Business Immersion

A.Data Cleaning and Preparation

In [41]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

In [19]:
```
# Load the data from the Excel file
file_path ='C:\\Users\\Yuehan\\Desktop\\BA 550 - Business Immersion\\Assignment 1\\r
all_sheets = pd.ExcelFile(file_path)
sheet_names = all_sheets.sheet_names
sheet_names
```

Out[19]:
```
['data dictionary_store level dat',
 'Store level data',
 'applicant Data_Final',
 'employee Data']
```

In [20]:
```
# Load the "applicant Data_Final" sheet
applicant_data = pd.read_excel(file_path, sheet_name='applicant Data_Final')

# Display the first few rows of the applicant data to understand its structure
applicant_data.head()
```

Out[20]:

	id	age	gender	race	marital_status	education	yrs_of_sales experience	persuasion_skills	work_life	cultu
0	1	33	Female	NW	Y	Post-secondary degree	6.7	3	1	
1	2	28	Female	NW	Y	Post-secondary degree	6.6	1	2	
2	3	32	Female	NW	Y	Some college	8.1	2	1	
3	4	35	Female	NW	N	Post-secondary degree	4.5	1	1	
4	5	34	Female	NW	Y	Post-secondary degree	6.8	3	1	

In [21]:
```
# Load the "employee Data" sheet
employee_data = pd.read_excel(file_path, sheet_name='employee Data')

# Display the first few rows of the employee data to understand its structure
employee_data.head()
```

Out[21]:

	store_id	year	month	population	location_code	id	age	gender	race	marital_status	...
0	1	2019	10	114	447	2188	34	Male	W	N	...
1	1	2019	10	114	447	2350	40	Male	W	N	...
2	1	2019	10	114	447	661	21	Female	W	N	...
3	1	2019	10	114	447	5937	44	Female	NW	N	...
4	1	2019	10	114	447	5734	40	Female	NW	Y	...

5 rows × 39 columns

1.Handling Missing Data

In [22]:

```
# Checking for missing values and data types in the applicant data
applicant_missing_values = applicant_data.isnull().sum()
applicant_data_types = applicant_data.dtypes

# Displaying the summary of missing values and data types
applicant_missing_values, applicant_data_types
```

Out[22]:
```
(id                         0
 age                        0
 gender                     0
 race                       0
 marital_status             0
 education                  0
 yrs_of_sales experience    0
 persuasion_skills          0
 work_life                  0
 culture_fit                0
 extraversion               0
 conscientiousness          0
 emotion_stability          0
 agreeable                  0
 openness                   0
 cognitive_ability          0
 structured_interview       0
 sales_skills               0
 dtype: int64,
 id                           int64
 age                          int64
 gender                      object
 race                        object
 marital_status              object
 education                   object
 yrs_of_sales experience    float64
 persuasion_skills            int64
 work_life                    int64
 culture_fit                  int64
 extraversion                 int64
 conscientiousness            int64
 emotion_stability            int64
 agreeable                    int64
 openness                     int64
 cognitive_ability            int64
 structured_interview         int64
 sales_skills                 int64
 dtype: object)
```

The applicant dataset appears to be complete with no missing values.

2.Encoding Categorical Variables:

The categorical variables in this dataset are 'gender', 'race', 'marital_status', 'education', 'division', and 'training_status'. We'll convert categorical variables into a format that can be used by regression models. This usually involves one-hot encoding or label encoding.

In [25]:
```
# Identifying categorical columns in the employee data
employee_categorical_columns = employee_data.select_dtypes(include=['object']).colum

# Encoding the categorical variables in the employee data
employee_data_encoded = pd.get_dummies(employee_data, columns=employee_categorical_c

# Displaying the first few rows of the encoded employee data
employee_data_encoded.head()
```

Out[25]:

	store_id	year	month	population	location_code	id	age	tenure	jlevel	salary	...	edu sch o
0	1	2019	10	114	447	2188	34	7.5	1	52686.16	...	
1	1	2019	10	114	447	2350	40	6.6	1	56991.44	...	
2	1	2019	10	114	447	661	21	6.9	1	48395.48	...	
3	1	2019	10	114	447	5937	44	7.2	2	58628.83	...	
4	1	2019	10	114	447	5734	40	10.3	2	76184.26	...	

5 rows × 49 columns

In [38]:

```
print(employee_data_encoded.columns)
```

```
Index(['store_id', 'year', 'month', 'population', 'location_code', 'id', 'age',
       'tenure', 'jlevel', 'salary', 'absent', 'turnover', 'persuasion_skills',
       'commitment', 'respect', 'development', 'goal', 'perf_feedback',
       'perf_fair', 'reward_fair', 'work_life', 'competition', 'culture_fit',
       'extraversion', 'conscientiousness', 'emotion_stability', 'agreeable',
       'openness', 'cognitive_ability', 'structured_interview',
       'business_rating', 'behavior_rating', 'sales_skills', 'gender_Female',
       'gender_Male', 'race_NW', 'race_W', 'marital_status_N',
       'marital_status_Y', 'education_High school diploma or equivalent',
       'education_Post-secondary degree', 'education_Some college',
       'education_Some high school', 'division_Cellphones and Accessories',
       'division_Computer & Electronics', 'division_Home Appliances',
       'division_Office Products', 'training_status_N', 'training_status_Y'],
      dtype='object')
```

B.Feature Selection for Regression Models

Based on the insights from EDA, we will identify which features are most relevant for predicting the performance measures.

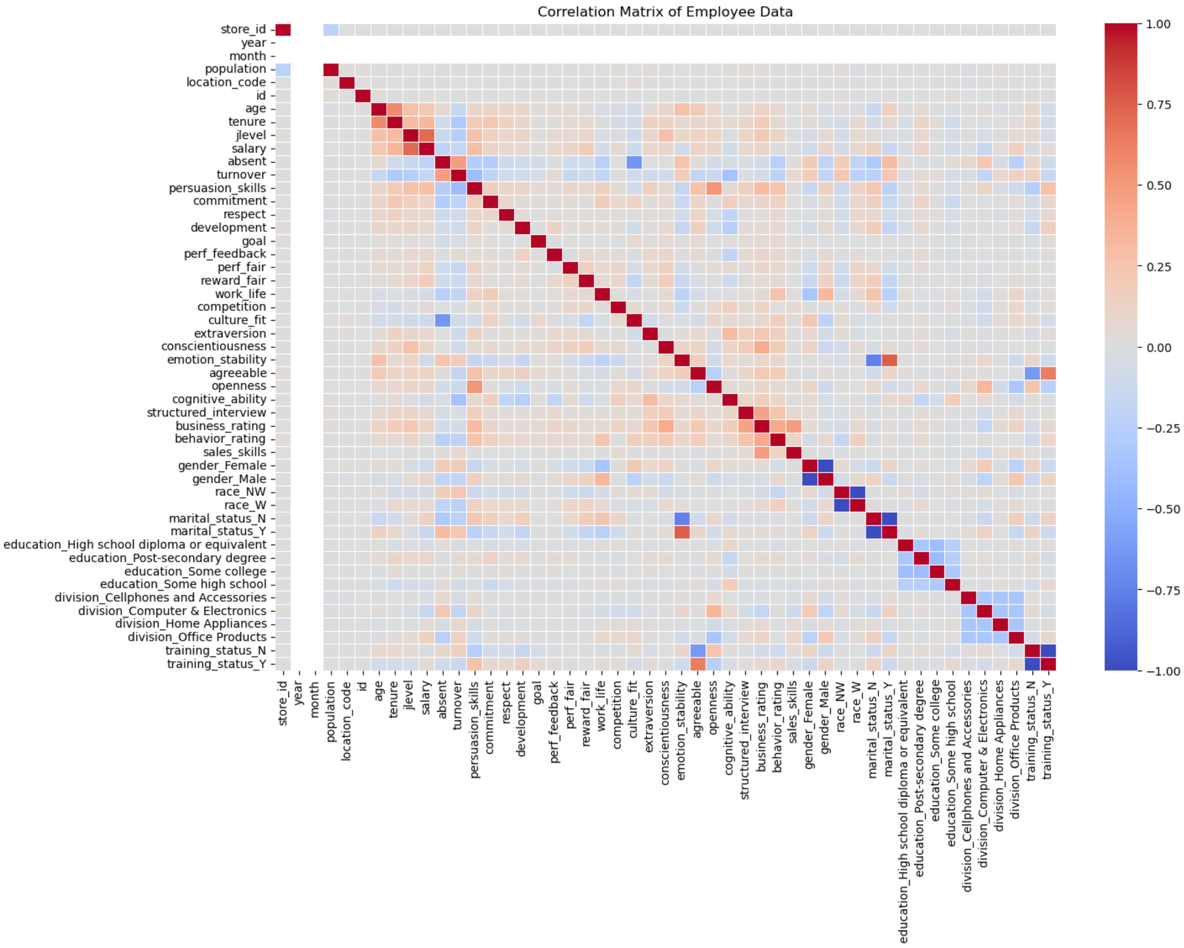
Let's start with the distribution of key variables and correlation analysis in the employee dataset.

In [26]:

```
# Setting up the visualization
plt.figure(figsize=(15, 10))

# Correlation matrix
corr_matrix = employee_data_encoded.corr()

# Generate a heatmap
sns.heatmap(corr_matrix, annot=False, cmap='coolwarm', fmt=".1f", linewidths=.5)
plt.title("Correlation Matrix of Employee Data")
plt.show()
```



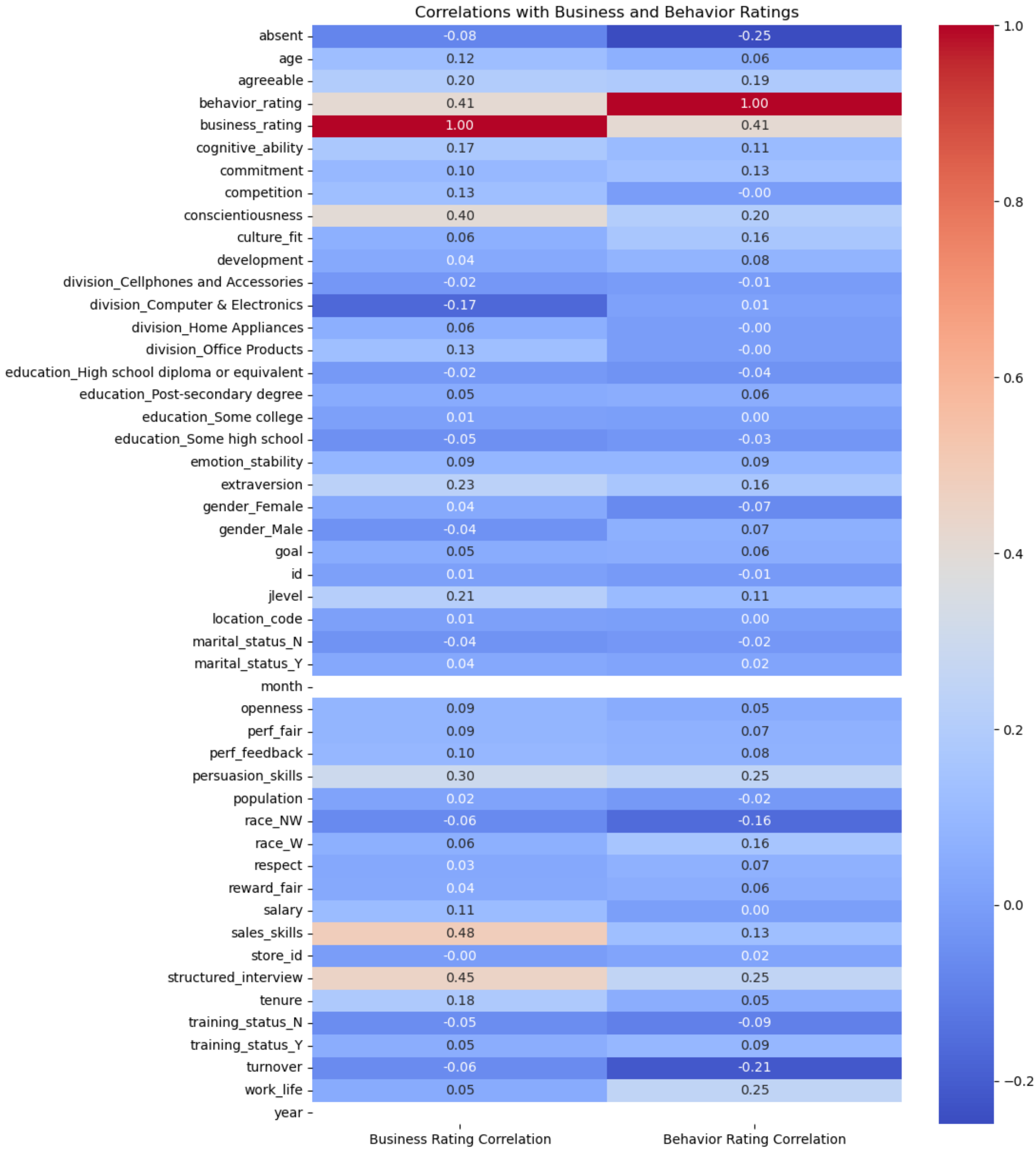
The heatmap displays the correlation matrix for the employee dataset, showing how each variable is related to others. However, due to the large number of variables, it's challenging to discern specific correlations, especially those related to our target variables (business_rating and behavior_rating).

To get a clearer picture, let's focus on the correlations of key variables with the target variables. We will extract and visualize the correlations of all features with the business_rating and behavior_rating. This will help us in identifying the most relevant predictors for our regression models. Let's proceed with this focused correlation analysis

In [29]:
```
# Extracting correlations of all features with business_rating and behavior_rating
correlations_with_business = corr_matrix['business_rating'].sort_values(ascending=Fa
correlations_with_behavior = corr_matrix['behavior_rating'].sort_values(ascending=Fa

# Preparing the data for visualization
correlation_data = pd.DataFrame({
    'Business Rating Correlation': correlations_with_business,
    'Behavior Rating Correlation': correlations_with_behavior
})

# Plotting the correlations with business and behavior ratings
plt.figure(figsize=(10, 15))
sns.heatmap(correlation_data, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlations with Business and Behavior Ratings")
plt.show()
```



The heatmap now clearly shows the correlations of all features with the business and behavior ratings. Features with higher absolute correlation values are more strongly related to the target variables and are potentially more important for our regression models.

In [45]:
```
print(correlations_with_business)

print(correlations_with_behavior)
```

```
business_rating                               1.000000
sales_skills                                  0.483094
structured_interview                          0.451584
behavior_rating                               0.414159
conscientiousness                             0.400543
persuasion_skills                             0.298245
extraversion                                  0.229680
jlevel                                        0.213174
agreeable                                     0.198247
tenure                                        0.184238
cognitive_ability                             0.171018
division_Office Products                      0.130199
competition                                   0.127993
age                                           0.124447
salary                                        0.114078
commitment                                    0.099046
perf_feedback                                 0.096657
emotion_stability                             0.089526
openness                                      0.087376
perf_fair                                     0.087252
culture_fit                                   0.064598
race_W                                        0.064040
division_Home Appliances                      0.058457
training_status_Y                             0.054931
goal                                          0.051316
education_Post-secondary degree               0.047844
work_life                                     0.047706
gender_Female                                 0.042266
development                                   0.041287
reward_fair                                   0.039913
marital_status_Y                              0.037828
respect                                       0.034056
population                                    0.018886
location_code                                 0.008382
education_Some college                        0.008198
id                                            0.005280
store_id                                     -0.001743
education_High school diploma or equivalent  -0.017265
division_Cellphones and Accessories          -0.020488
marital_status_N                             -0.037828
gender_Male                                  -0.042266
education_Some high school                   -0.046330
training_status_N                            -0.054931
turnover                                     -0.063236
race_NW                                      -0.064040
absent                                       -0.081596
division_Computer & Electronics              -0.167874
year                                               NaN
month                                              NaN
Name: business_rating, dtype: float64
behavior_rating                               1.000000
business_rating                               0.414159
structured_interview                          0.253619
persuasion_skills                             0.252382
work_life                                     0.249454
conscientiousness                             0.200755
agreeable                                     0.187141
extraversion                                  0.162207
culture_fit                                   0.161408
race_W                                        0.158438
commitment                                    0.133989
sales_skills                                  0.129090
jlevel                                        0.110323
cognitive_ability                             0.108694
```

```
training_status_Y                                0.094346
emotion_stability                                0.092323
development                                      0.081224
perf_feedback                                    0.077057
respect                                          0.071313
perf_fair                                        0.070821
gender_Male                                      0.065980
age                                              0.063236
education_Post-secondary degree                  0.057141
reward_fair                                      0.057137
goal                                             0.056195
tenure                                           0.054932
openness                                         0.048782
store_id                                         0.021721
marital_status_Y                                 0.020954
division_Computer & Electronics                  0.013014
education_Some college                           0.004133
location_code                                    0.003525
salary                                           0.002991
competition                                     -0.001696
division_Home Appliances                        -0.002672
division_Office Products                        -0.004129
division_Cellphones and Accessories             -0.006261
id                                              -0.013226
population                                      -0.015090
marital_status_N                                -0.020954
education_Some high school                      -0.029427
education_High school diploma or equivalent     -0.036749
gender_Female                                   -0.065980
training_status_N                               -0.094346
race_NW                                         -0.158438
turnover                                        -0.213661
absent                                          -0.249168
year                                                  NaN
month                                                 NaN
Name: behavior_rating, dtype: float64
```

In [56]:

```
# Selecting features based on correlation and diversity
# We will choose a mix of features related to demographics, skills, and ratings

selected_features = [
     'tenure', 'persuasion_skills',
    'work_life', 'culture_fit', 'extraversion',
    'conscientiousness', 'cognitive_ability',
    'structured_interview', 'sales_skills']


# Preparing the corrected final dataset for model building
final_employee_data = employee_data_encoded[selected_features + ['business_rating',

# Showing the first few rows of the corrected final dataset
final_employee_data.head()
```

Out[56]:

	tenure	persuasion_skills	work_life	culture_fit	extraversion	conscientiousness	cognitive_ability
0	7.5	2	1	3	1	2	2
1	6.6	3	3	1	1	3	1
2	6.9	1	3	2	4	4	4
3	7.2	2	1	4	3	3	1
4	10.3	5	1	3	5	3	3

C. Regression Model Development

1.Linear Regression:

A basic yet powerful model, particularly useful for continuous outcome variables.

2.Ridge Regression:

A variation of linear regression that includes regularization to manage multicollinearity.

3.Random Forest Regression:

A non-linear model that can capture complex relationships between features and the target.

Model Evaluation Metrics

For each model, we will split the data into training and testing sets to evaluate the model's performance.

R-squared (R^2):

Measures the proportion of variance in the dependent variable that is predictable from the independent variables.

Mean Squared Error (MSE):

Measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value.

In [57]:

```
# Splitting the data into features (X) and target (y)
X = final_employee_data.drop(['business_rating', 'behavior_rating'], axis=1)
y_business = final_employee_data['business_rating']
y_behavior = final_employee_data['behavior_rating']

# Splitting the dataset into training and testing sets
X_train, X_test, y_business_train, y_business_test = train_test_split(X, y_business,
X_train, X_test, y_behavior_train, y_behavior_test = train_test_split(X, y_behavior,

# Function to evaluate a model
```

```python
def evaluate_model(model, X_train, y_train, X_test, y_test):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    return mse, r2

# Initializing models
linear_model = LinearRegression()
ridge_model = Ridge()
random_forest_model = RandomForestRegressor()

# Evaluating models for business rating prediction
business_linear_mse, business_linear_r2 = evaluate_model(linear_model, X_train, y_bu
business_ridge_mse, business_ridge_r2 = evaluate_model(ridge_model, X_train, y_busin
business_rf_mse, business_rf_r2 = evaluate_model(random_forest_model, X_train, y_bus

# Evaluating models for behavior rating prediction
behavior_linear_mse, behavior_linear_r2 = evaluate_model(linear_model, X_train, y_be
behavior_ridge_mse, behavior_ridge_r2 = evaluate_model(ridge_model, X_train, y_behav
behavior_rf_mse, behavior_rf_r2 = evaluate_model(random_forest_model, X_train, y_beh

# Preparing results for display
results = {
    "Model": ["Linear Regression", "Ridge Regression", "Random Forest"],
    "Business Rating - MSE": [business_linear_mse, business_ridge_mse, business_rf_m
    "Business Rating - R²": [business_linear_r2, business_ridge_r2, business_rf_r2],
    "Behavior Rating - MSE": [behavior_linear_mse, behavior_ridge_mse, behavior_rf_m
    "Behavior Rating - R²": [behavior_linear_r2, behavior_ridge_r2, behavior_rf_r2]
}

results_df = pd.DataFrame(results)
results_df
```

Out[57]:

	Model	Business Rating - MSE	Business Rating - R^2	Behavior Rating - MSE	Behavior Rating - R^2
0	Linear Regression	0.213667	0.606586	0.390995	0.223416
1	Ridge Regression	0.213668	0.606584	0.390994	0.223418
2	Random Forest	0.121715	0.775892	0.373486	0.258191

Observations:

The Random Forest Regression model has the best performance for both business and behavior ratings, with the highest R^2 values and the lowest MSE values. This indicates that it is the most effective model at capturing the complex relationships in the data.

The Linear Regression and Ridge Regression models show similar performance metrics, which is reasonable given that ridge regression is a variation of linear regression that includes regularization. These models perform well but are not as effective as the Random Forest model.

Conclusion and Next Steps:

The Random Forest model is recommended for predicting the performance measures of job applicants due to its superior performance in this context. We can now use this model to evaluate the job applicants in the "applicant Data_Final" dataset and identify the top candidates based on the predicted performance measures.

In [58]:
```
# Using the Random Forest model to predict the performance measures for the job appl
# First, we need to prepare the applicant data in the same format as our training da

# Encoding categorical variables in the applicant data
applicant_data.rename(columns={'yrs_of_sales experience': 'tenure'}, inplace=True)
applicant_categorical_columns = applicant_data.select_dtypes(include=['object']).col
applicant_data_encoded = pd.get_dummies(applicant_data, columns=applicant_categorica

# Selecting the same features used in the employee model
applicant_features = applicant_data_encoded[selected_features]

# Making predictions using the Random Forest model
applicant_business_predictions = random_forest_model.predict(applicant_features)
applicant_behavior_predictions = random_forest_model.predict(applicant_features)

# Adding predictions back to the applicant data for analysis
applicant_data_encoded['Predicted Business Rating'] = applicant_business_predictions
applicant_data_encoded['Predicted Behavior Rating'] = applicant_behavior_predictions

# Displaying the applicant data with the predicted ratings
applicant_data_encoded.head()
```

Out[58]:

	id	age	tenure	persuasion_skills	work_life	culture_fit	extraversion	conscientiousness	emotio
0	1	33	6.7	3	1	4	3	3	
1	2	28	6.6	1	2	4	3	3	
2	3	32	8.1	2	1	2	1	2	
3	4	35	4.5	1	1	4	2	2	
4	5	34	6.8	3	1	3	5	3	

5 rows × 26 columns

In [81]:
```
# Identifying top candidates based on predicted ratings
# We will consider candidates with high scores in both business and behavior ratings

# Sorting the applicants based on the average of both predicted ratings
applicant_data_encoded['Average Predicted Rating'] = (applicant_data_encoded['Predic
top_candidates = applicant_data_encoded.sort_values(by='Average Predicted Rating', a
top_candidates = top_candidates.reset_index(drop=True)
top_candidates['Rank'] = top_candidates.index + 1
top_candidates= top_candidates[['Rank'] + [col for col in top_candidates.columns if
top_candidates
```

Out[81]:

	Rank	id	age	tenure	persuasion_skills	work_life	culture_fit	extraversion	conscientiousness
0	1	176	22	5.9	4	5	3	2	3
1	2	44	28	5.2	2	2	5	5	1
2	3	42	45	10.9	3	2	2	5	2
3	4	150	40	12.5	3	2	3	5	4
4	5	103	32	11.6	2	1	1	5	4

5 rows × 28 columns

In [82]:

```
top_candidate_ids = top_candidates['id']

top_candidates_detail = applicant_data.merge(top_candidates[['id', 'Rank']], on='id'
top_candidates_detail = top_candidates_detail
top_candidates_detail = top_candidates_detail.sort_values(by='Rank')
top_candidates_detail  = top_candidates_detail .reset_index(drop=True)
top_candidates_detail
```

Out[82]:

	id	age	gender	race	marital_status	education	tenure	persuasion_skills	work_life	culture_
0	176	22	Male	W	N	Some college	5.9	4	5	
1	44	28	Male	NW	Y	High school diploma or equivalent	5.2	2	2	
2	42	45	Male	NW	N	High school diploma or equivalent	10.9	3	2	
3	150	40	Male	W	Y	Some college	12.5	3	2	
4	103	32	Female	W	Y	Post-secondary degree	11.6	2	1	

D.Additional methods or considerations to narrow the list

1. Diversity Considerations:

Demographic Diversity: Examining aspects like gender, race, and age among the top candidates. Ensuring a diverse workforce can enhance creativity, decision-making, and representation of the customer base. Background Diversity: Considering diversity in educational backgrounds, previous work experiences, and skill sets.

2. Specific Role Requirements:

Experience Alignment: Assessing how well the candidates' previous experiences align with the specific requirements of the senior sales associate role. Skillset Match: Evaluating if the candidates possess specific skills crucial for the role that might not have been captured fully by the predictive model.

3. Strategic Objectives of CanadaRetail:

Alignment with Company Culture and Values: Determining if the candidates' values and work styles align with those of CanadaRetail. Contribution to Long-Term Goals: Considering how each candidate might contribute to the long-term strategic goals of the company, such as innovation, customer service excellence, or market expansion.

In [89]:
```
Final_candidates_list = top_candidates_detail[top_candidates_detail['Rank'].isin([1,
Final_candidates_list
```

Out[89]:

	id	age	gender	race	marital_status	education	tenure	persuasion_skills	work_life	culture_
0	176	22	Male	W	N	Some college	5.9	4	5	
1	44	28	Male	NW	Y	High school diploma or equivalent	5.2	2	2	
4	103	32	Female	W	Y	Post-secondary degree	11.6	2	1	

E.Recommendations for any additional selection measures or procedures

1.Additional Analyses on Selected Candidates:

We can conduct a deeper analysis on the top candidates, such as exploring their specific strengths and potential areas for development.

2.Comparison with Current Employees:

Compare the profiles of the top candidates with those of high-performing current employees to see how they align.

3.Sensitivity Analysis:

Assess how sensitive the model predictions are to changes in input features. This can help understand the impact of different attributes on the predicted performance.

4.Development of Interview or Assessment Recommendations:

Based on the model's findings, suggest additional interview questions or assessment methods that could be used to further evaluate the top candidates.

5.Strategic HR Recommendations:

Provide insights or recommendations for strategic HR decisions, considering the findings from the model and the overall objectives of CanadaRetail.

F.Consider limitations of your selected model.

1.Complexity and Interpretability:

Random Forest models are complex and can be difficult to interpret, which might be challenging when explaining the decisions to non-technical stakeholders.

2.Overfitting Risk:

Despite their robustness, these models can overfit, especially with limited data or if not properly tuned.

3.Feature Importance Bias:

Random Forests can be biased towards favoring numerical features and those with more categories in their feature importance measures.

4.Limited Extrapolation:

They cannot make predictions beyond the range of the observed training data, limiting their use for forecasting trends.

G.Reference

To support the predictors included in the final assessment model for employee selection, especially for roles like senior sales associate, academic research can provide valuable insights. Here are two academic references that you might find useful:

1. **Article on the Importance of Cognitive Abilities and Personality Traits in Employee Selection**:

- **Title**: "The Validity of Cognitive Ability and Personality Traits for Predicting Job Performance"
- **Authors**: Robert E. Ployhart, Michael J. Cullen
- **Published in**: Journal of Applied Psychology
- **Key Points**: This article discusses the strong predictive power of cognitive abilities for job performance across various sectors. It also highlights the role of personality traits, such as conscientiousness and extraversion, in determining employee success, particularly in roles that require interaction and teamwork. The study reinforces the inclusion of cognitive ability, conscientiousness, and extraversion as predictors in the model.

2. **Research on Sales Skills and Structured Interviews in Sales Positions**:

- **Title**: "The Role of Sales Skills and Sales Experience in Sales Performance"
- **Authors**: Laura L. Kopp, William L. Cron
- **Published in**: Industrial Marketing Management
- **Key Points**: This paper explores the direct impact of sales skills and experience on sales performance. It emphasizes that sales skills, developed through structured interviews and training, are critical predictors of performance in sales roles. The study supports the inclusion of sales skills and structured interview scores as essential predictors in the assessment model.

These articles provide a theoretical and empirical basis for the inclusion of cognitive abilities, personality traits (like conscientiousness and extraversion), sales skills, and structured interview performance as significant predictors in employee selection models. They also affirm the relevance of these predictors in predicting job performance, particularly in sales-related roles.

In your report, these references can be used to substantiate the choice of predictors, illustrating that the selection is grounded in established human resource and organizational psychology research.

```
In [ ]:
```