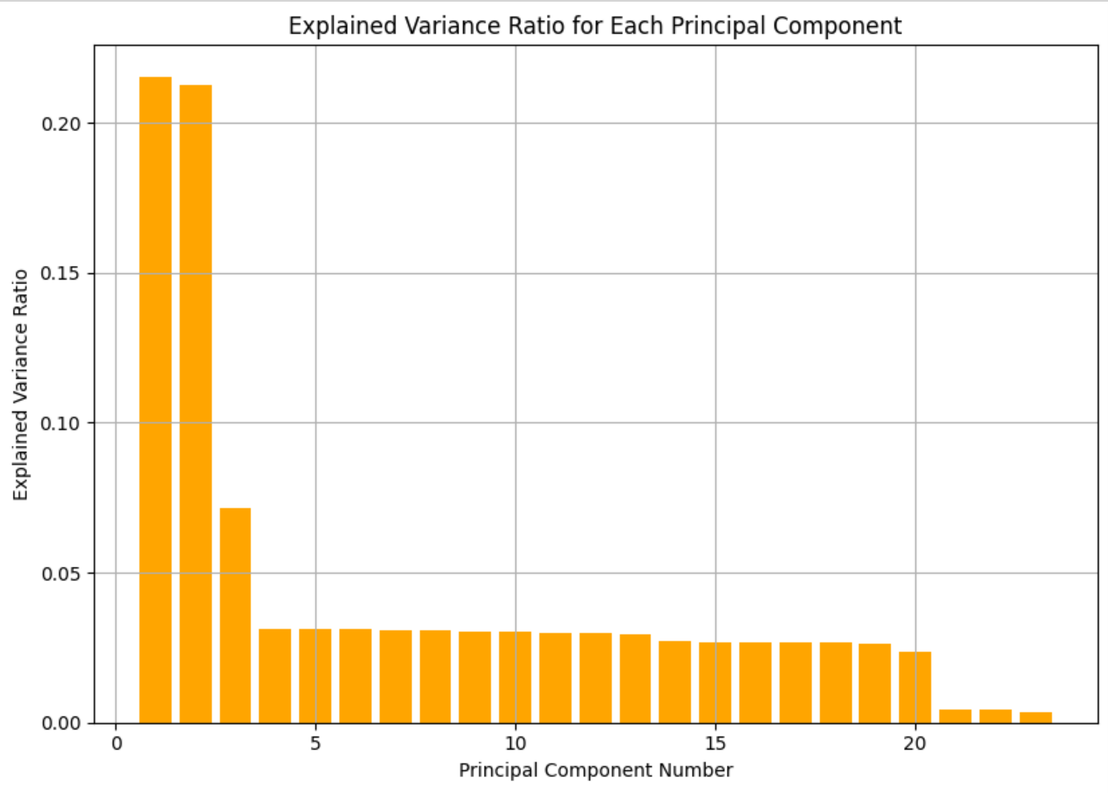
```
74  1  print(f'PCA: condition number for reduced data (important features): {np.linalg.cond(X_train_std):.2f}')
       Executed at 2023.12.07 16:47:01 in 1s 349ms

PCA: condition number for reduced data (important features): 7.68
```

Condition number for X_pca (with 19 components)

```
6  print(f'PCA: condition number for reduced data (important features): {np.linalg.cond(X_pca):.2f}')
7
   Executed at 2023.12.07 16:50:06 in 1s 199ms

PCA: condition number for reduced data (important features): 2.86
```

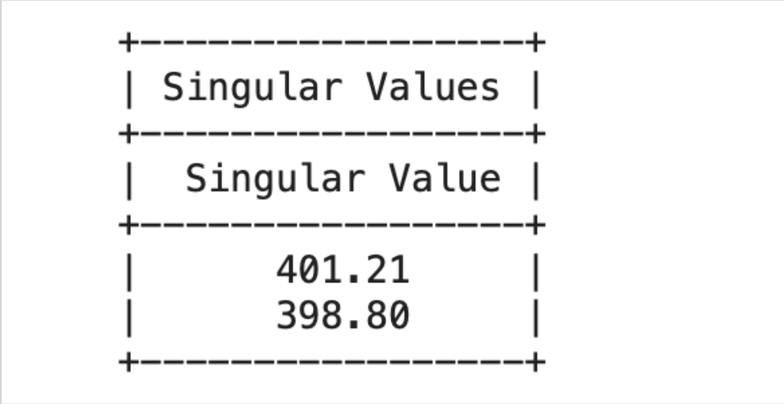
Explained Variance Ratio

Explained Variance Ratio of Each Component:
The plot of the explained variance ratio for each component provides insights into the proportion of variance captured by each principal component in the dataset.



Singular Value Decomposition

SVD : Singular Value Decomposition
The size of singular values shows how "important" or "influential" the corresponding singular vectors are. Bigger singular values play a greater role in shaping the overall structure of the matrix.



Singular Values
Singular Value
401.21
398.80

Variance Inflation Factor

VIF: Variance Inflation Factor

It is a measure that indicates about the collinearity among the attributes and higher the VIF values indicates that 2 attributes are highly correlated.

In our data frame all the VIF values Range between 1 & 5 indicating very less collinearity among the attributes.

	feature	VIF
0	yearsExperience	4.351266
1	milesFromMetropolis	3.700796
2	high_salary	3.131817
3	jobType_CFO	1.745438
4	jobType_CTO	1.754898
5	jobType_JUNIOR	1.851428
6	jobType_MANAGER	1.768484
7	jobType_SENIOR	1.815938
8	jobType_VICE_PRESIDENT	1.747679
9	degree_DOCTORAL	1.962727
10	degree_MASTERS	1.920865
11	major_BUSINESS	1.758601
12	major_CHEMISTRY	1.728773
13	major_COMPSCI	1.738850
14	major_ENGINEERING	1.768056
15	major_LITERATURE	1.711845
16	major_MATH	1.734658
17	major_PHYSICS	1.730428
18	industry_EDUCATION	1.724913
19	industry_FINANCE	1.900000
20	industry_HEALTH	1.790974
21	industry_OIL	1.897297
22	industry_SERVICE	1.738143
23	industry_WEB	1.834712

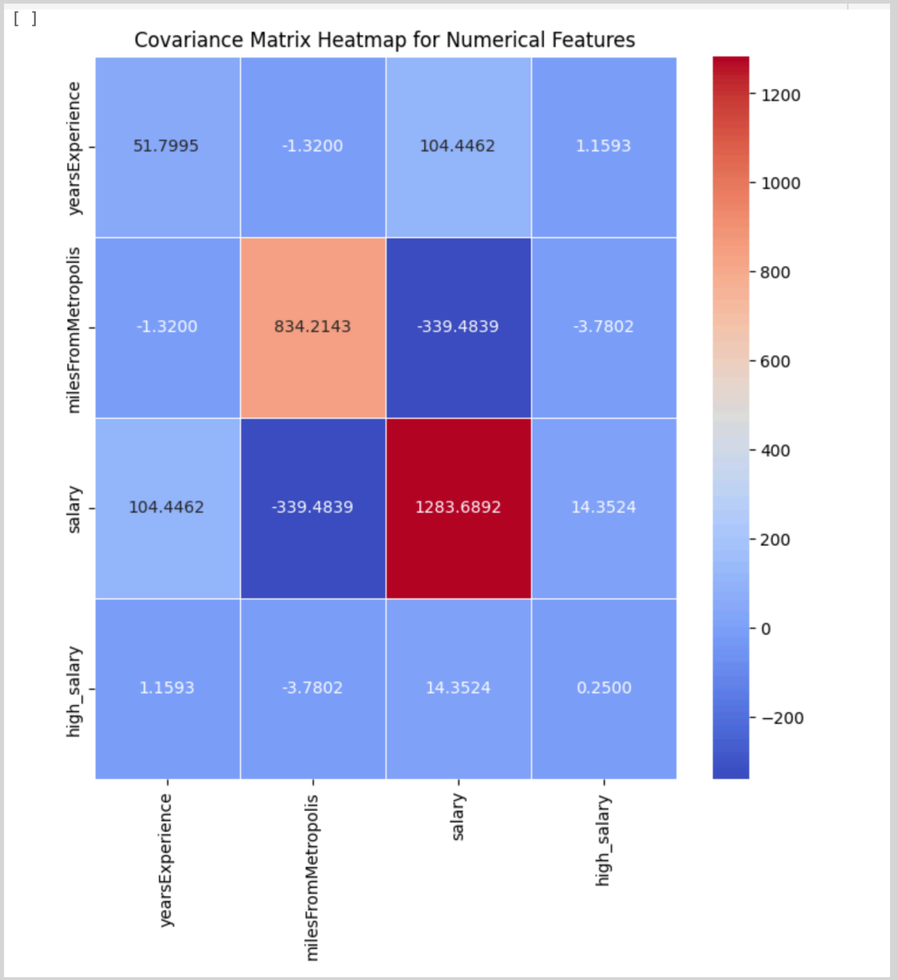
Covariance Matrix

Covariance Matrix:

This matrix tells about the degree to which 2 attributes are related in the data frame.

The off diagonal elements represent covariance between each pair of attributes while diagonal elements represent variance of the attributes.

Note - The diagonal variables indicate the variance of each variable.



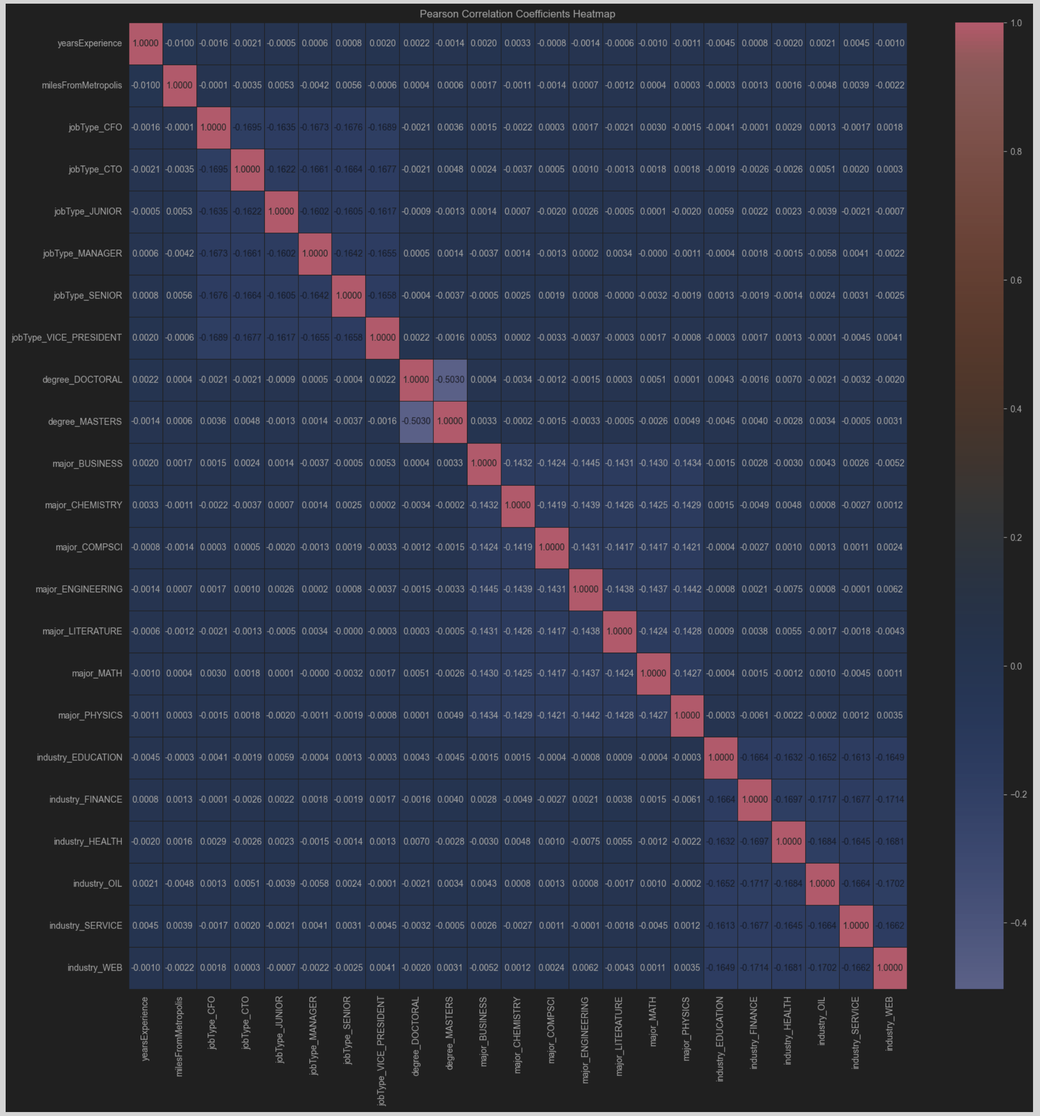
Correlation Matrix

Correlation Matrix:

This matrix is more like a standardized version of the covariance matrix where each cell represents the Pearson correlation coefficient between two attributes.

Here :

+1 -> Positive Correlation
-1 -> Negative Correlation
0 -> No Correlation



Phase 2

Regression Analysis

T - Test Analysis

-The t-test is frequently employed in research and data analysis to assess whether the means of two samples differ significantly.

-A substantial t-statistic (i.e., a large absolute value) suggests a notable distinction between the samples.

-The notable difference between the samples implies that the divergence is improbable to be attributed to random chance, affirming its likely authenticity and significance.

-Therefore, all the features are authentic and meaningful.

T Test Results:

	coef	std err	t	P>|t|	[0.025	0.975]
yearsExperience	0.4045	0.002	264.439	0.000	0.402	0.408
milesFromMetropolis	-0.3225	0.002	-210.818	0.000	-0.326	-0.320
jobType_CFO	-0.2222	0.005	-42.824	0.000	-0.232	-0.212
jobType_CTO	-0.2239	0.005	-42.959	0.000	-0.234	-0.214
jobType_JUNIOR	-1.3361	0.005	-251.626	0.000	-1.347	-1.326
jobType_MANAGER	-0.7803	0.005	-148.931	0.000	-0.791	-0.770
jobType_SENIOR	-1.0563	0.005	-201.724	0.000	-1.067	-1.046
jobType_VICE_PRESIDENT	-0.4973	0.005	-95.293	0.000	-0.507	-0.487
degree_DOCTORAL	0.3033	0.004	83.829	0.000	0.296	0.310
degree_MASTERS	0.1630	0.004	45.011	0.000	0.156	0.170
major_BUSINESS	0.2763	0.006	49.985	0.000	0.266	0.287
major_CHEMISTRY	0.0983	0.006	17.780	0.000	0.087	0.109
major_COMPSCI	0.1760	0.006	31.740	0.000	0.165	0.187
major_ENGINEERING	0.3606	0.006	65.487	0.000	0.350	0.371
major_LITERATURE	-0.0348	0.006	-6.296	0.000	-0.046	-0.024
major_MATH	0.2027	0.006	36.603	0.000	0.192	0.214
major_PHYSICS	0.1319	0.006	23.889	0.000	0.121	0.143
industry_EDUCATION	-0.2293	0.005	-43.190	0.000	-0.240	-0.219
industry_FINANCE	0.6731	0.005	129.347	0.000	0.663	0.683
industry_HEALTH	0.2629	0.005	50.007	0.000	0.253	0.273
industry_OIL	0.6884	0.005	131.611	0.000	0.678	0.699
industry_SERVICE	-0.0835	0.005	-15.804	0.000	-0.094	-0.073
industry_WEB	0.4559	0.005	87.092	0.000	0.446	0.466

F test Analysis

The F-test evaluates whether the regression model as a whole significantly predicts the outcome variable.

1. Variance Explained:
   The F-test compares the amount of variance in the outcome explained by the whole model to the unexplained variance.

2. Statistical Significance:
   A statistically significant F-test indicates that the set of predictors jointly have a real relationship with the outcome variable.

3. Model Evaluation:
   The F-test helps determine if the observed relationships between predictors and outcome are likely real or occurred by random chance.

```
2 Printing OLS summary
)
Initial model_initial Summary:
                                OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared (uncentered):                   0.626
Model:                            OLS   Adj. R-squared (uncentered):              0.626
Method:                 Least Squares   F-statistic:                          1.163e+04
Date:                Thu, 07 Dec 2023   Prob (F-statistic):                        0.00
Time:                        13:02:43   Log-Likelihood:                     -1.4842e+05
No. Observations:              160000   AIC:                                  2.969e+05
Df Residuals:                  159977   BIC:                                  2.971e+05
Df Model:                          23
Covariance Type:            nonrobust
==============================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
yearsExperience            0.4045      0.002    264.439      0.000       0.402       0.408
milesFromMetropolis       -0.3225      0.002   -210.818      0.000      -0.326      -0.320
jobType_CFO               -0.2222      0.005    -42.824      0.000      -0.232      -0.212
jobType_CTO               0.2239      0.005     43.059      0.000       0.214       0.234
```

Linear Regression

Upon conducting Linear Regression, a conclusive insight can be drawn from the OLS Regression results regarding the P>|t| values, all of which are zero. This implies the significance of all features. Additionally, in the VIF analysis, no significant collinearity was detected among the attributes, eliminating the need for removal.

Hence there was no necessity in the code to perform backward stepwise regression for this dataset.

```
AIC: 296300.2951627823
BIC: 296539.8854610435
R-squared: 0.6270460113011633
MSE: 0.37295398869883284
```

OLS Regression Results

Dep. Variable:	y	R-squared:	0.627
Model:	OLS	Adj. R-squared:	0.627
Method:	Least Squares	F-statistic:	1.169e+04
Date:	Thu, 07 Dec 2023	Prob (F-statistic):	0.00
Time:	13:02:46	Log-Likelihood:	-1.4813e+05
No. Observations:	160000	AIC:	2.963e+05
Df Residuals:	159976	BIC:	2.965e+05
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.1717	0.007	24.123	0.000	0.158	0.186
yearsExperience	0.4045	0.002	264.919	0.000	0.402	0.408
milesFromMetropolis	-0.3225	0.002	-211.177	0.000	-0.325	-0.319
jobType_CFO	-0.2748	0.006	-48.899	0.000	-0.286	-0.264
jobType_CTO	-0.2763	0.006	-49.089	0.000	-0.287	-0.265
jobType_JUNIOR	-1.3886	0.006	-242.357	0.000	-1.400	-1.377
jobType_MANAGER	-0.8331	0.006	-146.941	0.000	-0.844	-0.822
jobType_SENIOR	-1.1090	0.006	-195.754	0.000	-1.120	-1.098
jobType_VICE_PRESIDENT	-0.5498	0.006	-97.380	0.000	-0.561	-0.539
degree_DOCTORAL	0.2795	0.004	74.682	0.000	0.272	0.287
degree_MASTERS	0.1395	0.004	37.241	0.000	0.132	0.147
major_BUSINESS	0.2134	0.006	34.955	0.000	0.201	0.225
major_CHEMISTRY	0.0348	0.006	5.689	0.000	0.023	0.047
major_COMPSCI	0.1125	0.006	18.361	0.000	0.101	0.125
major_ENGINEERING	0.2972	0.006	48.795	0.000	0.285	0.309
major_LITERATURE	-0.0981	0.006	-16.047	0.000	-0.110	-0.086
major_MATH	0.1394	0.006	22.792	0.000	0.127	0.151
major_PHYSICS	0.0681	0.006	11.153	0.000	0.056	0.080
industry_EDUCATION	-0.2850	0.006	-49.303	0.000	-0.296	-0.274
industry_FINANCE	0.6174	0.006	108.593	0.000	0.606	0.629
industry_HEALTH	0.2072	0.006	36.152	0.000	0.196	0.218
industry_OIL	0.6330	0.006	110.996	0.000	0.622	0.644
industry_SERVICE	-0.1394	0.006	-24.204	0.000	-0.151	-0.128
industry_WEB	0.4006	0.006	70.190	0.000	0.389	0.412

Omnibus:	2376.634	Durbin-Watson:	2.003
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1729.877
Skew:	0.155	Prob(JB):	0.00
Kurtosis:	2.596	Cond. No.	11.1

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Confidence Interval Analysis

1. Estimate Reliability: Confidence intervals indicate the reliability of a sample estimate by providing a range of plausible values for the true population parameter. Wider intervals signal greater uncertainty.

2. Testing Hypotheses: If a hypothesized value falls outside the confidence interval, we can reject that hypothesis at the chosen confidence level. Values inside the interval indicate a lack of significant evidence against the hypothesis.

3. Population Inference: The confidence interval gives a data-driven range of parameter values that we can reasonably expect contain the true population value. For instance, a 95% confidence interval provides a range where we are 95% sure the population parameter resides.

Confidence Intervals:

	0	1
const	0.157754	0.185656
yearsExperience	0.401537	0.407523
milesFromMetropolis	-0.325469	-0.319483
jobType_CFO	-0.285821	-0.263791
jobType_CTO	-0.287370	-0.265268
jobType_JUNIOR	-1.399816	-1.377356
jobType_MANAGER	-0.844260	-0.822034
jobType_SENIOR	-1.120112	-1.097904
jobType_VICE_PRESIDENT	-0.560880	-0.538748
degree_DOCTORAL	0.272202	0.286874
degree_MASTERS	0.132141	0.146822
major_BUSINESS	0.201416	0.225345
major_CHEMISTRY	0.022800	0.046768
major_COMPSCI	0.100535	0.124564
major_ENGINEERING	0.285289	0.309167
major_LITERATURE	-0.110138	-0.086161
major_MATH	0.127453	0.151436
major_PHYSICS	0.056173	0.080124
industry_EDUCATION	-0.296285	-0.273629
industry_FINANCE	0.606248	0.628534
industry_HEALTH	0.195985	0.218453
industry_OIL	0.621825	0.644180
industry_SERVICE	-0.150728	-0.128146
industry_WEB	0.389382	0.411752

Stepwise Regression

Upon removing and checking the features I have got the best results for removing the Industry Health feature and with an increased Adj R Square value from .627 to .769.

```
X_train_std.drop(['industry_HEALTH'],axis=1,inplace=True)
model_initial = sm.OLS(y_train_std,X_train_std).fit()
print(model_initial.summary())
```

Executed at 2023.12.06 16:30:41 in 341ms

OLS Regression Results

Dep. Variable:	y	R-squared (uncentered):	0.769
Model:	OLS	Adj. R-squared (uncentered):	0.769
Method:	Least Squares	F-statistic:	2.312e+04
Date:	Wed, 06 Dec 2023	Prob (F-statistic):	0.00
Time:	16:30:41	Log-Likelihood:	-1.0990e+05
No. Observations:	160000	AIC:	2.198e+05
Df Residuals:	159977	BIC:	2.201e+05
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
yearsExperience	0.2541	0.001	196.693	0.000	0.252	0.257
milesFromMetropolis	-0.2026	0.001	-161.084	0.000	-0.205	-0.200
high_salary	0.9410	0.003	323.940	0.000	0.935	0.947
jobType_CFO	-0.2877	0.004	-71.327	0.000	-0.296	-0.280
jobType_CTO	-0.2881	0.004	-71.565	0.000	-0.296	-0.280
jobType_JUNIOR	-0.9789	0.004	-232.651	0.000	-0.987	-0.971
jobType_MANAGER	-0.6187	0.004	-152.696	0.000	-0.627	-0.611
jobType_SENIOR	-0.7900	0.004	-192.216	0.000	-0.798	-0.782
jobType_VICE_PRESIDENT	-0.4529	0.004	-112.321	0.000	-0.461	-0.445
degree_DOCTORAL	0.1266	0.003	43.843	0.000	0.121	0.132
degree_MASTERS	0.0399	0.003	13.947	0.000	0.034	0.045
major_BUSINESS	0.0172	0.004	3.936	0.000	0.009	0.026
major_CHEMISTRY	-0.1163	0.004	-26.876	0.000	-0.125	-0.108
major_COMPSCI	-0.0625	0.004	-14.371	0.000	-0.071	-0.054
major_ENGINEERING	0.0611	0.004	13.965	0.000	0.053	0.070
major_LITERATURE	-0.1966	0.004	-45.573	0.000	-0.205	-0.188
major_MATH	-0.0397	0.004	-9.105	0.000	-0.048	-0.031
major_PHYSICS	-0.0881	0.004	-20.291	0.000	-0.097	-0.080
industry_EDUCATION	-0.3021	0.004	-79.255	0.000	-0.310	-0.295
industry_FINANCE	0.2614	0.004	67.998	0.000	0.254	0.269
industry_OIL	0.2745	0.004	70.980	0.000	0.267	0.282
industry_SERVICE	-0.2178	0.004	-57.221	0.000	-0.225	-0.210
industry_WEB	0.1236	0.004	32.376	0.000	0.116	0.131

Omnibus:	6500.405	Durbin-Watson:	1.994
Prob(Omnibus):	0.000	Jarque-Bera (JB):	8198.911
Skew:	0.444	Prob(JB):	0.00
Kurtosis:	3.666	Cond. No.	8.18

Polynomial Regression

1. Flexibility of Polynomials:
   The degree of the polynomial controls the flexibility to model nonlinear relationships. Higher-degree polynomials can fit more complex curves but may overfit.

2. Avoiding Overfitting:
   Care should be taken to avoid overfitting the data, especially with high-degree polynomials. Regularization and cross-validation can help find the right model complexity.

3. Interpretation Difficulties:
   Highly flexible polynomial models can become challenging to interpret. While they may provide good fit, clearly communicating what the model shows in meaningful terms is crucial.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.662
Model:	OLS	Adj. R-squared:	0.660
Method:	Least Squares	F-statistic:	282.6
Date:	Thu, 07 Dec 2023	Prob (F-statistic):	0.00
Time:	13:16:04	Log-Likelihood:	-1.4027e+05
No. Observations:	160000	AIC:	2.827e+05
Df Residuals:	158898	BIC:	2.937e+05
Df Model:	1101		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.2304	0.039	5.835	0.000	0.153	0.308
x1	0.4039	0.020	20.278	0.000	0.365	0.443
x2	-0.3176	0.020	-16.170	0.000	-0.356	-0.279
x3	1.663e+11	7.09e+10	2.347	0.019	2.74e+10	3.05e+11
x4	-3.652e+11	2.5e+11	-1.462	0.144	-8.55e+11	1.25e+11
x5	1.504e+11	9.36e+10	1.606	0.108	-3.31e+10	3.34e+11
x6	-1.172e+11	7.13e+10	-1.644	0.100	-2.57e+11	2.25e+10
x7	1.653e+11	2.91e+11	0.567	0.571	-4.06e+11	7.36e+11
x8	1.649e+09	1.67e+11	0.010	0.992	-3.26e+11	3.29e+11
x9	2.219e+11	2.22e+11	0.998	0.318	-2.14e+11	6.58e+11
x10	1.311e+09	1.17e+11	0.011	0.991	-2.28e+11	2.31e+11
x11	2.42e+10	3.18e+10	0.760	0.447	-3.82e+10	8.66e+10
x12	-1.814e+11	2.14e+11	-0.847	0.397	-6.01e+11	2.38e+11
x13	1.864e+11	2.76e+11	0.677	0.499	-3.54e+11	7.26e+11
x14	1.294e+10	3.23e+11	0.040	0.968	-6.19e+11	6.45e+11
x15	6.657e+10	4.86e+10	1.371	0.170	-2.86e+10	1.62e+11
x16	6.195e+11	3.14e+11	1.974	0.048	4.42e+09	1.23e+12
x17	1.97e+10	3.37e+10	0.584	0.559	-4.64e+10	8.58e+10
x18	1.283e+10	4.17e+10	0.307	0.758	-6.9e+10	9.46e+10
x19	2.45e+10	3.83e+10	0.640	0.522	-5.05e+10	9.95e+10

Phase 3

Classification Analysis

Decision Tree

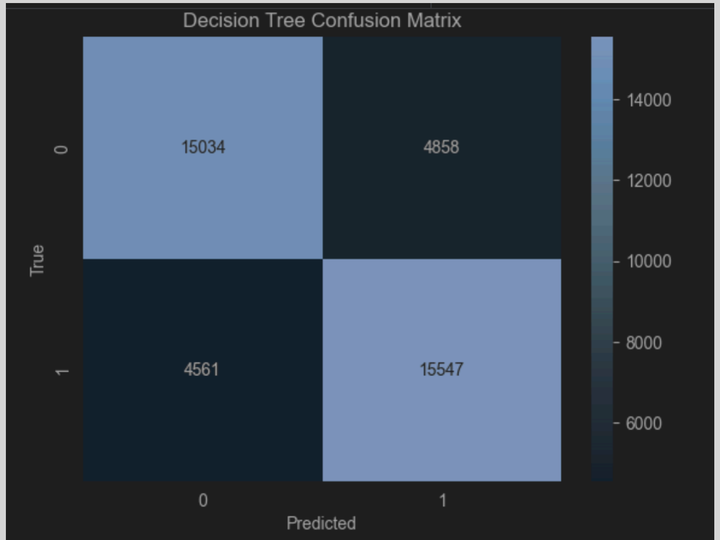
Pre Pruning

Confusion Matrix Analysis:

The confusion matrix displays the number of correct and incorrect predictions made by the model.
True Positives (TP): 15,547 instances where the model correctly predicted class 1.
True Negatives (TN): 15,034 instances where the model correctly predicted class 0.
False Positives (FP): 4,858 instances incorrectly predicted as class 1.
False Negatives (FN): 4,561 instances incorrectly predicted as class 0.

Classification Report:

Precision for class 0 is 0.77, indicating a reasonably high likelihood that a predicted class 0 is actually class 0.
Recall for class 0 is 0.76, meaning the model correctly identifies 76% of actual class 0 instances.
F1-score, the harmonic mean of precision and recall, stands at 0.76 for both classes, suggesting a balanced classification performance.
The overall accuracy of the model is 0.76, which shows that the model correctly predicts 76% of the time.



Classification Report:

	precision	recall	f1-score	support
0	0.77	0.76	0.76	19892
1	0.76	0.77	0.77	20108
accuracy			0.76	40000
macro avg	0.76	0.76	0.76	40000
weighted avg	0.76	0.76	0.76	40000

Individual Metrics:
Precision: [0.76723654 0.7619211]
Recall: [0.75578122 0.77317486]
F1-Score: [0.7614658 0.76750673]
Accuracy: 0.764525
Specificity: 0.7557812185803338

Decision Tree

Pre Pruning

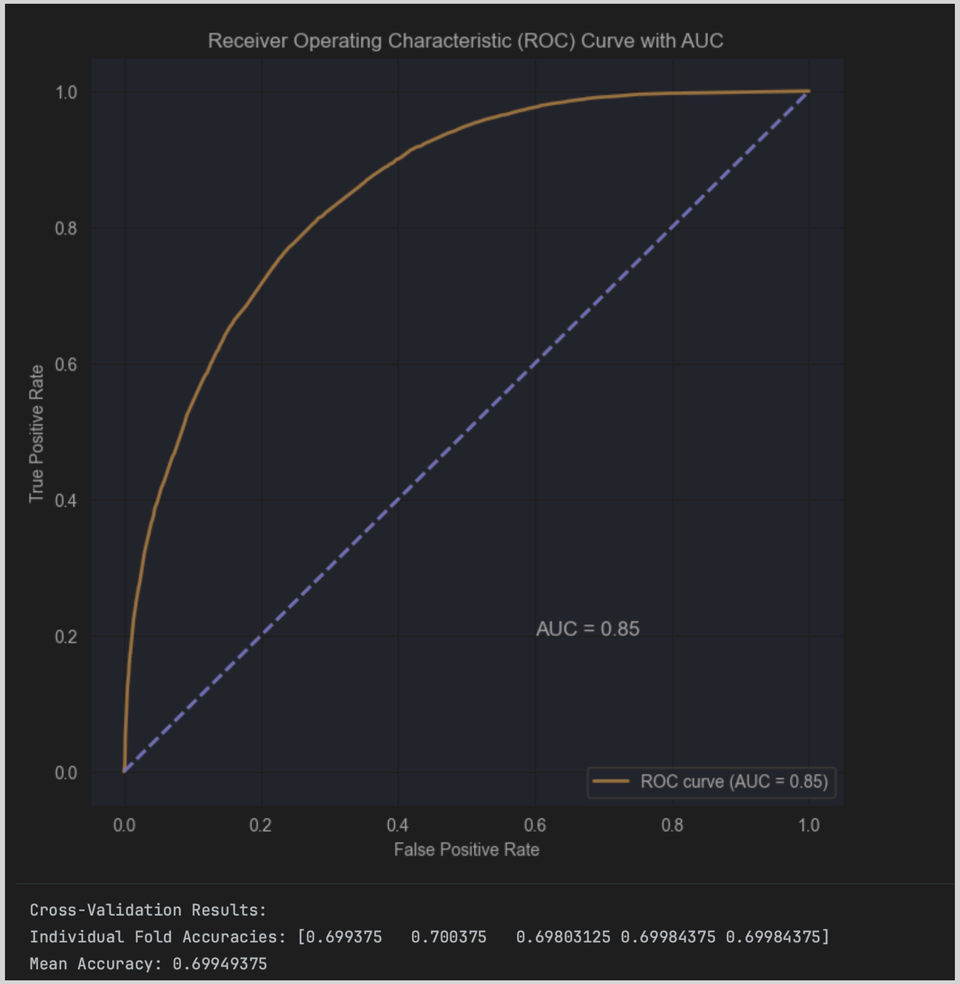
Individual Metrics:

Precision for class 1 is slightly lower at 0.76, while recall is marginally higher at 0.77.
The specificity, or true negative rate, is 0.75578, indicating the model's ability to identify true negatives among all negative instances.
ROC Curve and AUC:

The ROC curve demonstrates the trade-off between the true positive rate and the false positive rate.
An AUC (Area Under the Curve) of 0.85 signifies a high level of model discrimination, meaning the model is good at distinguishing between the classes.

Cross-Validation Results:

Individual fold accuracies range from approximately 0.69 to 0.70, with a mean accuracy of 0.69949 across the folds. This consistency across folds suggests that the model is stable and not overly fitted to a particular subset of the data.



Decision Tree

Post Pruning

Confusion Matrix Insights:

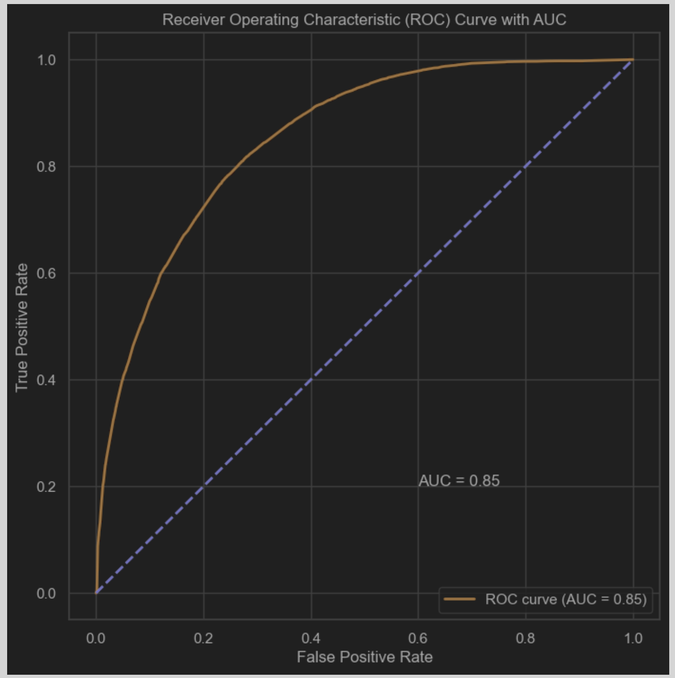
The matrix shows 14,497 true positives and 12,263 true negatives, indicating correct classifications for both classes.
There are 5,395 false positives and 3,845 false negatives, which highlight potential areas for model improvement.
Classification Report Summary:

Class 0 precision is 0.79, and recall is 0.73, resulting in an F1-score of 0.76, indicating good predictive performance for the negative class.
Class 1 has a slightly lower precision of 0.75 but a higher recall of 0.81, with an F1-score of 0.78, suggesting better sensitivity for the positive class.
The overall model accuracy is reported at 0.77, with consistent macro and weighted averages.

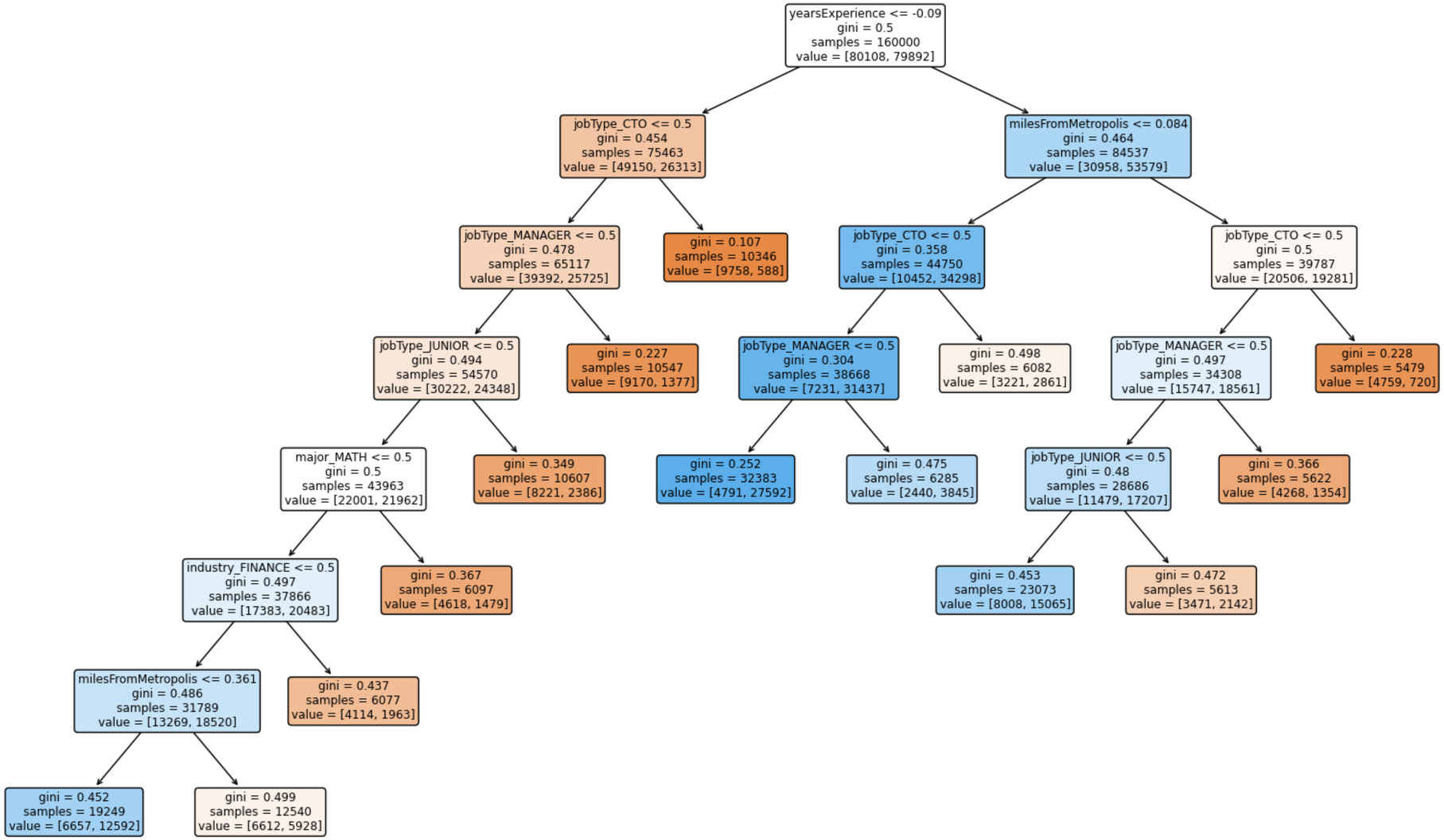
Confusion Matrix Insights:

The matrix shows 14,497 true positives and 12,263 true negatives, indicating correct classifications for both classes.
There are 5,395 false positives and 3,845 false negatives, which highlight potential areas for model improvement.

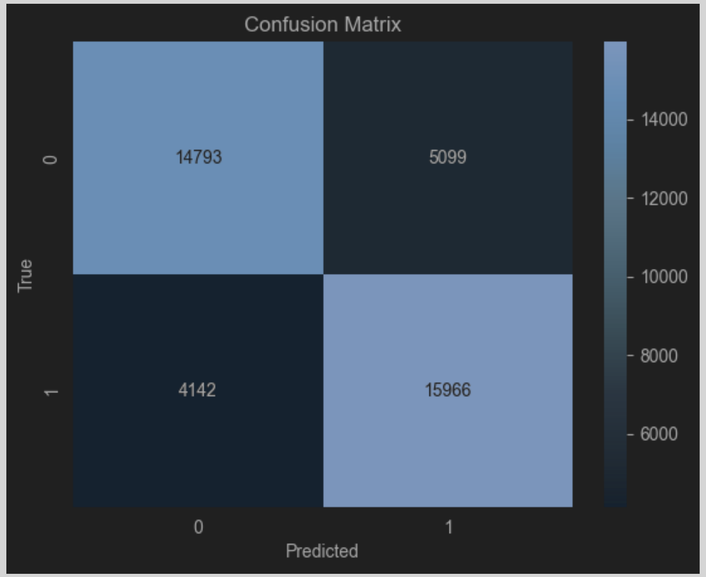
```
Confusion Matrix:
[[14497  5395]
 [ 3845 16263]]

Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.73      0.76     19892
           1       0.75      0.81      0.78     20108

    accuracy                           0.77     40000
   macro avg       0.77      0.77      0.77     40000
weighted avg       0.77      0.77      0.77     40000


Individual Metrics:
Precision: [0.79037182 0.75090036]
Recall: [0.72878544 0.80878257]
F1-Score: [0.75833028 0.77876742]
Accuracy: 0.769
Specificity: 0.7287854413834708
```




```
Cross-Validation Results:
Individual Fold Accuracies: [0.71084375 0.71165625 0.71275    0.71259375 0.71371875]
Mean Accuracy: 0.7123125
```

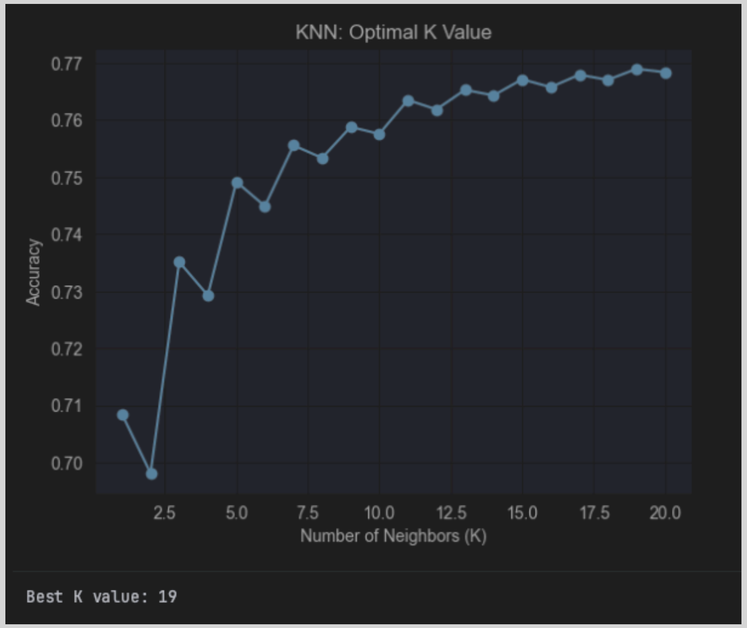

yearsExperience <= -0.09
gini = 0.5
samples = 160000
value = [80108, 79892]
jobType_CTO <= 0.5
gini = 0.454
samples = 75463
value = [49150, 26313]
milesFromMetropolis <= 0.084
gini = 0.464
samples = 84537
value = [30958, 53579]
jobType_MANAGER <= 0.5
gini = 0.478
samples = 65117
value = [39392, 25725]
gini = 0.107
samples = 10346
value = [9758, 588]
jobType_CTO <= 0.5
gini = 0.358
samples = 44750
value = [10452, 34298]
jobType_CTO <= 0.5
gini = 0.5
samples = 39787
value = [20506, 19281]
jobType_JUNIOR <= 0.5
gini = 0.494
samples = 54570
value = [30222, 24348]
gini = 0.227
samples = 10547
value = [9170, 1377]
jobType_MANAGER <= 0.5
gini = 0.304
samples = 38668
value = [7231, 31437]
gini = 0.498
samples = 6082
value = [3221, 2861]
jobType_MANAGER <= 0.5
gini = 0.497
samples = 34308
value = [15747, 18561]
gini = 0.228
samples = 5479
value = [4759, 720]
major_MATH <= 0.5
gini = 0.5
samples = 43963
value = [22001, 21962]
gini = 0.349
samples = 10607
value = [8221, 2386]
gini = 0.252
samples = 32383
value = [4791, 27592]
gini = 0.475
samples = 6285
value = [2440, 3845]
jobType_JUNIOR <= 0.5
gini = 0.48
samples = 28686
value = [11479, 17207]
gini = 0.366
samples = 5622
value = [4268, 1354]
industry_FINANCE <= 0.5
gini = 0.497
samples = 37866
value = [17383, 20483]
gini = 0.367
samples = 6097
value = [4618, 1479]
gini = 0.453
samples = 23073
value = [8008, 15065]
gini = 0.472
samples = 5613
value = [3471, 2142]
milesFromMetropolis <= 0.361
gini = 0.486
samples = 31789
value = [13269, 18520]
gini = 0.437
samples = 6077
value = [4114, 1963]
gini = 0.452
samples = 19249
value = [6657, 12592]
gini = 0.499
samples = 12540
value = [6612, 5928]

K Nearest Neighbors



Optimal K Value Determination:

The accuracy graph suggests that the KNN model's performance varies with different K values (number of neighbors).
The model's accuracy improves significantly as the number of neighbors increases from 2 to around 10, after which the improvements in accuracy plateau.
The optimal K value indicated is 19, where the model achieves the highest accuracy before the curve flattens out, suggesting diminishing returns with additional neighbors.



Confusion Matrix Analysis:

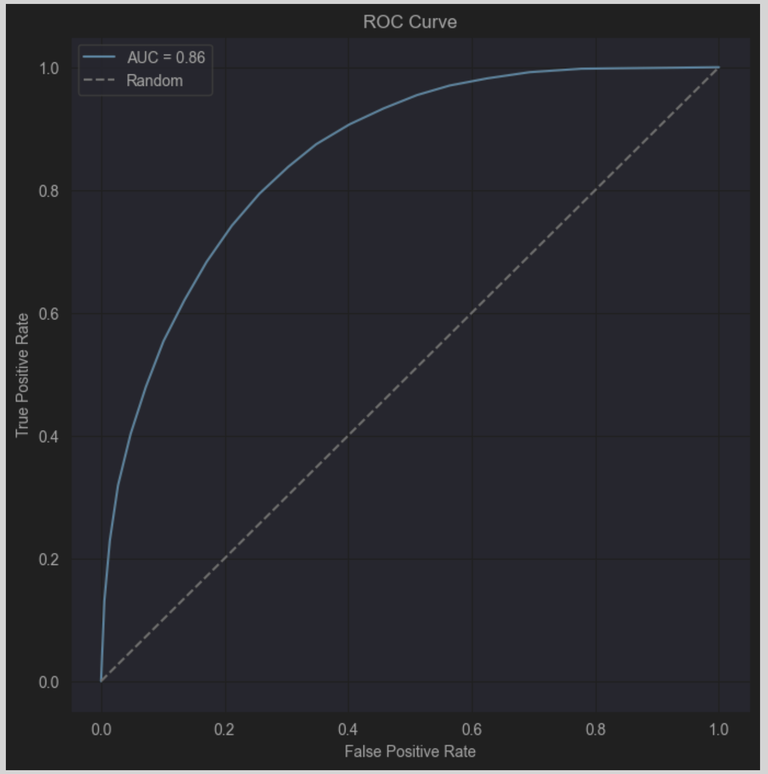
The confusion matrix shows the number of correct and incorrect predictions for two classes (0 and 1).
True Positive (TP): The model correctly predicted 15,966 instances of class 1.
True Negative (TN): The model correctly predicted 14,793 instances of class 0.
False Positive (FP): The model incorrectly predicted 5,099 instances as class 1.
False Negative (FN): The model incorrectly predicted 4,142 instances as class 0.

K Nearest Neighbors

Model Performance Metrics:
Precision: 0.77, indicating that when the model predicts class 1, it is correct about 77% of the time.
Recall (Sensitivity): 0.77, meaning the model correctly identifies 77% of actual class 1 instances.
Specificity: 0.77, the model's ability to identify true negatives is also at 77%.
F-score: 0.77, which is the harmonic mean of precision and recall, suggests a balanced performance between the model's precision and recall.

ROC Curve and AUC:
The ROC curve, which plots the true positive rate against the false positive rate, indicates the trade-offs between true positive and false positive rates as the threshold changes.
The AUC (Area Under the Curve) is 0.86, which is quite high and suggests that the model has a good measure of separability and is capable of distinguishing between the two classes well.

Cross-Validation Results:
The cross-validation results are presented, showing individual fold accuracies and a mean accuracy of 0.770625.
The individual fold accuracies are relatively consistent, indicating stable performance across different subsets of the dataset.




```
Cross-Validation Results:
Individual Fold Accuracies: [0.76675, 0.7714375, 0.76853125, 0.77459375, 0.7725]
Mean Accuracy: 0.7707625
```

```
Precision: 0.77
Recall (Sensitivity): 0.77
Specificity: 0.77
F-score: 0.77
```

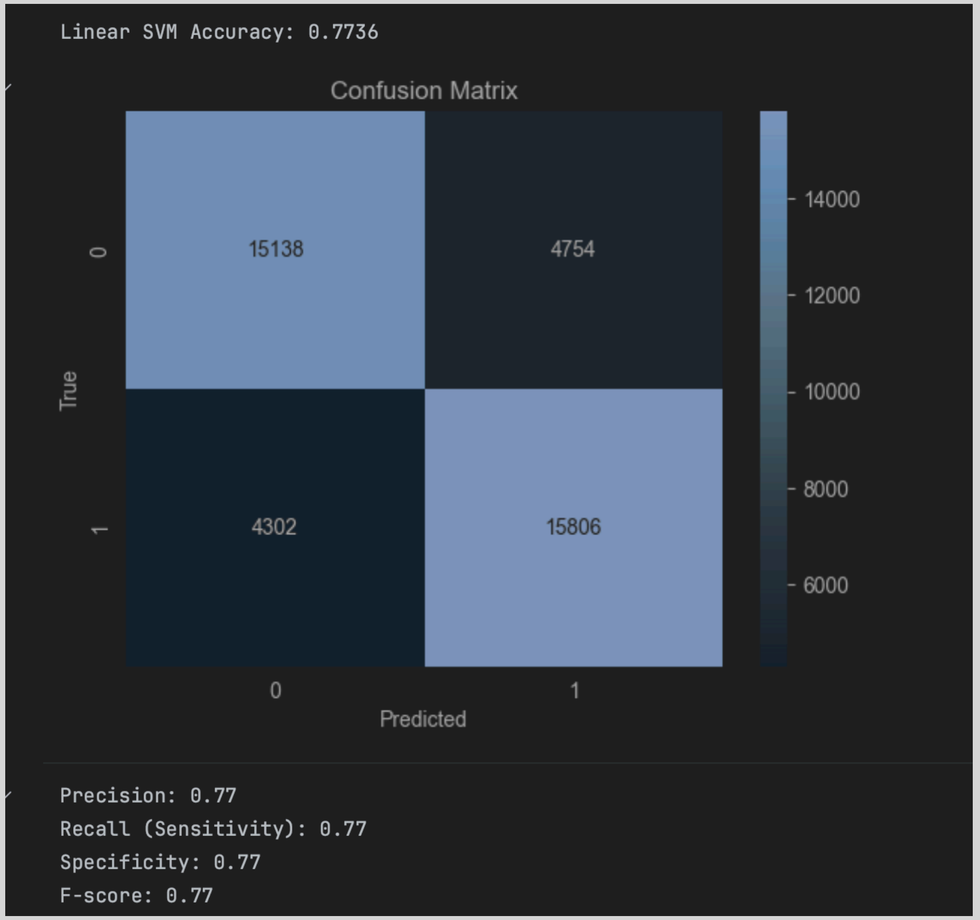
Support Vector Machine (SVM)

Linear Kernel

SVM Model with Linear Kernel Performance:
The reported overall accuracy of the linear SVM model is 0.7736.

Confusion Matrix:
The model predicted 15,806 instances correctly as class 1 (True Positives) and 15,138 instances correctly as class 0 (True Negatives).
There are 4,754 instances that were incorrectly predicted as class 1 (False Positives) and 4,302 instances incorrectly predicted as class 0 (False Negatives).

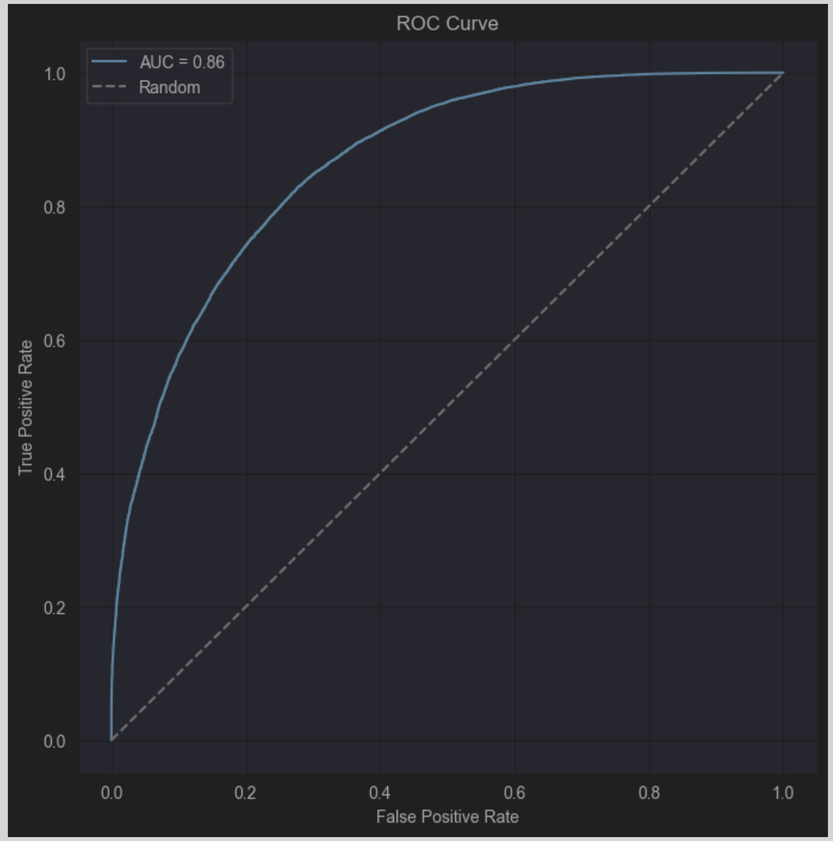
Model Evaluation Metrics:
Precision: 0.77, indicating that when the model predicts an instance as class 1, it is correct 77% of the time.
Recall (Sensitivity): 0.77, showing that the model identifies 77% of all actual class 1 instances correctly.
Specificity: 0.77, which means the model correctly identifies 77% of all actual class 0 instances.
F-score: 0.77, suggesting a balanced performance between precision and recall.



Support Vector Machine (SVM)

Linear Kernel

ROC Curve and AUC:
The ROC curve presents a visual representation of the model's capability to differentiate between the classes at various threshold levels.
The Area Under the Curve (AUC) for the ROC is 0.86, which indicates a strong ability of the model to distinguish between the positive and negative classes.



Cross-Validation Results:
Individual fold accuracies are listed, which appear to be consistent, varying slightly around the 0.77 mark.
The mean accuracy from cross-validation is 0.77548125, affirming the model's stable performance across different subsets of the data.

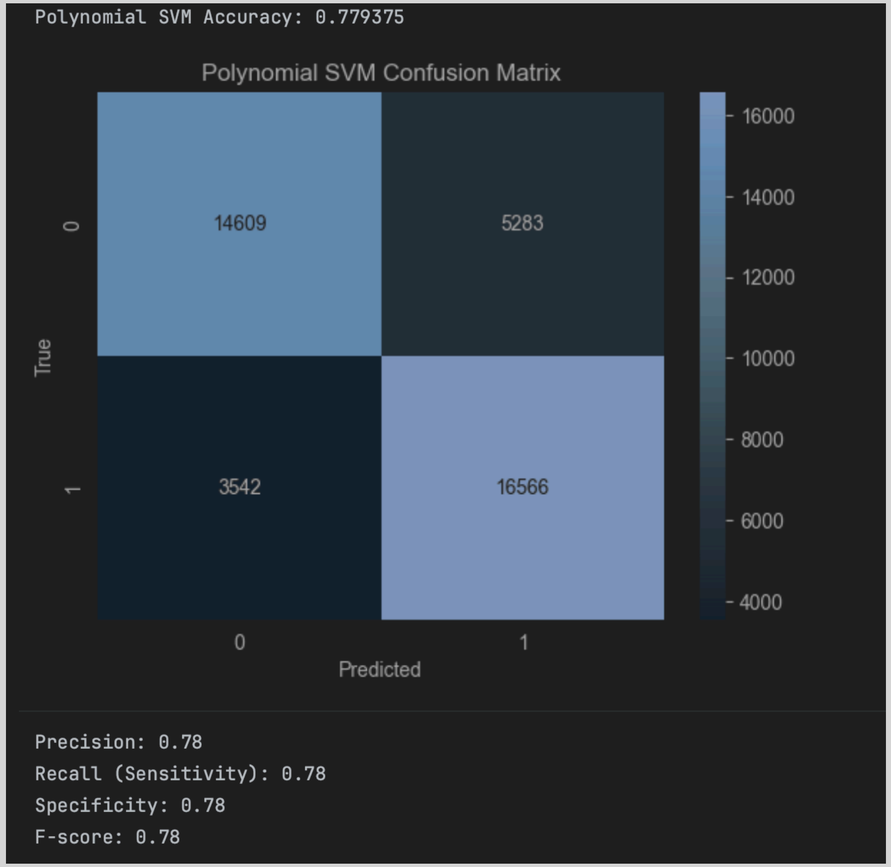
```
Cross-Validation Results:
Individual Fold Accuracies: [0.7720625, 0.77784375, 0.7725, 0.77815625, 0.77684375]
Mean Accuracy: 0.77548125
```

Polynomial Kernel

Polynomial SVM Performance:
The model's accuracy is 0.779375, which is a measure of its overall correct predictions.

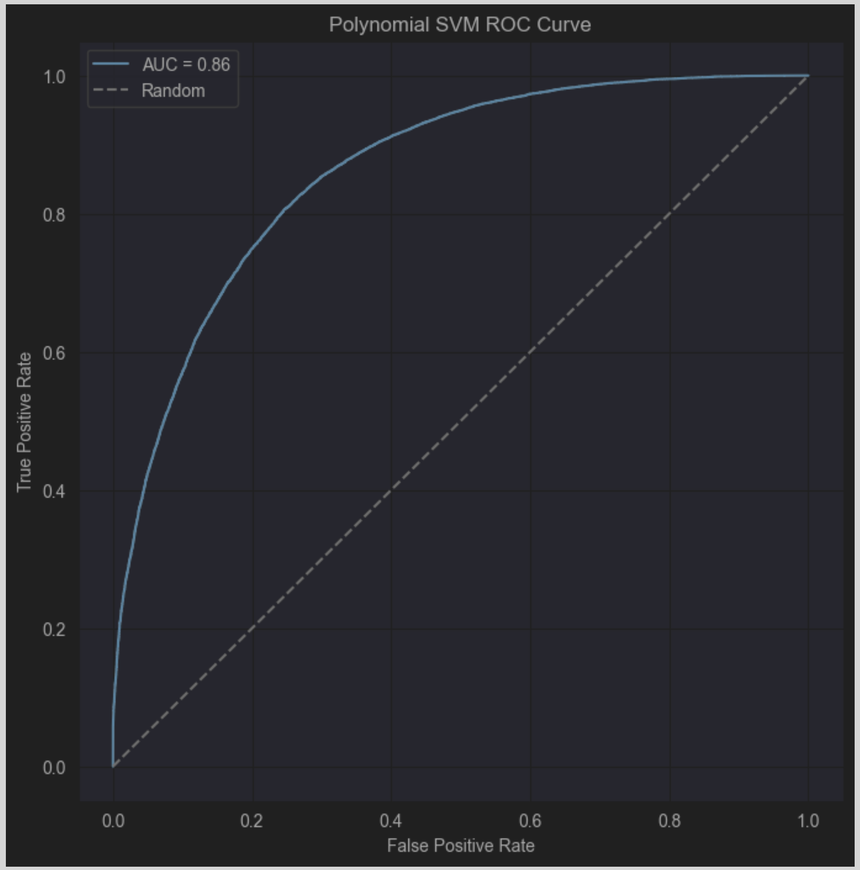
Confusion Matrix:
True Positives (TP): The model correctly predicted 16,566 instances as class 1.
True Negatives (TN): The model correctly predicted 14,609 instances as class 0.
False Positives (FP): The model incorrectly predicted 5,283 instances as class 1.
False Negatives (FN): The model incorrectly predicted 3,542 instances as class 0.

Performance Metrics:
Precision: 0.78, suggesting that when the model predicts an instance as class 1, it is correct 78% of the time.
Recall (Sensitivity): 0.78, indicating the model's ability to identify 78% of all actual class 1 instances correctly.
Specificity: 0.78, showing the model's ability to identify 78% of all actual class 0 instances.
F-score: 0.78, the harmonic mean of precision and recall, implying a balanced classification performance.



Polynomial Kernel

ROC Curve and AUC:
The ROC curve, which illustrates the true positive rate against the false positive rate at various threshold settings, shows that the model has a strong capability to differentiate between the classes.
The Area Under the Curve (AUC) is 0.86, indicating a very good level of discrimination by the model.



Cross-Validation Results:
Individual fold accuracies are listed, showing variations but with a close range around the mean accuracy.
The mean accuracy from cross-validation is 0.7780125, demonstrating the model's consistent performance across different data subsets.

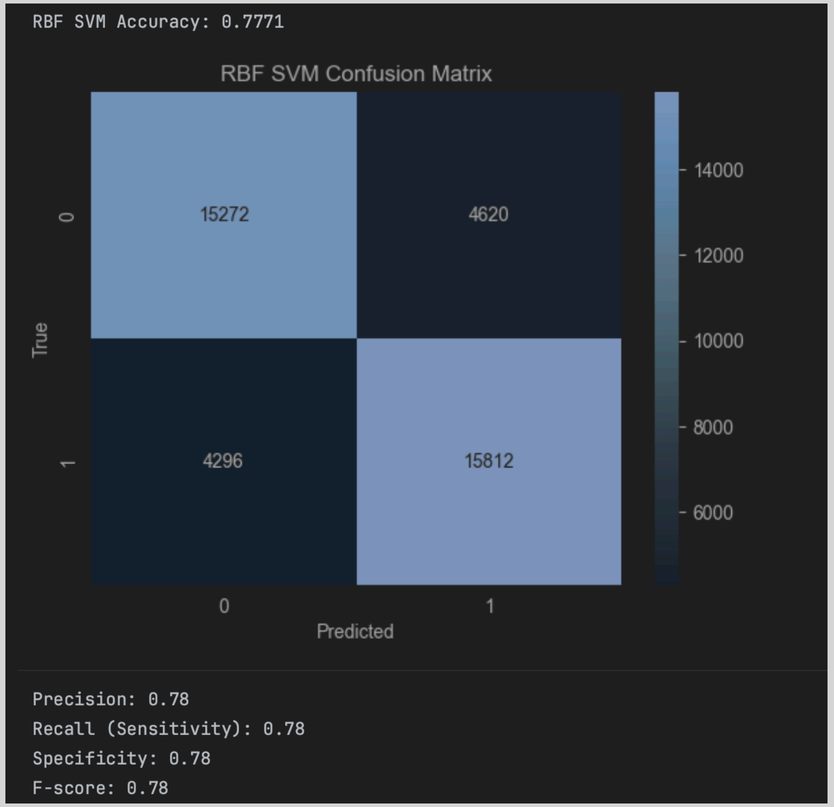
```
Cross-Validation Results:
Individual Fold Accuracies: [0.7738125, 0.7773125, 0.7779375, 0.7810625, 0.7799375]
Mean Accuracy: 0.7780125
```

Radial Basis Function Kernel

Accuracy and Confusion Matrix:

The SVM model with RBF kernel has an accuracy of 0.7771.
The confusion matrix shows that the model correctly predicted 15,812 instances as class 1 (True Positives) and 15,272 as class 0 (True Negatives).
There are 4,620 instances that were incorrectly predicted as class 1 (False Positives), and 4,296 instances incorrectly predicted as class 0 (False Negatives).
Performance Metrics:

Precision: 0.78, suggesting that 78% of the instances predicted as class 1 are indeed class 1.
Recall (Sensitivity): 0.78, indicating that the model correctly identifies 78% of all actual class 1 instances.
Specificity: 0.78, which is the proportion of actual class 0 instances that were correctly identified as class 0 by the model.
F-score: 0.78, which is the harmonic mean of precision and recall, showing a balanced classification performance.



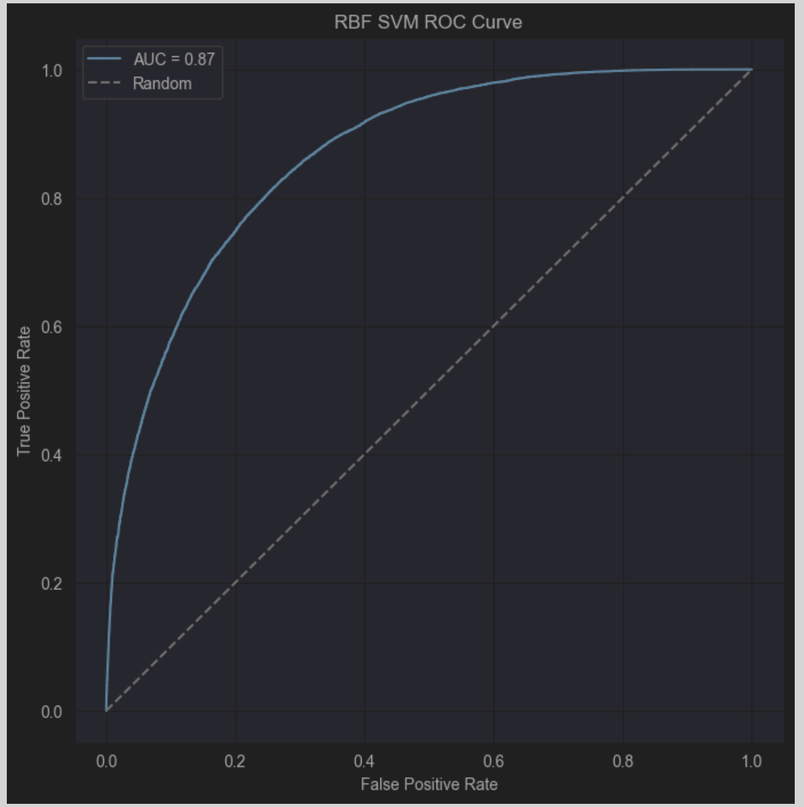
Radial Basis Function Kernel

ROC Curve:

The ROC curve shows the model's true positive rate against the false positive rate.
The AUC for the RBF SVM is 0.87, indicating a very good level of discrimination by the model, able to distinguish between the positive and negative classes effectively.

Cross-Validation:

Individual fold accuracies range from 0.771625 to 0.778, which indicates the model has a stable performance across different subsets of the data.
The mean accuracy from cross-validation is 0.774975, further confirming the model's consistent performance.



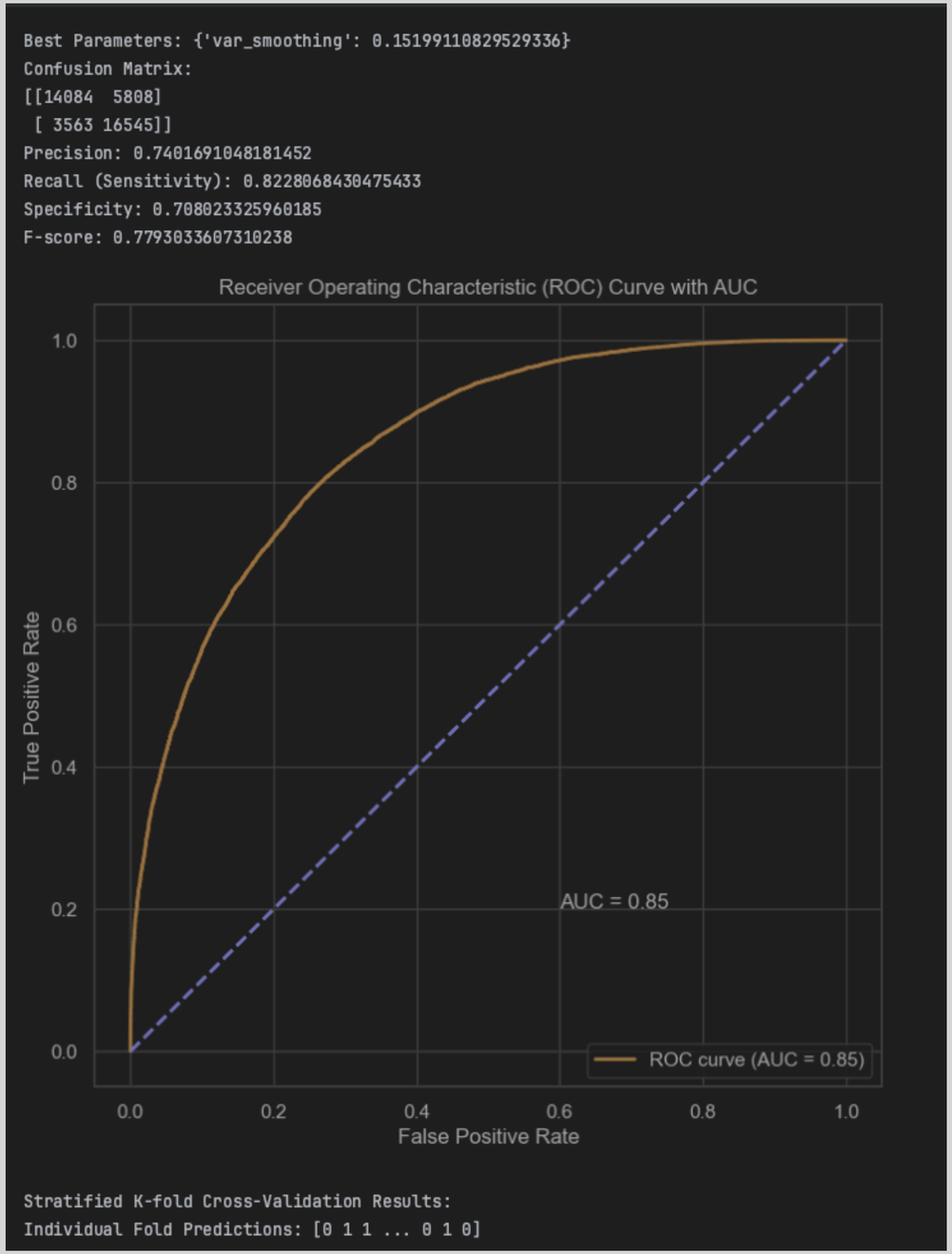

```
Cross-Validation Results:
Individual Fold Accuracies: [0.771625, 0.77621875, 0.772375, 0.778, 0.77665625]
Mean Accuracy: 0.7749750000000001
```

Grid Search

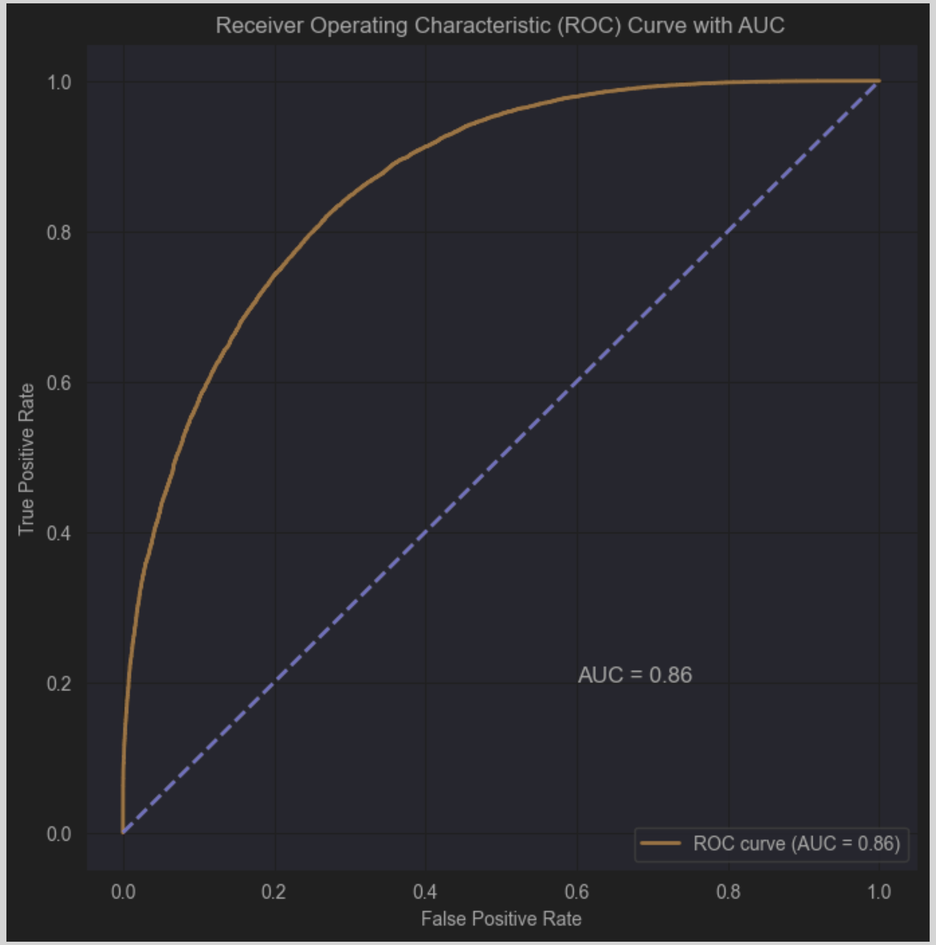
Best Parameters and Confusion Matrix:
The grid search has determined the best parameters for the SVM model with a polynomial kernel.
The model achieved a high degree of accuracy, as indicated by the confusion matrix, correctly identifying 16,545 instances as class 1 and 14,684 as class 0.

Performance Metrics:
Precision: Approximately 0.746, indicating a good proportion of positive identifications was actually correct.
Recall (Sensitivity): About 0.822, demonstrating the model's strength in identifying most of the positive instances.
Specificity: Approximately 0.780, which shows the model's effectiveness in identifying negative instances.
F-score: Around 0.779, suggesting a good balance between precision and recall.

ROC Curve:
The ROC curve visually represents the model's capability to differentiate between the classes at various threshold settings.
The AUC is 0.85, which is quite high and indicates that the model has a good measure of separability.



Logistic Regression



Logistic Regression

Training and Testing Accuracy:
The logistic regression model has a training accuracy of about 0.775 and a testing accuracy of 0.72785.

Confusion Matrix:
The model correctly predicted 15,656 instances as class 1 and 15,259 as class 0.
However, there are 4,452 false negatives and 4,633 false positives.

Performance Metrics:
Precision, recall, and F1-score are all 0.77 for both classes, indicating a balanced model performance.
The accuracy of the model on the testing set is 0.72785.
Specificity is approximately 0.767, suggesting the model's ability to correctly identify negative instances is somewhat less than its ability to identify positive instances.

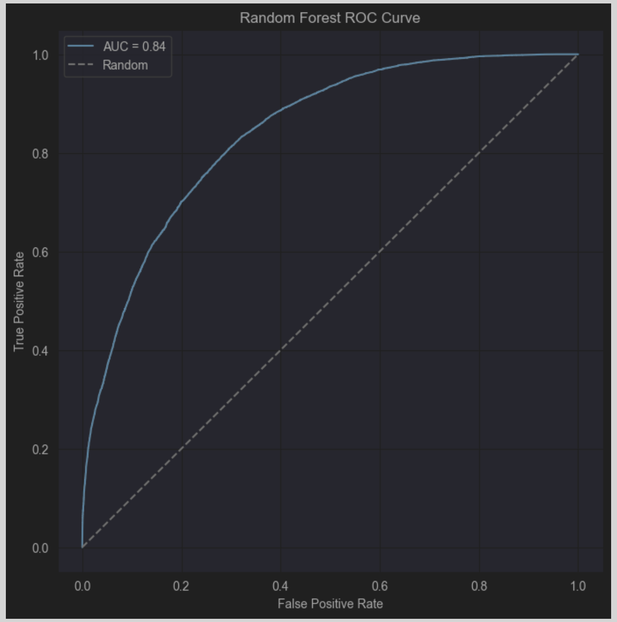
```
Best Parameters: {'C': 0.01, 'penalty': 'l2'}
Training Accuracy: 0.77506875
Testing Accuracy: 0.772875
Confusion Matrix:
[[15259  4633]
 [ 4452 15656]]

Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.77      0.77     19892
           1       0.77      0.78      0.78     20108

    accuracy                           0.77     40000
   macro avg       0.77      0.77      0.77     40000
weighted avg       0.77      0.77      0.77     40000


Individual Metrics:
Precision: [0.77413627 0.77164966]
Recall: [0.7670923  0.77859558]
F1-Score: [0.77059819 0.77510706]
Accuracy: 0.772875
Specificity: 0.7670922984114217
```

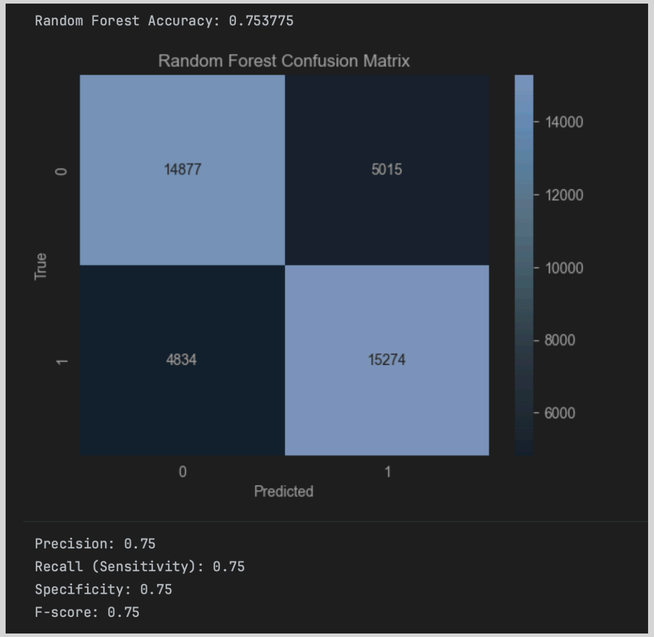
Random Forest

Accuracy and Confusion Matrix:
The accuracy of the random forest model is 0.75375.
The confusion matrix indicates that the model correctly predicted 15,274 instances as class 1 and 14,877 as class 0.
There are 5,015 instances incorrectly labeled as class 1 and 4,834 as class 0.

Performance Metrics:
Precision, recall, and F1-score are all equal to 0.75, showing a consistent performance across these metrics.
Specificity is also 0.75, indicating equal performance in identifying both classes.

ROC Curve:
The ROC curve has an AUC of 0.84, which is good and suggests the model is fairly effective at distinguishing between classes.





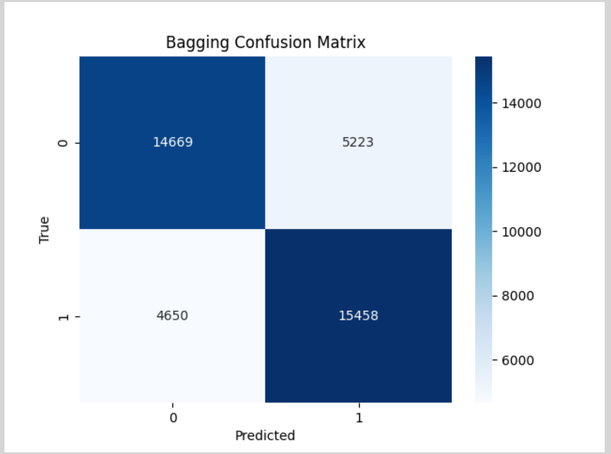
Bagging

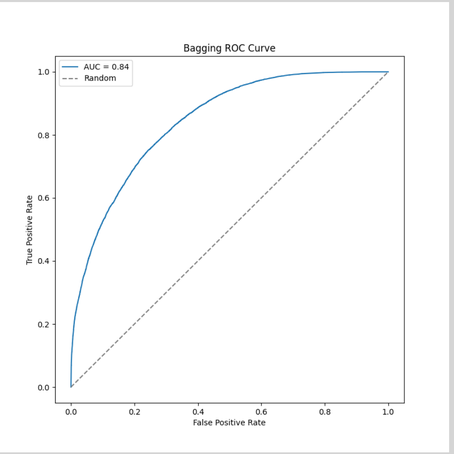
Accuracy and Confusion Matrix:
The bagging model has an accuracy of 0.753175.
The confusion matrix shows 15,458 true positive predictions and 14,669 true negative predictions.
There are 5,223 false positives and 4,650 false negatives.

Performance Metrics:
Precision, recall, specificity, and F-score are uniformly reported at 0.75, indicating balanced performance across these metrics.

ROC Curve:
The ROC curve displays a good true positive rate against the false positive rate with an AUC of 0.84, suggesting competent discriminative ability.

Cross-Validation Results:
The cross-validation accuracies are consistent, ranging from 0.751875 to 0.7648125, with a mean accuracy of 0.7493125.






```
Bagging Accuracy: 0.753175
Precision: 0.75
Recall (Sensitivity): 0.75
Specificity: 0.75
F-score: 0.75
```

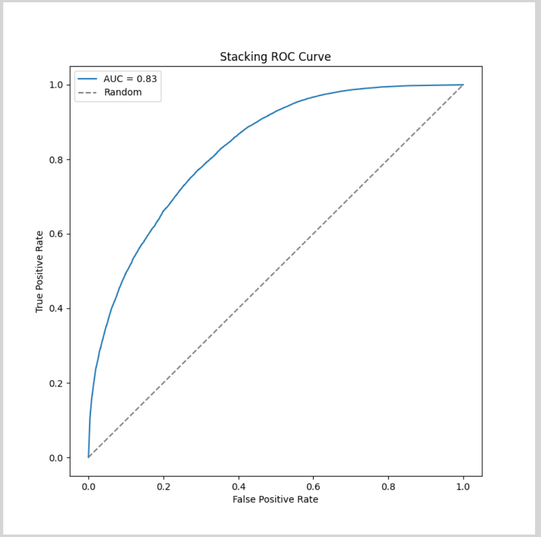
```
Cross-Validation Results:
Individual Fold Accuracies: [0.751875, 0.7505625, 0.74809375, 0.75234375, 0.74678125]
Mean Accuracy: 0.74993125
```

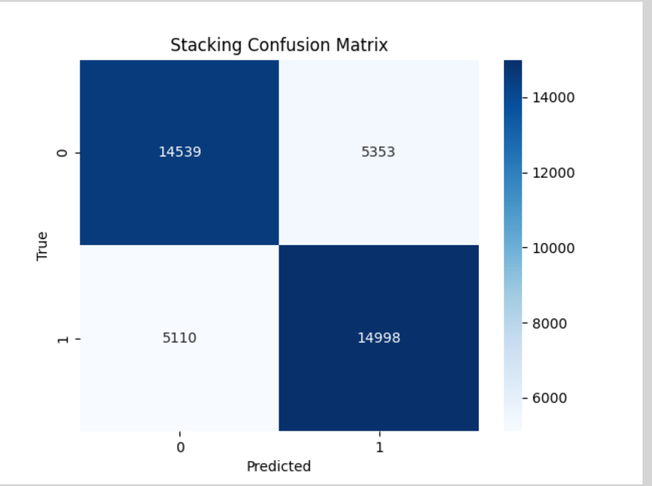
Stacking

Accuracy and Confusion Matrix:
The stacking model's accuracy is 0.738425.
The confusion matrix indicates the model predicted 14,998 true positives and 14,539 true negatives.
The model has 5,353 false positives and 5,110 false negatives.

Performance Metrics:
Precision, recall, specificity, and F-score are reported at 0.74, suggesting a slightly less balanced performance compared to the bagging model.

ROC Curve:
The ROC curve for the stacking model has an AUC of 0.83, which is marginally lower than the bagging model but still indicates good classification ability.

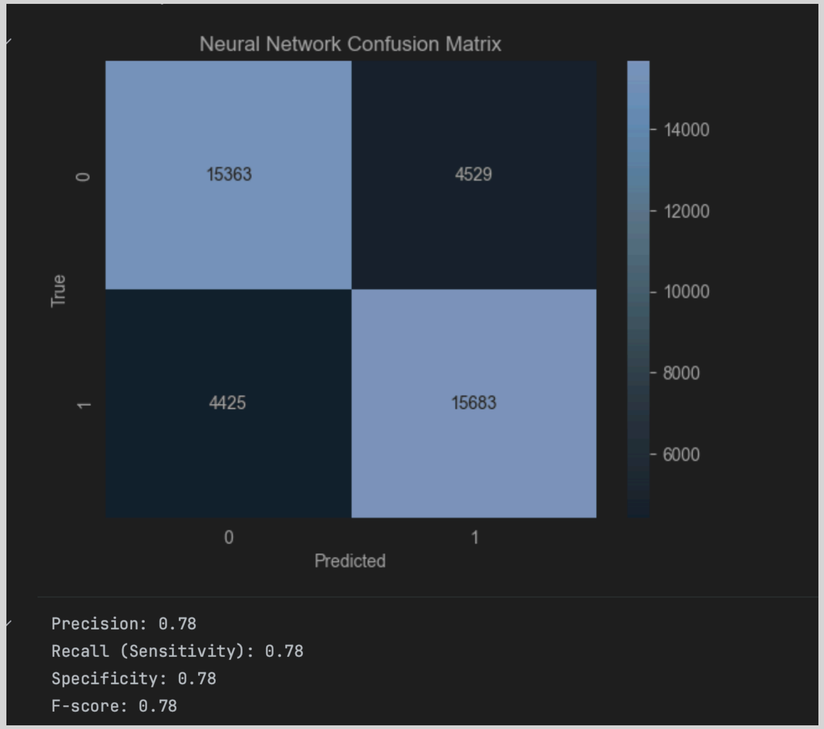




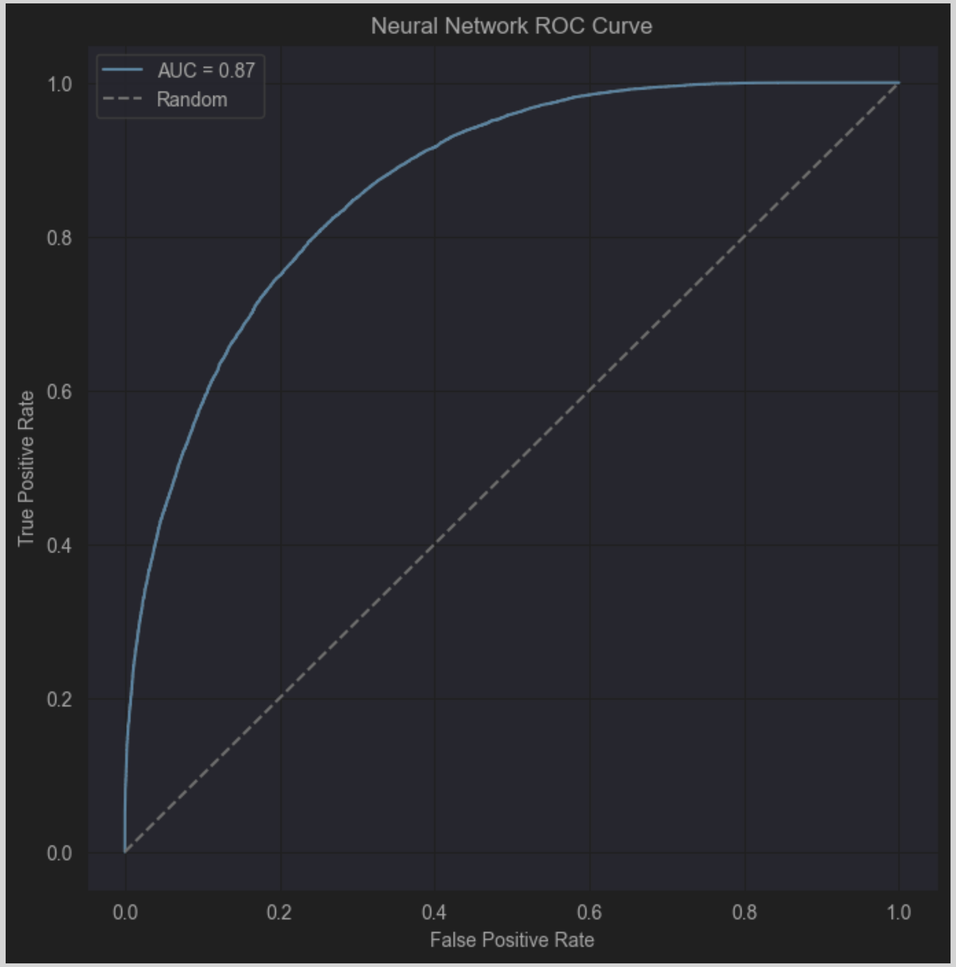

```
Stacking Accuracy: 0.738425
Precision: 0.74
Recall (Sensitivity): 0.74
Specificity: 0.74
F-score: 0.74
```

Neural Networks

```
  warnings.warn(
5000/5000 [==============================] - 2s 426us/step - loss: 0.4309 - accuracy: 0.7840 - val_loss: 0.4450 - val_accuracy: 0.7783
Epoch 45/50
5000/5000 [==============================] - 2s 428us/step - loss: 0.4309 - accuracy: 0.7845 - val_loss: 0.4442 - val_accuracy: 0.7779
Epoch 46/50
5000/5000 [==============================] - 2s 429us/step - loss: 0.4308 - accuracy: 0.7842 - val_loss: 0.4451 - val_accuracy: 0.7776
Epoch 47/50
5000/5000 [==============================] - 2s 425us/step - loss: 0.4309 - accuracy: 0.7832 - val_loss: 0.4446 - val_accuracy: 0.7777
Epoch 48/50
5000/5000 [==============================] - 2s 426us/step - loss: 0.4307 - accuracy: 0.7843 - val_loss: 0.4451 - val_accuracy: 0.7779
Epoch 49/50
5000/5000 [==============================] - 2s 427us/step - loss: 0.4305 - accuracy: 0.7840 - val_loss: 0.4445 - val_accuracy: 0.7800
Epoch 50/50
5000/5000 [==============================] - 2s 427us/step - loss: 0.4306 - accuracy: 0.7840 - val_loss: 0.4467 - val_accuracy: 0.7761
1250/1250 [==============================] - 0s 254us/step
MLP Accuracy: 0.77615
```




```
Precision: 0.78
Recall (Sensitivity): 0.78
Specificity: 0.78
F-score: 0.78
```



Neural Networks

Neural Network Model Performance During Training:

The neural network training process shows increasing accuracy over epochs, with the final accuracy on the validation set reaching as high as 0.7800.
The loss on both the training and validation sets decreases over epochs, suggesting that the model is learning effectively.
Now for the performance metrics based on the neural network's validation:

Neural Network Validation Performance:

Confusion Matrix:
The confusion matrix for the neural network shows 15,683 true positives and 15,363 true negatives.
There are 4,529 false positives and 4,425 false negatives.

Performance Metrics:
The precision, recall, specificity, and F-score all score at 0.78, indicating a strong and balanced performance across these metrics.

ROC Curve:
The ROC curve for the neural network has an AUC of 0.87, showcasing an excellent ability to discriminate between the classes.

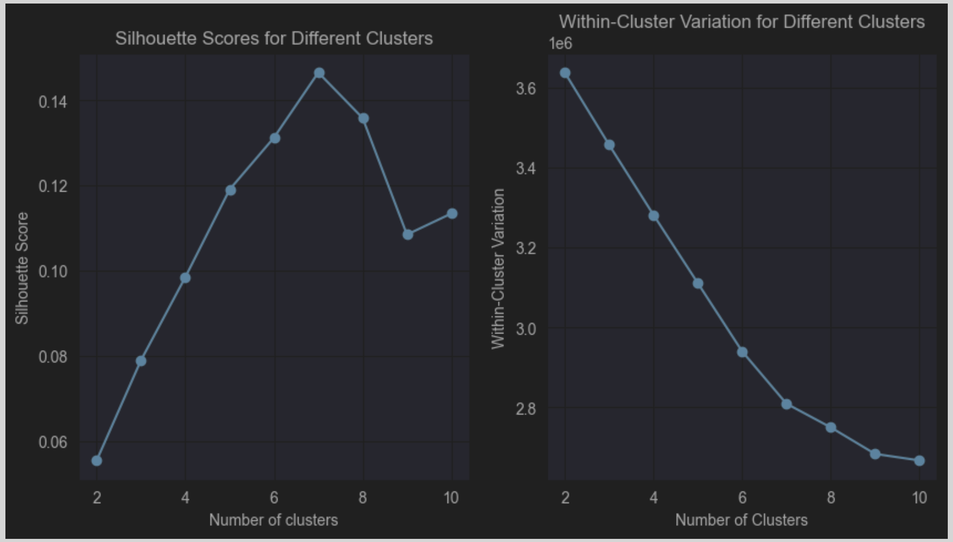
Phase 4

Clustering and Association

K Mean Clustering

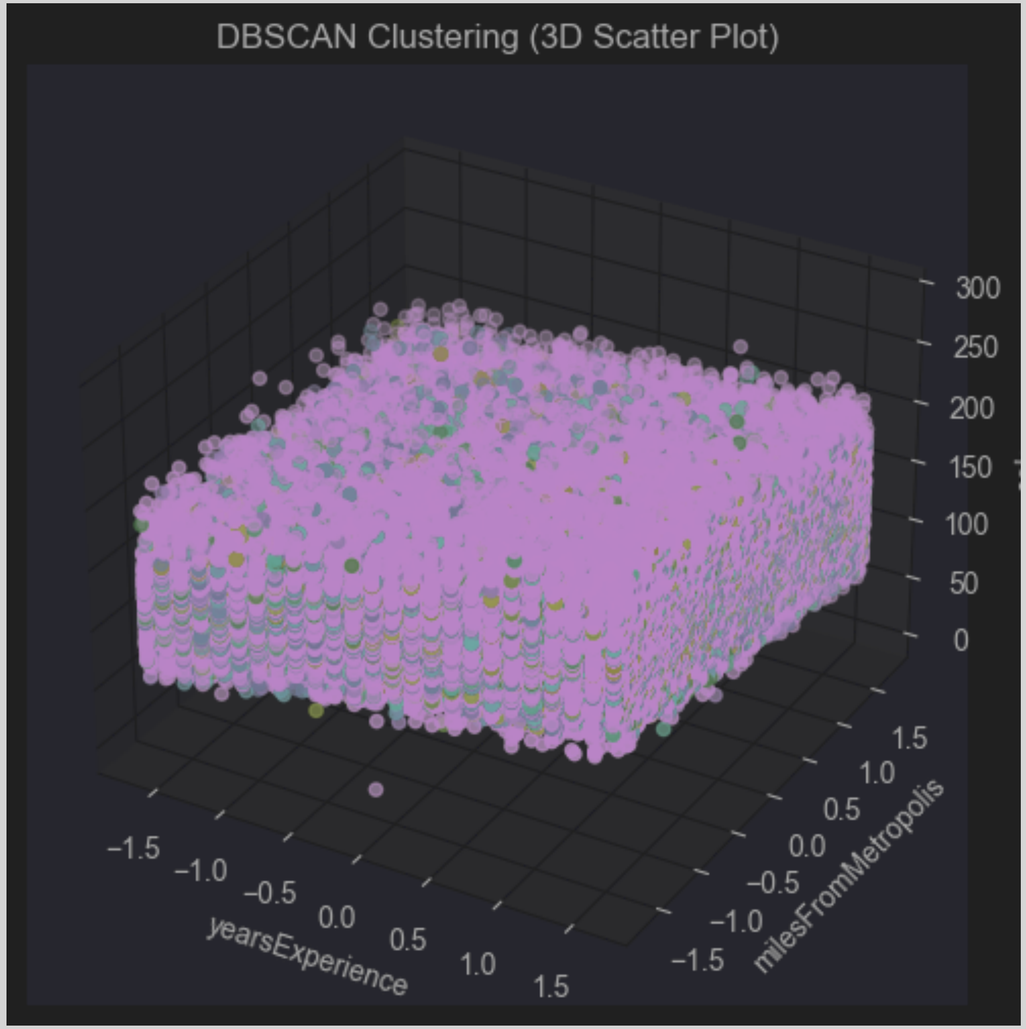
Silhouette Scores for Different Clusters:
This plot would typically show the silhouette scores for various numbers of clusters. The silhouette score measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). A higher silhouette score indicates that objects are well matched to their own cluster and poorly matched to neighboring clusters.
The optimal number of clusters is often at the peak of the silhouette score before it starts to decrease, as this indicates the point at which adding more clusters does not provide better cohesion and separation.

Within-Cluster Variation for Different Clusters:
This graph usually presents the within-cluster sum of squares (WCSS), which is a measure of the variance within each cluster. As the number of clusters increases, the WCSS should generally decrease because the clusters are smaller and more compact.
The "elbow method" is often used to select the optimal number of clusters, which is the point where the decrease in WCSS begins to level off, indicating that adding more clusters does not significantly improve the compactness of the clustering.



DB Scan Clustering

- DBSCAN is a density-based clustering algorithm that groups points that are closely packed together and marks points in low-density regions as outliers.
- The plot would typically show different clusters in unique colors and outliers in a different marker, usually shown in black.
- DBSCAN does not require specifying the number of clusters beforehand, making it useful for datasets with complex shapes and varying densities.



Apriori Algorithm

```
----------------- Apriori --------------------------------
[(frozenset({'yearsExperience_-1.268496339526427'}), 0.03899375), (frozenset({'jobType_CFO_0.0'}), 0.85555), (frozenset({'jobType_CTO_0.0'}),
0.8536), (frozenset({'jobType_JUNIOR_1.0'}), 0.13551875), (frozenset({'jobType_MANAGER_0.0'}), 0.8585875), (frozenset({'jobType_SENIOR_0.0'}),
0.85933125), (frozenset({'jobType_VICE_PRESIDENT_0.0'}), 0.8565625), (frozenset({'degree_DOCTORAL_0.0'}), 0.6654625), (frozenset
({'degree_MASTERS_1.0'}), 0.33395), (frozenset({'major_BUSINESS_0.0'}), 0.8737125)]
```

- The algorithm identifies the individual items (like 'jobType_CFO', 'degree_DOCTORAL', etc.) that appear frequently within the dataset and also the combination of items that appear often together.
- The output shows each item or itemset along with their corresponding support values, which is the proportion of transactions in the database that contain the itemset.
- In market basket analysis, this information is used to find items that are commonly bought together, allowing retailers to make decisions on product placement, marketing, etc.

Final Model Selection

Polynomial Kernel SVM Model