MAS456 Final Project

20200639 Woojin Chae

Objectives

disc_hire
Response to the question, "Have you ever experienced discrimination in getting hired?"

To identify…

1. Which factors are strongly associated with experience of discrimination in hiring?

2. Whether the experience of discrimination stands out in a particular group

Variable	Description	Possible Answers
disc_hire	Response to the question,question, “Have you ever experienced discrimination in getting hired?”	0: 'No', 1: 'Yes', NA: 'Not Applicable'
gender	Gender	0: male, 1: female
age	Age	0: 16~24, 1: 25~34, 2: 35~44, 3: 45~54, 4: 55~64, 5: 65 years old
edu_cat	Educational Level	0: middle school graduate or less, 1: high school graduate, 2: college graduate or more
marriage	Marital status	0: never married, 1: currently married, 2: previously married
emp_fin	Employment status	0: permanent, 1: non permanent
income_quartile	Total household income divided by the square root of the number of household members	0: Q1, 1: Q2, 3: Q3, 4: Q4
birth_region	Birth region	1: Jeolla do, 0: other regions
health	Response to the question, "How would you rate your health?	0: 'very good', 1: 'good', 2: 'poor', 3: 'very poor'
disability	Response to the question "Do you have any impairment or disability?"	0: 'No, 1: 'Yes'
residence	Residential areas	1: 'Seoul', 2: 'Pusan', 3: 'Daegu', 4: 'Daejeon', 5: 'Incheon', 6: 'Gwangju', 7: 'Ulsan', 8: 'Kyunggi', 9: 'Kangwon', 10: 'Choongbuk', 11: 'Choongnam', 12: 'Jeonbuk', 13: 'Jeonnam', 14: 'Kyungbuk', 15: 'Kyungnam'
disc_wage	Experience of discrimination in receiving income	0: 'No', 1: 'Yes', 2: 'Not Applicable'
disc_jobedu	Experience of discrimination in training	0: 'No', 1: 'Yes', 2: 'Not Applicable'
disc_promotion	Experience of discrimination in getting promoted	0: 'No', 1: 'Yes', 2: 'Not Applicable'
disc_resign	Experience of discrimination in being fired	0: 'No', 1: 'Yes', 2: 'Not Applicable'
disc_edu	Experience of discrimination in obtaining higher education	0: 'No', 1: 'Yes', 2: 'Not Applicable'
disc_home	Experience of discrimination at home	0: 'No', 1: 'Yes', 2: 'Not Applicable'
disc_social	Experience of discrimination at general social activities	0: 'No', 1: 'Yes', 2: 'Not Applicable'

provided by KLIPS

1. Identifying and Analyzing Key Variables in Hiring Discrimination Experiences

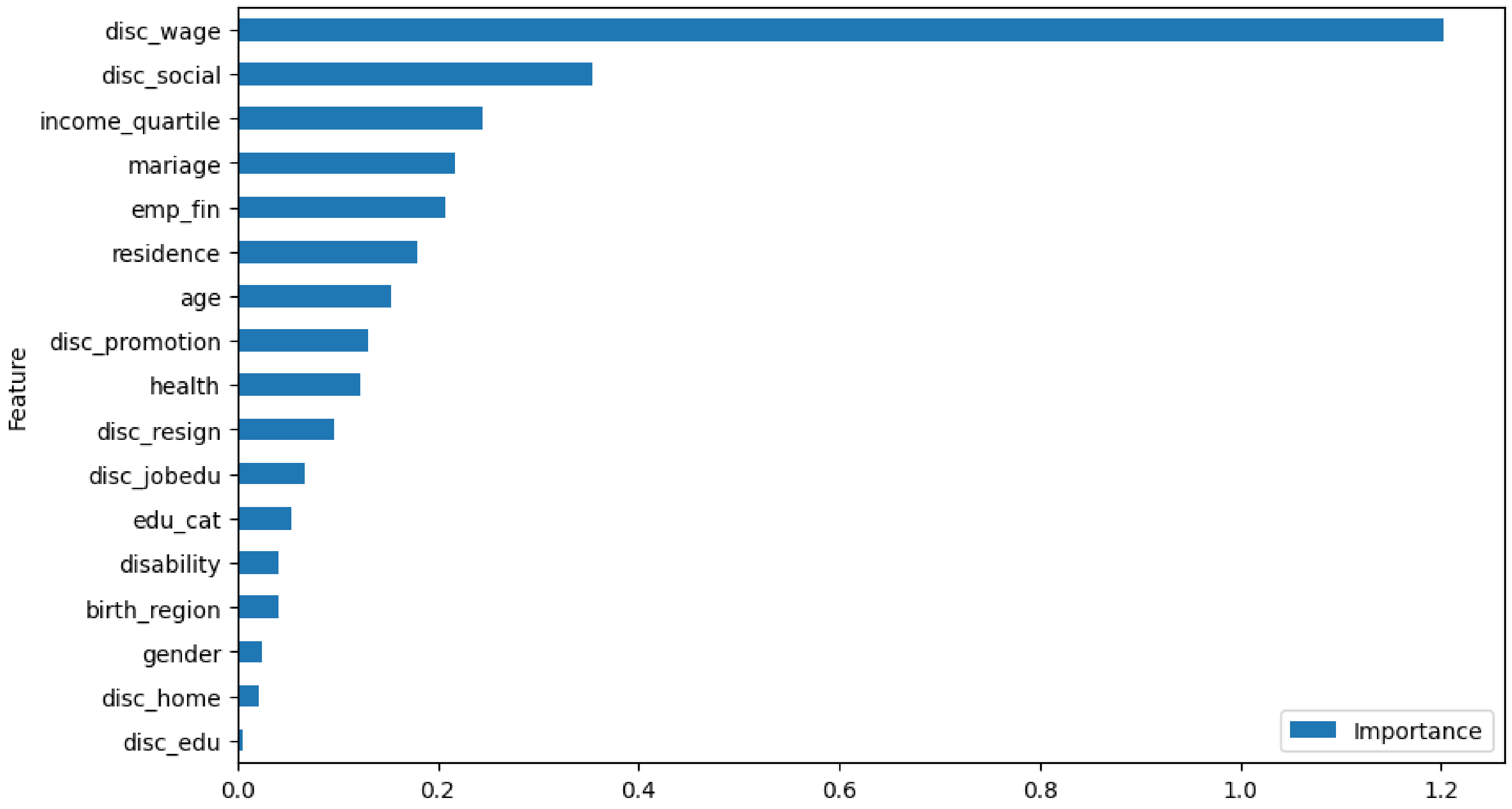
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix
import numpy as np

# Filtering the dataset for 'disc_hire' values of 0 or 1
filtered_data = data[data['disc_hire'].isin([0, 1])]

# Splitting the data into features (X) and target (y)
X = filtered_data.drop('disc_hire', axis=1)
y = filtered_data['disc_hire']


# Standardize predictors
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Creating the logistic regression model
logreg = LogisticRegression()

# Fitting the model with the training data
logreg.fit(X_scaled, y)

# Plot importance of the coefficients
coefficients = logreg.coef_[0]
feature_importance = pd.DataFrame({'Feature': X.columns, 'Importance': np.abs(coefficients)})
feature_importance = feature_importance.sort_values('Importance', ascending=True)
feature_importance.plot(x='Feature', y='Importance', kind='barh', figsize=(10, 6)).legend(loc='lower right')
```

Unify scale of features

:to make comparison of coefficients available

	Variable	Coefficient
10	disc_wage	1.202336
16	disc_social	0.353487
4	emp_fin	0.207148
9	residence	0.179417
1	age	0.152763
12	disc_promotion	0.131177
7	health	0.122233
13	disc_resign	0.096465
11	disc_jobedu	0.066369
8	disability	0.041251
14	disc_edu	-0.005665
15	disc_home	-0.022320
0	gender	-0.025431
6	birth_region	-0.041039
2	edu_cat	-0.053538
3	mariage	-0.217372
5	income_quartile	-0.244076

(a) Coefficients



(b) Importance Rank

```
print(len(data[(data['disc_wage']==2) & (data['disc_hire'].isna()==False)]))
print(len(data[(data['disc_social']==2) & (data['disc_hire'].isna()==False)]))
```

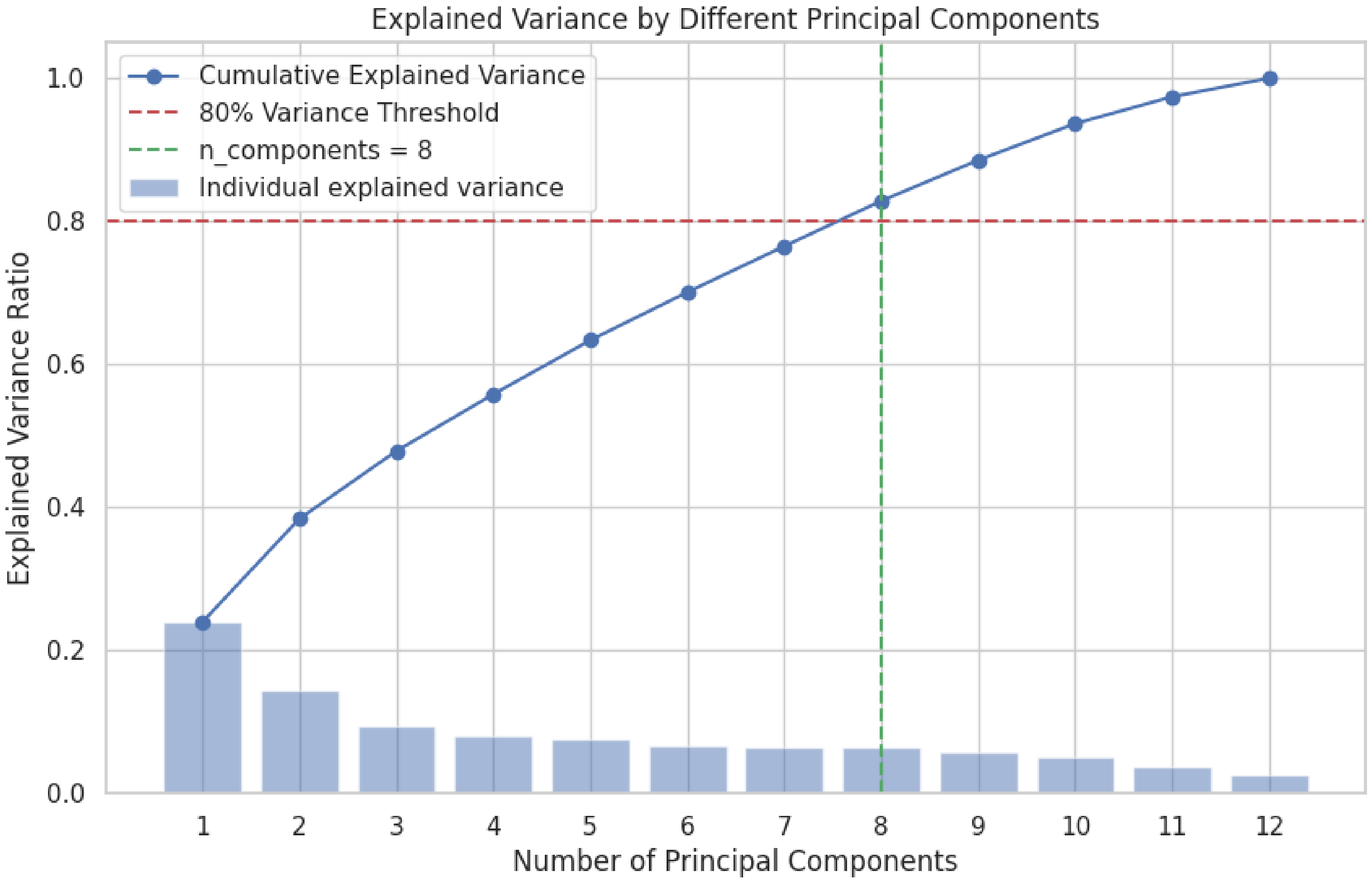
```
11
18
```

Caveat from bias can be ignored…

2. PCA of 12 Explanatory Variables Influencing Discrimination Experiences

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Selecting only the relevant columns for PCA
pca_columns = ['gender', 'age', 'edu_cat', 'emp_fin', 'income_quartile',
               'health', 'disability', 'disc_wage', 'disc_jobedu',
               'disc_promotion', 'disc_resign', 'disc_edu']
pca_data = filtered_data[pca_columns]

# Standardizing the data - important for PCA
scaler = StandardScaler()
pca_data_scaled = scaler.fit_transform(pca_data)

# Creating a PCA object
pca = PCA()

# Fitting PCA on the standardized data
pca.fit(pca_data_scaled)

# Explained variance ratio
explained_variance_ratio = pca.explained_variance_ratio_

# Choosing the number of components that captures at least 80% of total variance
cumulative_variance = explained_variance_ratio.cumsum()
n_components = (cumulative_variance < 0.80).sum() + 1

# Creating a PCA object with the selected number of components
pca_selected = PCA(n_components=n_components)
pca_selected.fit(pca_data_scaled)

# Principal components
principal_components = pca_selected.components_

n_components
```

Increase the number of principal components until 80% variance of the total data is captured

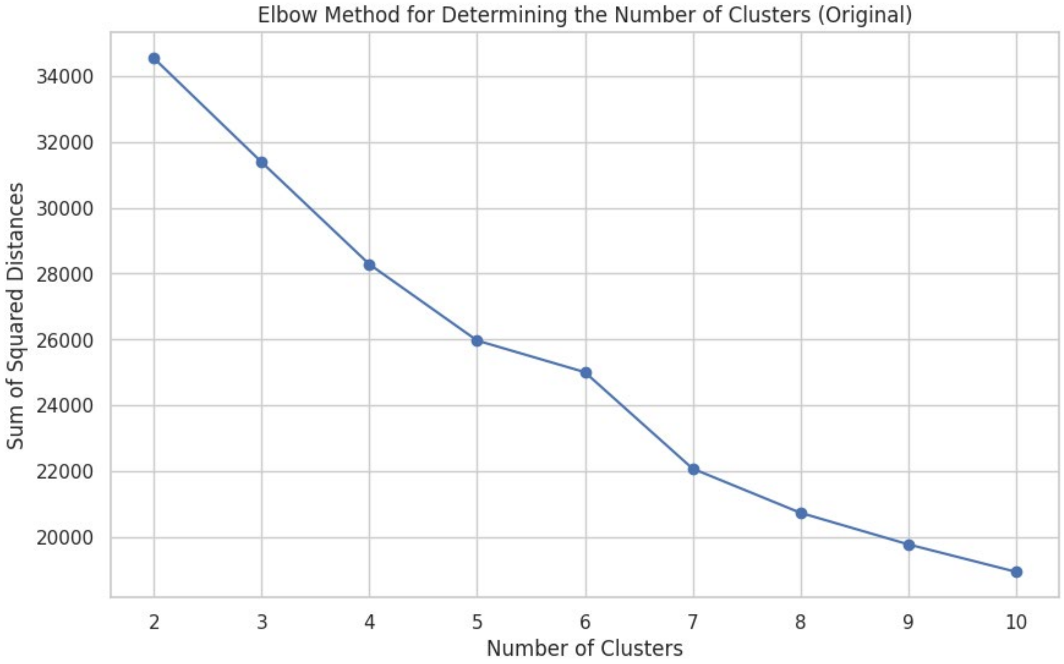

Explained Variance by Different Principal Components
Cumulative Explained Variance
80% Variance Threshold
n_components = 8
Individual explained variance
Explained Variance Ratio
0.0
0.2
0.4
0.6
0.8
1.0
1 2 3 4 5 6 7 8 9 10 11 12
Number of Principal Components

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
gender	0.089538	0.032022	-0.744339	0.074049	0.439272	0.244500	0.000159	0.292778
age	0.193179	-0.460230	0.288012	-0.419802	-0.043219	0.071368	-0.181086	0.256663
edu_cat	-0.317663	0.435727	0.043905	0.094564	0.119541	-0.197533	0.047350	-0.279522
emp_fin	0.258238	-0.268106	-0.324447	0.236912	-0.254659	-0.272319	-0.342972	0.049566
income_quartile	-0.269721	0.226450	0.123351	-0.417678	0.386347	0.091562	-0.515944	0.257841
health	0.161455	-0.370171	-0.063349	-0.196172	0.540831	-0.234417	-0.012612	-0.659036
disability	0.096011	-0.149119	0.449821	0.698306	0.451850	0.037794	-0.116607	0.220415
disc_wage	0.310777	0.122292	0.110413	0.007341	-0.010445	0.796374	0.038444	-0.322754
disc_jobedu	0.442208	0.256614	0.018209	-0.053637	-0.070566	-0.059792	-0.113748	-0.018951
disc_promotion	0.441883	0.296304	0.005844	-0.029094	-0.061403	-0.105613	-0.172826	-0.019509
disc_resign	0.304390	0.365858	0.090480	-0.002840	0.097441	-0.246806	-0.273581	-0.035214
disc_edu	0.314624	0.111478	0.116208	-0.221039	0.248489	-0.217560	0.669515	0.335071

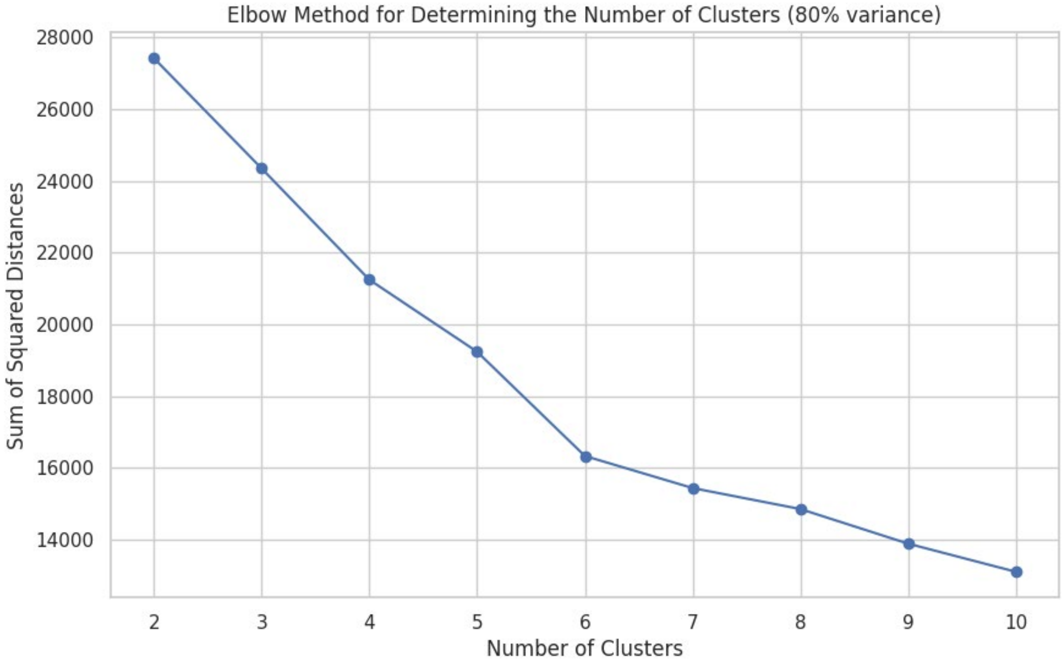
PC2: Changing Educational trends over generation

PC1: Exposure to high discrimination

3. Comparative Analysis of Subgroup Clustering Based on Original Variables and Principal Components

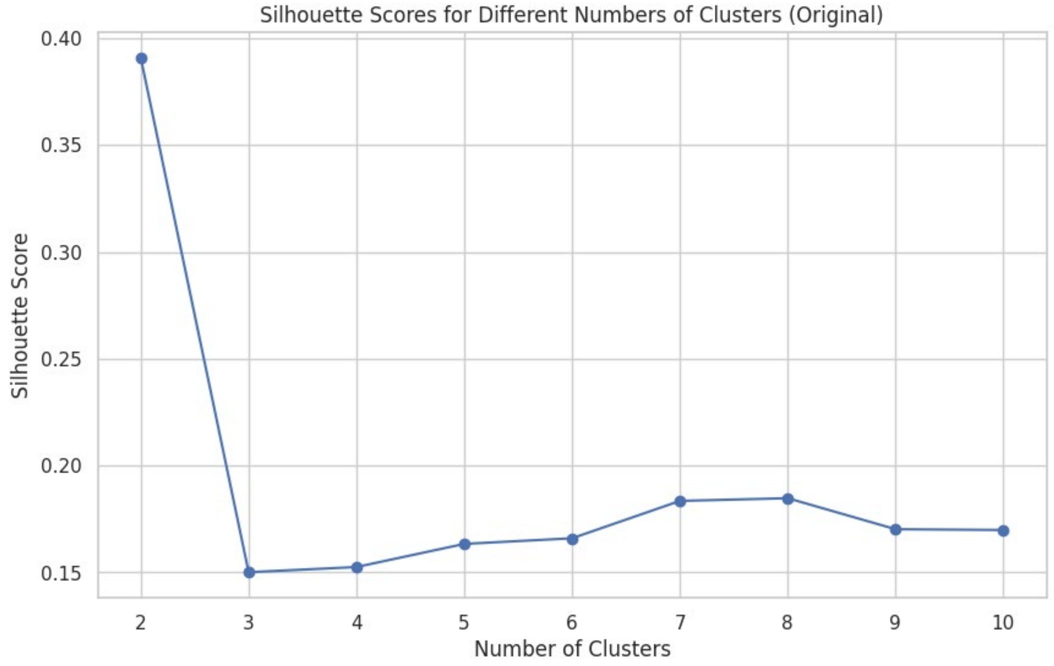


(a) sum of squared distance (original)

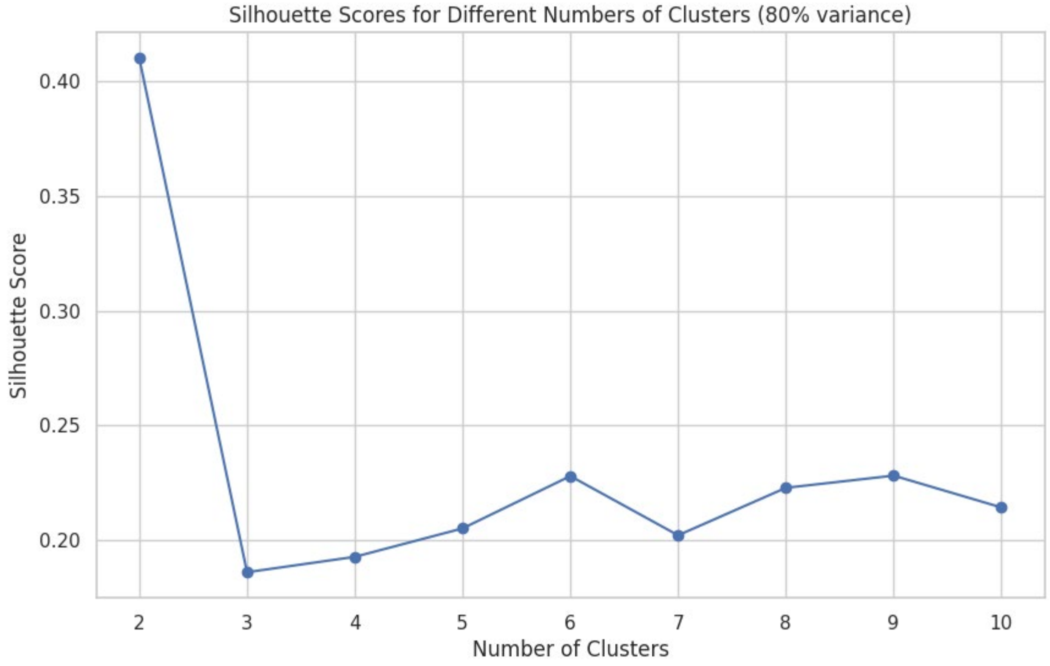


(b) sum of squared distance (PCs)

Hard to determine only considering this..!



(a) silhoutte score (original)



(b) silhoutte score (PCs)

Both case yields optimal number of cluster as 2!

Cluster_Original	0	1
gender	0.379458	0.507761
age	1.949141	2.197339
edu_cat	1.182299	0.713969
emp_fin	0.203765	0.452328
income_quartile	1.833223	1.177384
health	1.383091	1.503326
disability	0.022127	0.048780
disc_wage	0.104029	0.532151
disc_jobedu	0.021466	1.436807
disc_promotion	0.083884	1.651885
disc_resign	0.090159	1.066519
disc_edu	0.016843	0.536585

(a) cluster (Original)

Cluster_PCA	0	1
PC1	-0.520409	3.485130
PC2	-0.186704	1.250336
PC3	-0.016444	0.110124
PC4	0.016347	-0.109472
PC5	0.000633	-0.004241
PC6	0.028606	-0.191570
PC7	0.021561	-0.144390
PC8	-0.001614	0.010808

(b) cluster (PCs)

4. Under-Reporting of Hiring Discrimination in terms of Gender

```
# Setting up cross-validation and model training
# Logistic Regression with Lasso (L1) penalty
log_reg_param_grid = {'C': [0.01, 0.1, 1, 10, 100]} # smaller C denotes stronger regularization
log_reg_grid = GridSearchCV(LogisticRegression(penalty='l1', solver='liblinear', random_state=42),
                            log_reg_param_grid, cv=5, scoring='roc_auc')
log_reg_grid.fit(X_train_scaled, y_train)

# Best models and their parameters
best_log_reg_model = log_reg_grid.best_estimator_

# AUC scores for the best models
log_reg_auc = roc_auc_score(y_train, best_log_reg_model.predict_proba(X_train_scaled)[:, 1])
```

Hyper-parameter tuning to obtain best **logistic regression** model in terms of **AUC**

```
# Setting up cross-validation and model training
# Random Forest
rf_param_grid = {'n_estimators': [100, 200, 300], 'max_depth': [5, 10, 15]}
rf_grid = GridSearchCV(RandomForestClassifier(random_state=42), rf_param_grid, cv=5, scoring='roc_auc')
rf_grid.fit(X_train_scaled, y_train)

# Best models and their parameters
best_rf_model = rf_grid.best_estimator_

# AUC scores for the best models
rf_auc = roc_auc_score(y_train, best_rf_model.predict_proba(X_train_scaled)[:, 1])
```

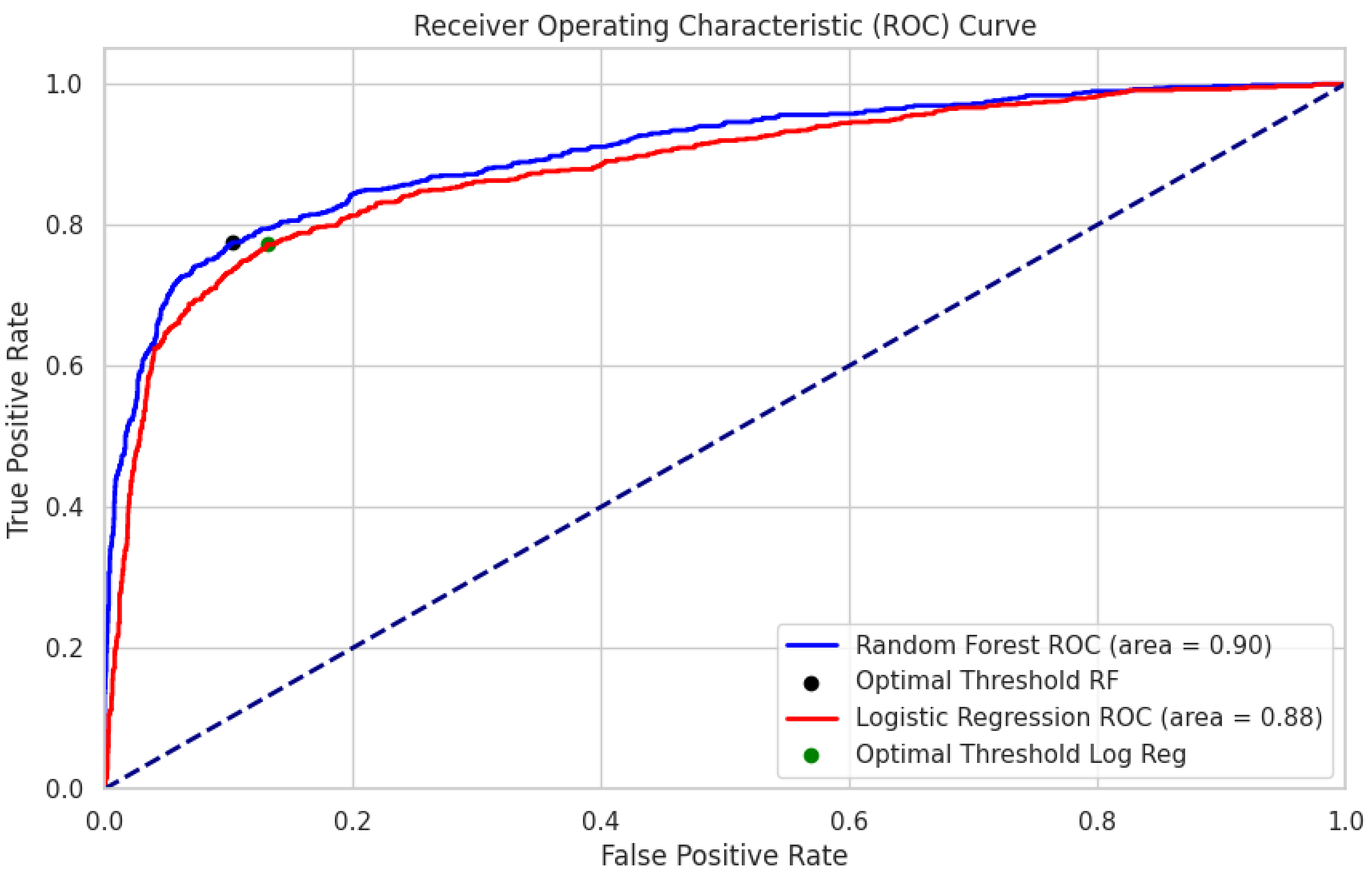
Hyper-parameter tuning to obtain best **random forest** model in terms of **AUC**

```
(rf_grid.best_params_, log_reg_grid.best_params_, rf_auc, log_reg_auc)
```

```
({'max_depth': 5, 'n_estimators': 200},
 {'C': 0.1},
 0.9011937903901778,
 0.8797141228748673)
```

```
# Compute predicted probabilities for both models
y_pred_probs_rf = best_rf_model.predict_proba(X_train_scaled)[:, 1]
y_pred_probs_log_reg = best_log_reg_model.predict_proba(X_train_scaled)[:, 1]

# Compute ROC curve for Random Forest
fpr_rf, tpr_rf, thresholds_rf = roc_curve(y_train, y_pred_probs_rf)

# Compute ROC curve for Logistic Regression
fpr_log_reg, tpr_log_reg, thresholds_log_reg = roc_curve(y_train, y_pred_probs_log_reg)

# Function to find the optimal threshold
def find_optimal_threshold(fpr, tpr, thresholds):
    sum_sensitivity_specificity = tpr + (1 - fpr)
    optimal_idx = np.argmax(sum_sensitivity_specificity)
    optimal_threshold = thresholds[optimal_idx]
    return optimal_threshold

# Optimal thresholds
optimal_threshold_rf = find_optimal_threshold(fpr_rf, tpr_rf, thresholds_rf)
optimal_threshold_log_reg = find_optimal_threshold(fpr_log_reg, tpr_log_reg, thresholds_log_reg)

# Use these thresholds for making final predictions on the training set
optimal_predictions_rf = np.where(y_pred_probs_rf >= optimal_threshold_rf, 1, 0)
optimal_predictions_log_reg = np.where(y_pred_probs_log_reg >= optimal_threshold_log_reg, 1, 0)

(optimal_threshold_rf, optimal_threshold_log_reg, optimal_predictions_rf, optimal_predictions_log_reg)
```

```
(0.21835354867789095,
 0.14743352461661669,
 array([1, 0, 0, ..., 0, 0, 0]),
 array([1, 0, 0, ..., 0, 0, 0]))
```


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Random Forest ROC (area = 0.90)
Optimal Threshold RF
Logistic Regression ROC (area = 0.88)
Optimal Threshold Log Reg

disc_hire	0.0	1.0
gender		
0	0.811994	0.188006
1	0.788824	0.211176

(a) Original Data

predicted_log_reg	0	1
gender		
0	0.531250	0.468750
1	0.121212	0.878788

(b) Logistic Regression

predicted_rf	0	1
gender		
0	0.515625	0.484375
1	0.090909	0.909091

(c) Random Forest

5. Link Between Hiring Discrimination Experiences and Self-Rated Health Score

```
from scipy.stats import chi2_contingency

# Creating a new column to categorize respondents into the four groups
data['group_category'] = data.apply(
    lambda x: 'No' if x['disc_hire'] == 0 else (
        'Yes' if x['disc_hire'] == 1 else (
            'Predicted No' if x['disc_hire_predicted'] == 0 else 'Predicted Yes'
        )
    ),
    axis=1
)

# Creating a contingency table for health across the four groups
contingency_table = pd.crosstab(data['group_category'], data['health'])

# Performing the chi-square test
chi2, p, dof, expected = chi2_contingency(contingency_table)
```

Categorize data into four groups according to the results obtained from random forest prediction model

(1) Answered No

(2) Answered Yes

(3) Answered NAN, but Predicted No

(4) Answered NAN, but Predicted Yes

5. Link Between Hiring Discrimination Experiences and Self-Rated Health Score

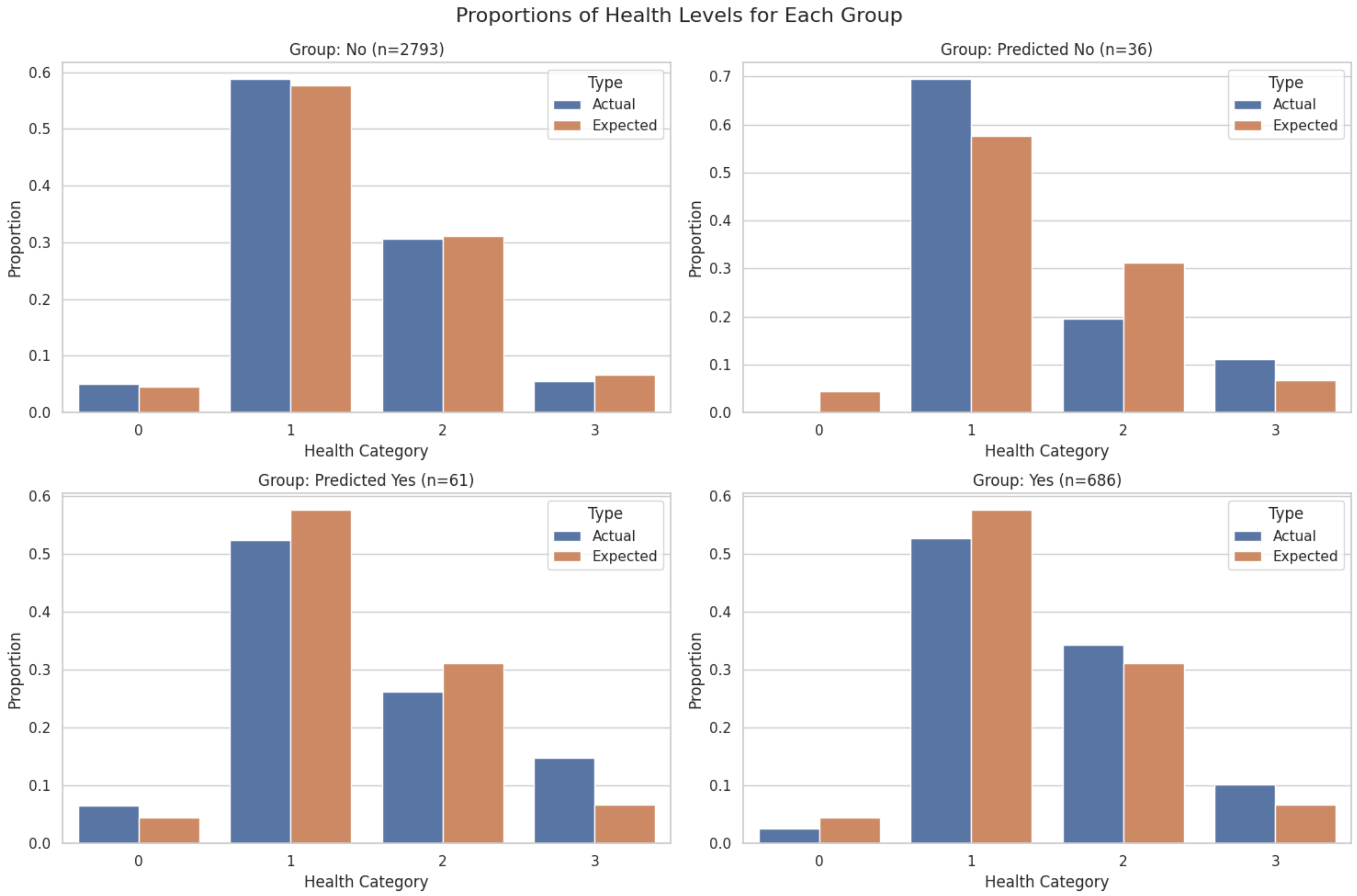
health	0	1	2	3
group_category				
No	139	1642	856	156
Predicted No	0	25	7	4
Predicted Yes	4	32	16	9
Yes	18	362	236	70

(a) Observed Frequency

health	0	1	2	3
group_category				
No	125.747483	1609.723993	870.859899	186.668624
Predicted No	1.620805	20.748322	11.224832	2.406040
Predicted Yes	2.746365	35.156879	19.019855	4.076902
Yes	30.885347	395.370805	213.895414	45.848434

(b) Expected Frequency

If there are no association between the experience of hiring discrimination and health,…


Proportions of Health Levels for Each Group
Group: No (n=2793)
Group: Predicted No (n=36)
Group: Predicted Yes (n=61)
Group: Yes (n=686)
Proportion
Health Category
0
1
2
3
Type
Actual
Expected

Conduct chi-square test (Distribution test)

Observed Frequency

Expected Frequency

$$\chi^2 = \sum_{k=1}^{n} \frac{(O_k - E_k)^2}{E_k}$$

Under null hypothesis, it follows chi-square distribution with df=(r-1)X(c-1)=9

$$TS = 42.953$$

$$p - value = 2.198e - 06$$

∴ Reject Null Hypothesis

Conduct Pairwise Comparison

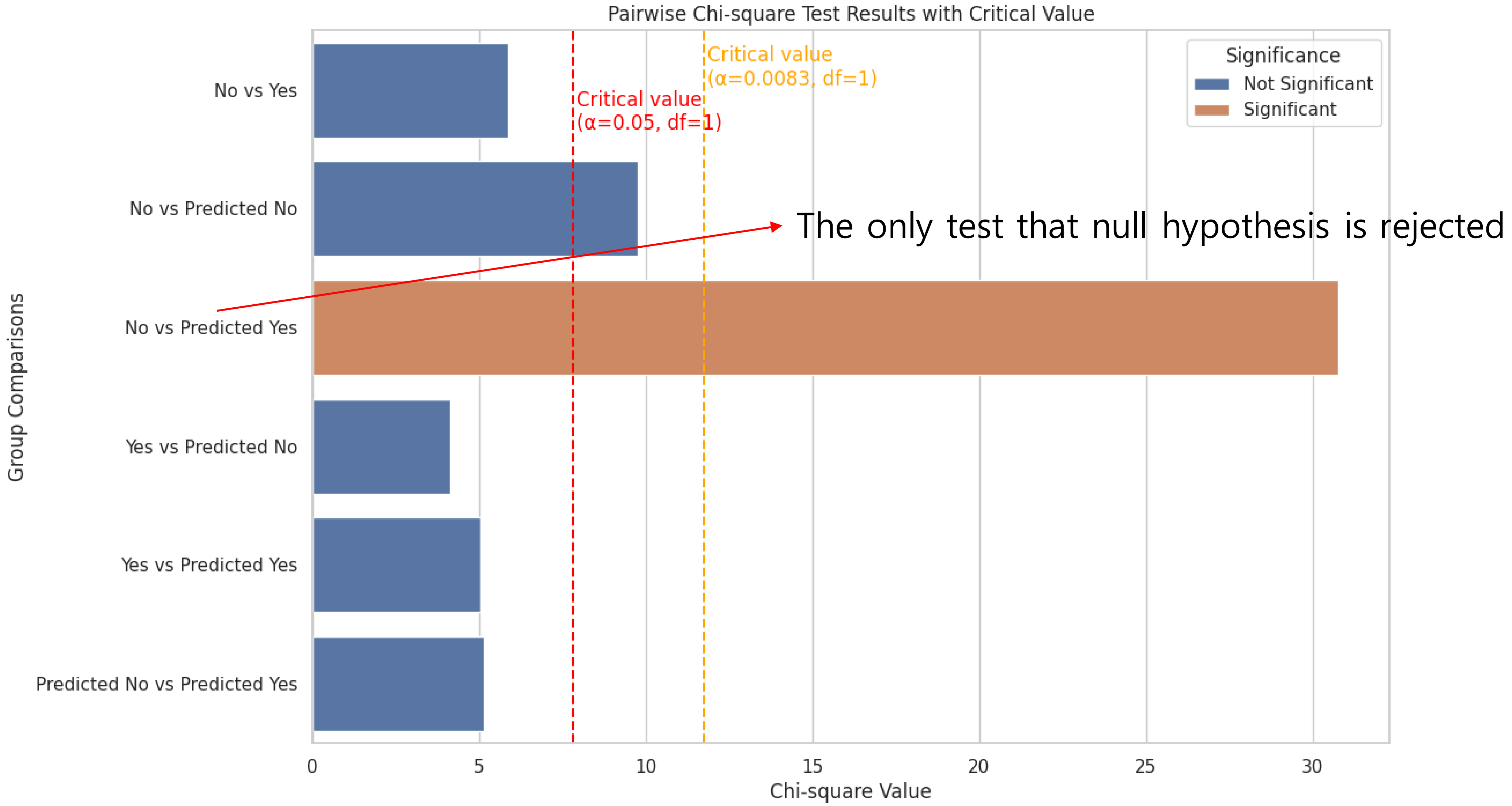
[55]

```
# Function to perform pairwise chi-square tests
def pairwise_chi2_test(table, groups, corrected_alpha):
    results = []
    for i in range(len(groups)):
        for j in range(i + 1, len(groups)):
            sub_table = table.iloc[[i, j]]
            chi2, p, _, _ = chi2_contingency(sub_table)
            result = {
                'Group 1': groups[i],
                'Group 2': groups[j],
                'Chi-square': chi2,
                'p-value': p,
                'Significant at alpha 0.0083': p < corrected_alpha
            }
            results.append(result)
    return results

# Pairwise comparisons
groups = ['No', 'Yes', 'Predicted No', 'Predicted Yes']
corrected_alpha = 0.05 / 6 # Bonferroni Correction
pairwise_results = pairwise_chi2_test(contingency_table, groups, corrected_alpha)
```

Total $\binom{4}{2} = 6$ tests

Perform chi-square test for 6 pairs

To control FWER, apply bonferroni criteria
→ Reject test with p-value under 0.05/6=0.083


Pairwise Chi-square Test Results with Critical Value
Critical value
(α=0.05, df=1)
Critical value
(α=0.0083, df=1)
Significance
Not Significant
Significant
No vs Yes
No vs Predicted No
No vs Predicted Yes
Yes vs Predicted No
Yes vs Predicted Yes
Predicted No vs Predicted Yes
Group Comparisons
0
5
10
15
20
25
30
Chi-square Value
The only test that null hypothesis is rejected

How about Benjamini-Hochberg Procedure?

	Comparison	Chi-square	p-value	Adjusted p-value	BH Significance
0	No vs Yes	5.857588	1.187482e-01	0.204385	Not Significant
1	No vs Predicted No	9.767299	2.065136e-02	0.061954	Not Significant
2	No vs Predicted Yes	30.779064	9.461453e-07	0.000006	Significant
3	Yes vs Predicted No	4.135896	2.471588e-01	0.247159	Not Significant
4	Yes vs Predicted Yes	5.020241	1.703210e-01	0.204385	Not Significant
5	Predicted No vs Predicted Yes	5.149757	1.611540e-01	0.204385	Not Significant

Same as before…

Conclusion

1. **Identifying and Analyzing Key Variables in Hiring Discrimination Experiences**

   : disc_wage, disc_social, income_quartile

2. **PCA of 12 Explanatory Variables Influencing Discrimination Experiences**

   : PC1 capture overall experience in discrimination, …

3. Comparative Analysis of Subgroup Clustering Based on Original Variables and Principal Components

   : similarity in characteristics between clusters from original variables and PCs has confirmed

4. Under-Reporting of Hiring Discrimination in terms of Gender

   : Maybe …, but should consider whether it might have been cherry-picked

5. Link Between Hiring Discrimination Experiences and Self-Rated Health Score

   : Maybe …, but should consider whether it might have been cherry-picked

Thank you!