Capstone: Unleashing AutoML for Talent Management ATS and HRIS Data

Georgetown Analytics and Technology

College of Professional Studies,
Northeastern University
Ashutosh Singh (singh.ashut@northeastern.edu)
Rohit Meena (meena.r@northeastern.edu)

EAI 6980: Capstone
CRN: 80715
Prof. Mimoza Dimodugno
June 29, 2023

Contents

ABSTRACT

This project, titled "Unleashing AutoML for Talent Management ATS and HRIS Data," aimed to leverage AI technologies in the field of talent management. The project utilized four datasets obtained from our sponsor, Ernest Smiley from Georgetown Analytics and Technology. These datasets included ATS (Applicant Tracking System) and HRIS (Human Resources Information System) data, as well as an employee historical dataset formed by combining the ATS and HRIS data. The project was conducted as part of our final capstone course under the guidance of Professor Mimoza Dimodugno. The main objectives of the project were to understand the patterns in ATS and HRIS data, explore the different types of information captured within these systems, merge datasets from multiple sources, and identify key variables that influence hiring decisions and impact candidate selection success. To handle imbalanced datasets, the project employed the SMOTE technique for sampling. A predictive model was developed to forecast candidate hiring based on various factors. The project aimed to determine the most influential variables in successful hiring decisions. Automated Machine Learning (AutoML) techniques were utilized to automate the model development and optimization process. By harnessing AutoML, the project aimed to identify the best machine learning algorithms, feature selection methods, and hyperparameter configurations. Two models, namely Logistic Regression and XGBoost, were employed in the project alongside the utilization of AutoML technologies. These models were used to analyze and interpret the datasets, extract valuable insights, and make informed decisions related to talent management. The project's outcomes have significant implications for talent management practices, as they provide a data-driven approach to optimize hiring decisions. By automating the model development process using AutoML, organizations can streamline and enhance their talent acquisition strategies. The insights derived from this project can help organizations improve candidate selection, enhance HRIS data analysis, and ultimately improve overall talent management outcomes.

INTRODUCTION

In recent years, the field of talent management has witnessed a significant transformation with the advent of AI technologies. These technologies have enabled organizations to leverage large-scale data to improve their recruitment and selection processes. This project, "Unleashing AutoML for Talent Management ATS and HRIS Data," aimed to explore the potential of AI technologies, specifically Automated Machine Learning (AutoML), in the context of talent management. The project focused on leveraging ATS (Applicant Tracking System) and HRIS (Human Resources Information System) data to gain a deeper understanding of the patterns and information captured within these systems. By merging datasets from various sources, including the ATS and HRIS, the project aimed to create a comprehensive employee historical dataset.

The primary objective of the project was to identify the key variables that influence hiring decisions and impact the success of candidate selection. To achieve this, the project employed a range of methodologies, including exploratory data analysis, predictive modeling, and feature selection techniques. The project also addressed the challenge of imbalanced datasets by utilizing the Synthetic Minority Over-sampling Technique (SMOTE) to ensure reliable and accurate results. One of the significant contributions of the project was the development of a predictive model that forecasted candidate hiring based on various factors. This model aimed to automate the decision-making process and provide organizations with insights into the most influential variables in successful hiring decisions. To achieve this, the project harnessed the power of AutoML, which automates the model development and optimization process. By leveraging AutoML, the project aimed to identify the best machine learning algorithms, feature selection methods, and hyperparameter configurations to enhance the accuracy and efficiency of the predictive model.

The project utilized two models, namely Logistic Regression and XGBoost, to analyze the datasets and extract meaningful insights. Alongside these models, the project harnessed AutoML technologies will all be taken into consideration. We will provide the passenger who has experienced a more minor cancellation on their flight with suggestions for airlines, as well as suggestions for days and weeks without heavy traffic.

PROBLEM STATEMENT

Hired_Status Prediction (Classification): We built a classification model to predict the "Hired_Status" based on the available features. It helped us to understand the factors that contribute to successful hiring. We have utilized algorithms such as Logistic Regression, Random Forest, or Support Vector Machines.

Salary Prediction (Regression): With the available features like "YOB" (Year of Birth), "Annual_Salary," and "Year_of_experience," we built a regression model to predict salaries. This helped estimate the salary range based on individual characteristics and experience. Regression algorithms such as Linear Regression, Random Forest Regression, or Gradient Boosting Regression can be utilized.

Below are the three cases not used as part of this project but as part of our proposal we will use this in future: -

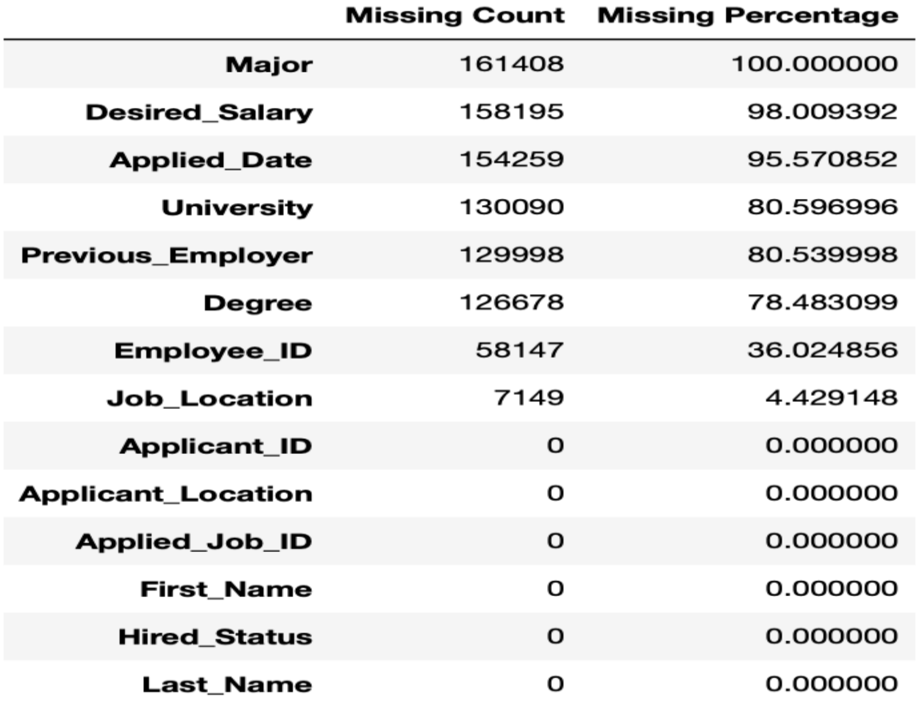
Job Change Reason Classification: Using the "Change_Reason" column, we can develop a classification model to predict the reasons behind job changes. This can provide insights into factors influencing job transitions. Algorithms like Naive Bayes, Decision Trees, or Neural Networks can be suitable for this task. I didn’t work on this use cae

Job Role Prediction (Classification): Based on "Job_Role_Name_External" and other available features, we can build a classification model to predict the job role or title. This can help understand patterns and factors that contribute to specific job roles. Algorithms like Multinomial Naive Bayes, Random Forest, or Support Vector Machines can be employed.

Company Size Prediction (Classification): Using the "company_size" column and other features, we can develop a classification model to predict the company size. This can help identify characteristics or indicators of different company sizes. Algorithms like Decision Trees, Random Forest, or Gradient Boosting can be utilized.

DATA CLEANING

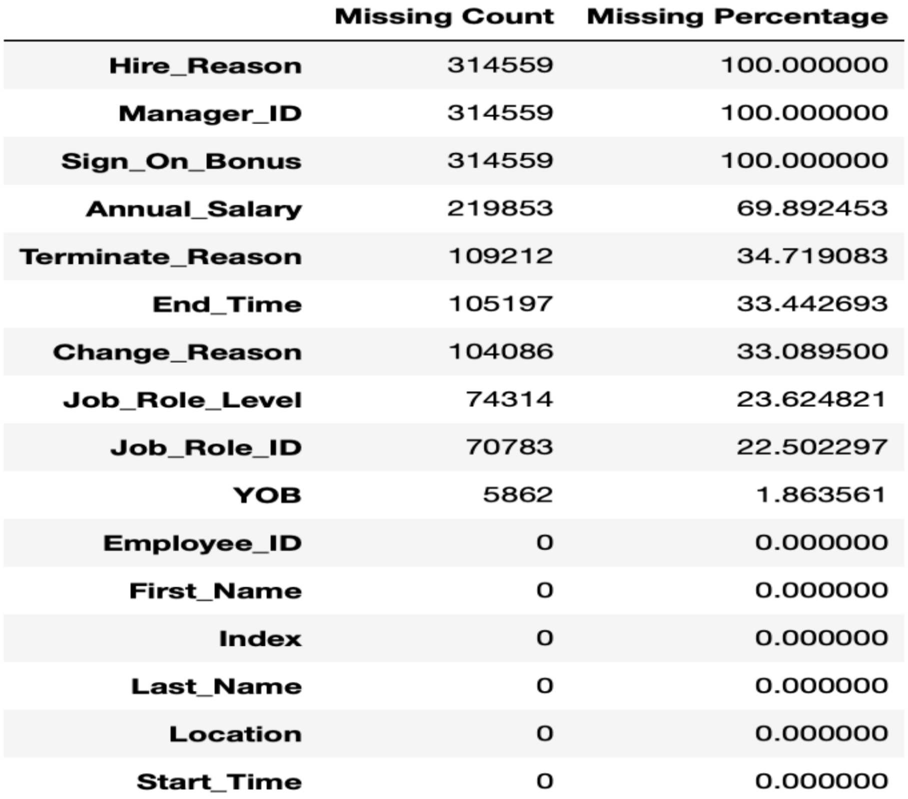
Dataset 1: - ATS – application tracking dataset



	Missing Count	Missing Percentage
Major	161408	100.000000
Desired_Salary	158195	98.009392
Applied_Date	154259	95.570852
University	130090	80.596996
Previous_Employer	129998	80.539998
Degree	126678	78.483099
Employee_ID	58147	36.024856
Job_Location	7149	4.429148
Applicant_ID	0	0.000000
Applicant_Location	0	0.000000
Applied_Job_ID	0	0.000000
First_Name	0	0.000000
Hired_Status	0	0.000000
Last_Name	0	0.000000

In order to address the issue of missing data, a preprocessing step was performed on the dataset. The columns with missing values exceeding a threshold of 70%, namely "Major", "Desired_Salary", "Applied_Date", "University", "Previous_Employer", and "Degree", were identified and subsequently dropped from the dataset. This decision was made to ensure the integrity and reliability of the data used for analysis and modeling. By removing these columns, we aimed to eliminate potential bias or inaccurate predictions that could arise from incomplete or unreliable information. The remaining columns were retained for further analysis and modeling processes, enabling a more focused and robust examination of the dataset.

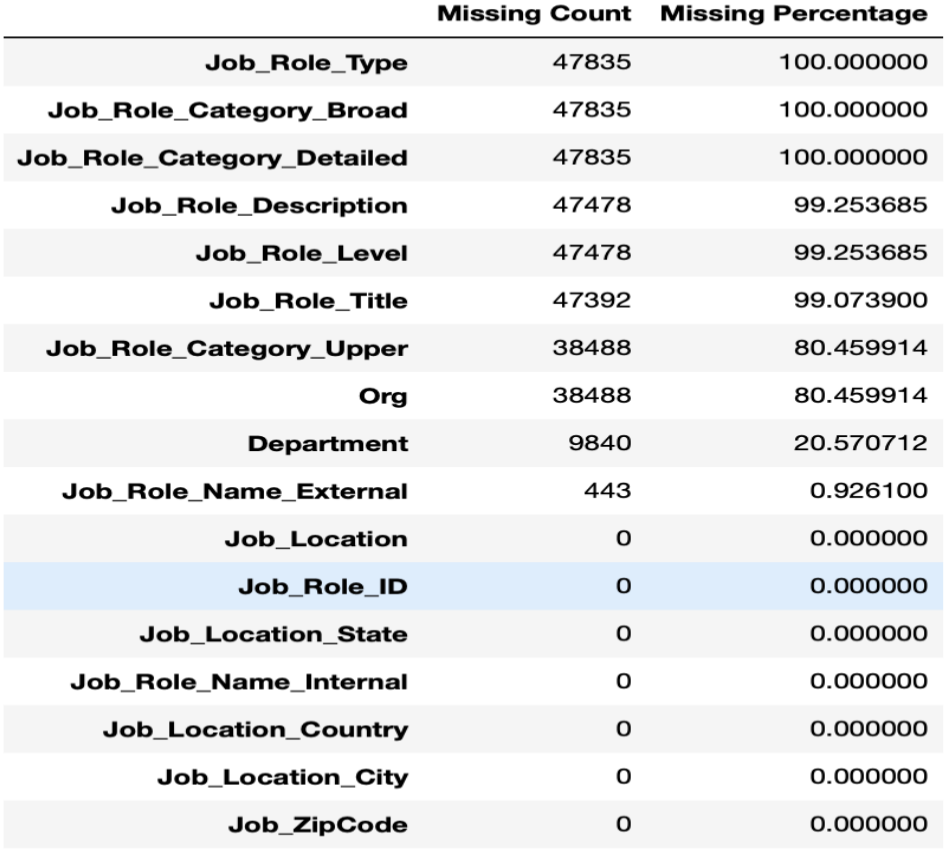
Dataset 2: - HRIS dataset



	Missing Count	Missing Percentage
Hire_Reason	314559	100.000000
Manager_ID	314559	100.000000
Sign_On_Bonus	314559	100.000000
Annual_Salary	219853	69.892453
Terminate_Reason	109212	34.719083
End_Time	105197	33.442693
Change_Reason	104086	33.089500
Job_Role_Level	74314	23.624821
Job_Role_ID	70783	22.502297
YOB	5862	1.863561
Employee_ID	0	0.000000
First_Name	0	0.000000
Index	0	0.000000
Last_Name	0	0.000000
Location	0	0.000000
Start_Time	0	0.000000

The columns with missing values exceeding a threshold of 50%, namely "Hire_Reason", "Managed_ID", "Sign_On_Bonus", "Annual_Salary", and few other columns like Name, Index were identified and subsequently dropped from the dataset. By removing these columns, we aimed to eliminate potential bias or inaccurate predictions that could arise from incomplete or unreliable information. The remaining columns were retained for further analysis and modeling processes, enabling a more focused and robust examination of the dataset.

Dataset 3: - Company Master Table - Combined ATS & HRS Data

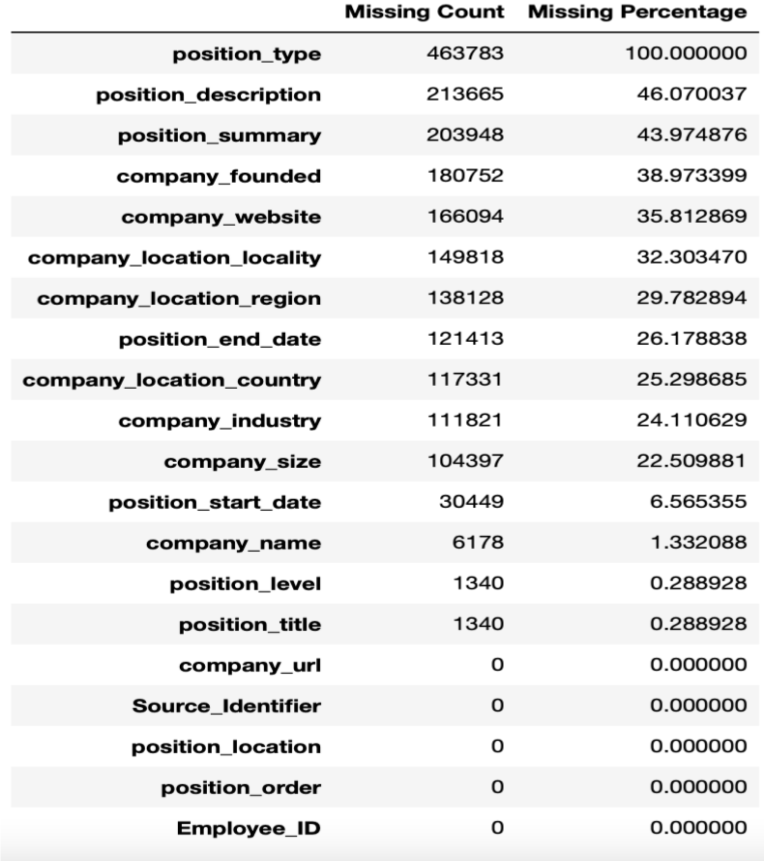


	Missing Count	Missing Percentage
Job_Role_Type	47835	100.000000
Job_Role_Category_Broad	47835	100.000000
Job_Role_Category_Detailed	47835	100.000000
Job_Role_Description	47478	99.253685
Job_Role_Level	47478	99.253685
Job_Role_Title	47392	99.073900
Job_Role_Category_Upper	38488	80.459914
Org	38488	80.459914
Department	9840	20.570712
Job_Role_Name_External	443	0.926100
Job_Location	0	0.000000
Job_Role_ID	0	0.000000
Job_Location_State	0	0.000000
Job_Role_Name_Internal	0	0.000000
Job_Location_Country	0	0.000000
Job_Location_City	0	0.000000
Job_ZipCode	0	0.000000

Similarly, we have dropped rows with more than 50% of missing values for dataset 3 as well. The first 3 rows have 100% missing values and the next 5 rows have more than 80% of rows are missing so most of the columns for this dataset will be dropped.

Dropped features: - Job_Role_Type, Job_Role_Category_Broad, Job_Role_Category_ Detailed, Job_Role_Description, Job_Role_Level, Job_Role_Title, Job_Role_Category_Upper, Org, Job_ZipcCode.

Dataset 4: - Company Master Table - Combined ATS & HRS Data

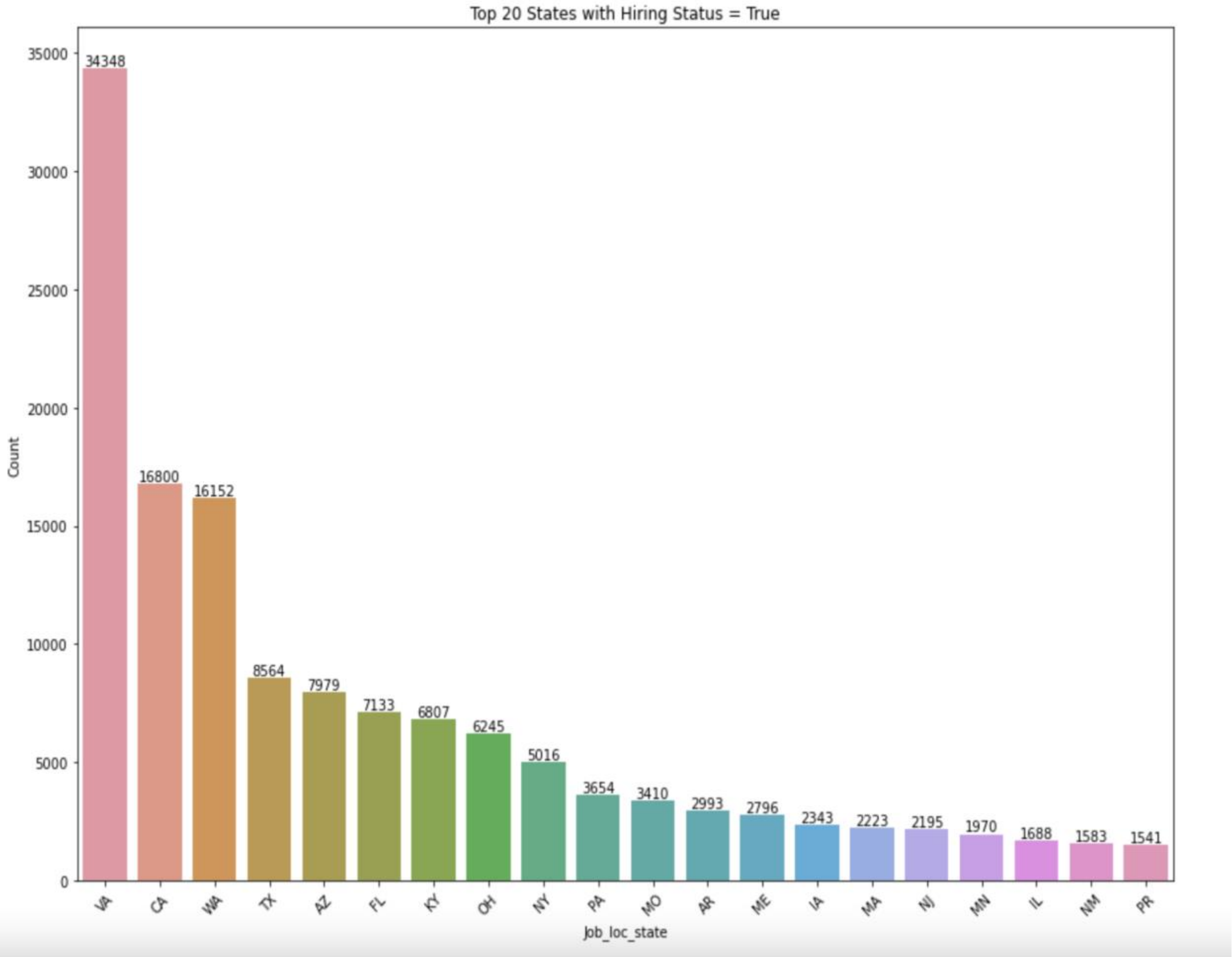


	Missing Count	Missing Percentage
position_type	463783	100.000000
position_description	213665	46.070037
position_summary	203948	43.974876
company_founded	180752	38.973399
company_website	166094	35.812869
company_location_locality	149818	32.303470
company_location_region	138128	29.782894
position_end_date	121413	26.178838
company_location_country	117331	25.298685
company_industry	111821	24.110629
company_size	104397	22.509881
position_start_date	30449	6.565355
company_name	6178	1.332088
position_level	1340	0.288928
position_title	1340	0.288928
company_url	0	0.000000
Source_Identifier	0	0.000000
position_location	0	0.000000
position_order	0	0.000000
Employee_ID	0	0.000000

Data preprocessing step was performed on the dataset to drop unwanted variablles/features. The columns with missing values exceeding a threshold of 35%, namely "position_type", " position _description", "position_summary", "company_founded", company_ website", “company_location_locality”, “company_location_locality” and "company_url", were identified and subsequently dropped from the dataset. So finally we are done with the data cleaning step now our next step is EDA which is to be performed all the 4 datasets.

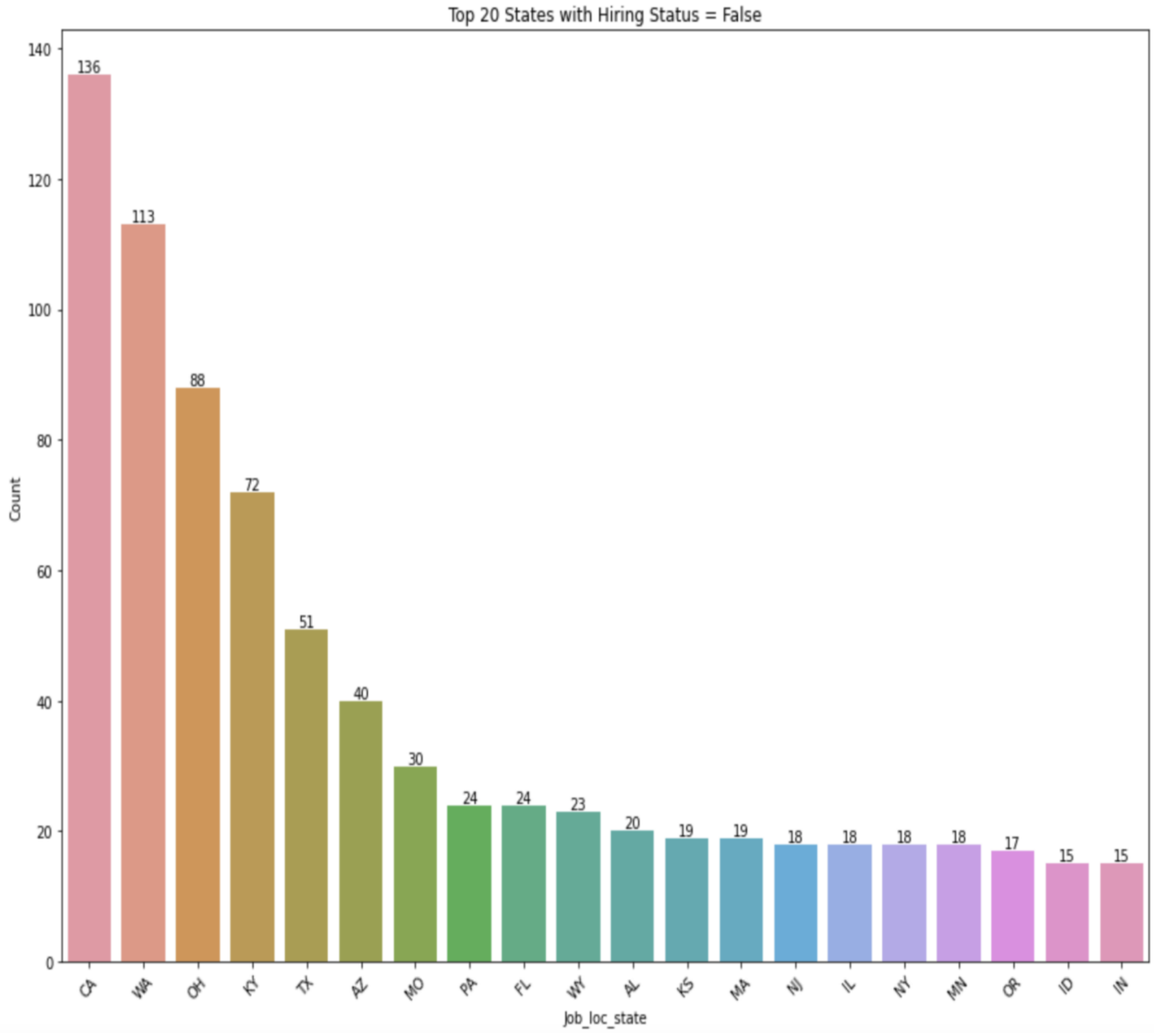
EXPLORATORY DATA ANALYSIS

Q 1. Which state exhibits both a high volume of job openings and a strong success rate in terms of successfully hired candidates?



The graph indicates that Virginia City has the highest number of job openings and active hiring, followed by California in second place. Texas, Arizona, Florida, and New York are also among the top cities with a significant number of job openings. This means that Virginia City has the most job opportunities available and is actively hiring individuals. California is the second-highest city in terms of job openings, and Texas, Arizona, Florida, and New York also have a substantial number of job opportunities.

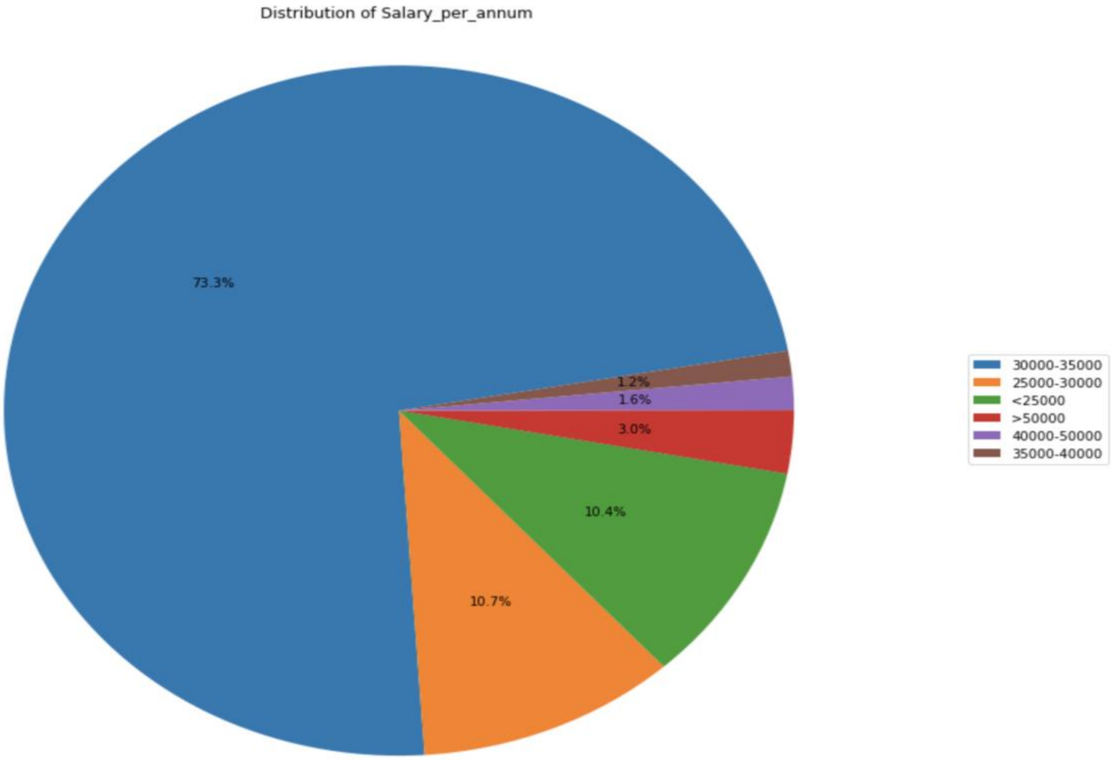
Q 2. Which state exhibits both a low volume of job openings and a weak success rate in terms of hired candidates?



California has the lowest percentage of job openings, and for the purpose of your study, you are not considering Virginia due to a large number of data points.

This suggests that California has the lowest proportion of job openings compared to other cities or states. Due to the high number of data points in Virginia, you have decided to exclude it from your study to focus on other locations.

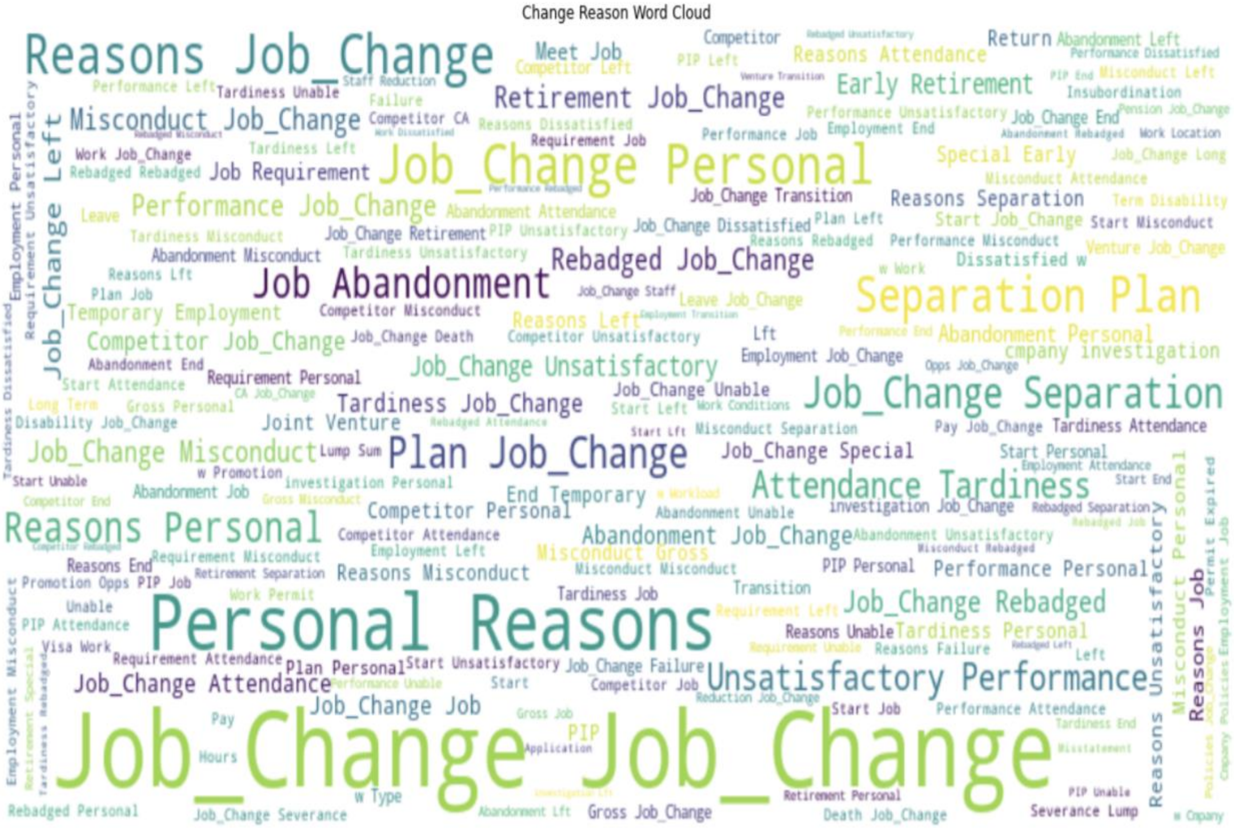
Q 3. What is the overall distribution of salaries across the entire population?



Approximately 73% of people have salaries ranging from $30,000 to $35,000, and this salary band has the highest number of individuals regardless of their years of experience and birth year. It seems that most people earn between $30,000 and $35,000. The second-highest percentage is 10.7% with a salary band of $25,000 to $30,000, and 10.4% earn less than $25,000. Therefore, approximately 95% of the population falls into a salary band lower than $35,000.

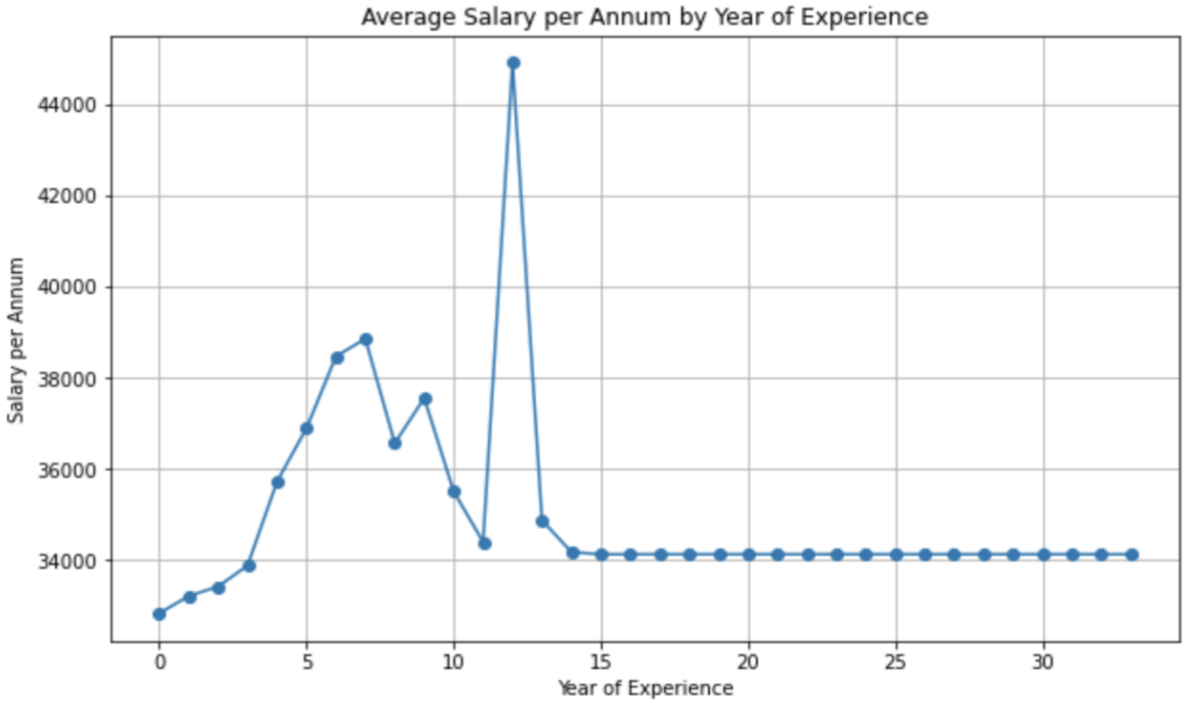
This indicates that the majority of individuals (around 73%) earn salaries between $30,000 and $35,000. This salary band has the highest number of people, regardless of their years of experience or birth year. The second-highest percentage (10.7%) falls into the salary range of $25,000 to $30,000, and 10.4% earn less than $25,000. In total, approximately 95% of the population falls into a salary band lower than $35,000.

Q 4. What is the most cited reason for job changes among employees?



According to the word cloud, the most commonly cited reasons for changing jobs are personal reasons and job change. In addition to unsatisfactory performance, early retirement, and misconduct are also among the top reasons. The word cloud visually represents the frequency of different reasons cited for changing jobs. The most prominent reasons mentioned by individuals are personal reasons and job change. This suggests that people often switch jobs due to personal factors or to pursue new career opportunities. Apart from those, unsatisfactory performance, early retirement, and misconduct are also cited as significant reasons for changing jobs.

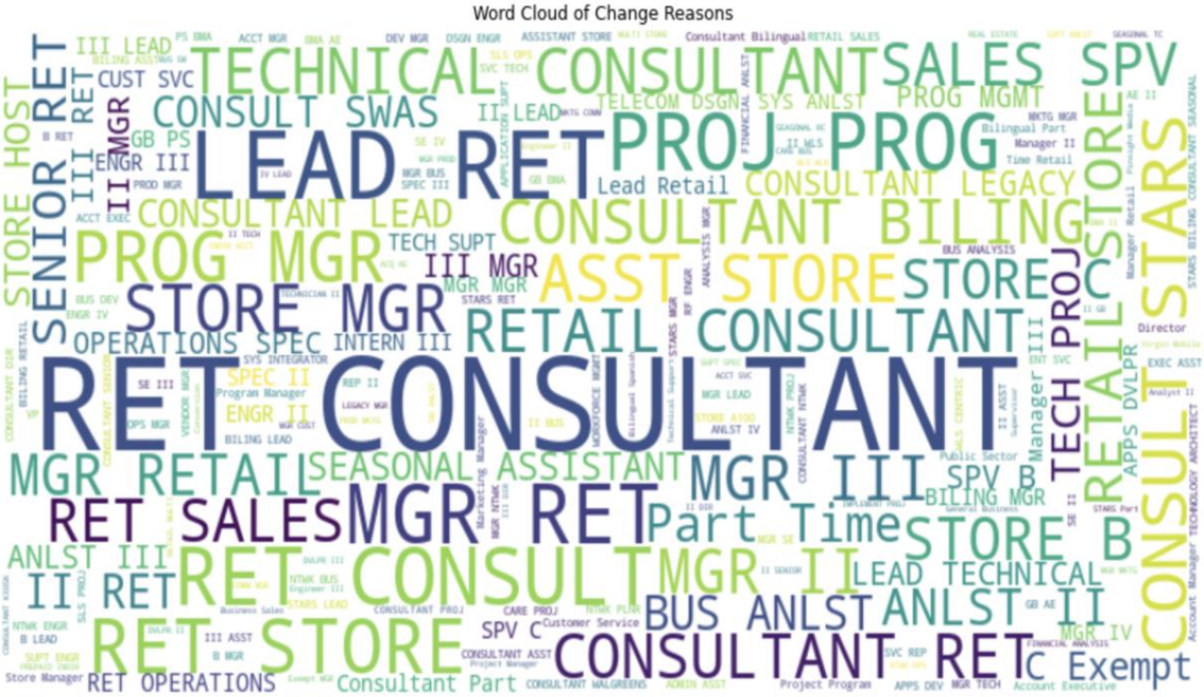
Q 5. How does the average annual salary vary based on years of experience?



Although the line chart is visually appealing, it does not exhibit any significant patterns. Initially, as the years of experience increase, so does the salary, and the highest point is reached between 10 and 15 years of experience. After that, there is no further increase in salary, and the growth becomes stagnant.

The line chart represents the relationship between years of experience and salary. It shows that as the number of years of experience increases, the salary also tends to increase. The highest point on the chart is reached between 10 and 15 years of experience, indicating that individuals with that level of experience tend to earn the highest salaries. However, beyond that point, the chart shows no further increase in salary, suggesting that the salary growth becomes stagnant after a certain threshold.

Q 6. Which Job profile has the best job openings?


Word Cloud of Change Reasons

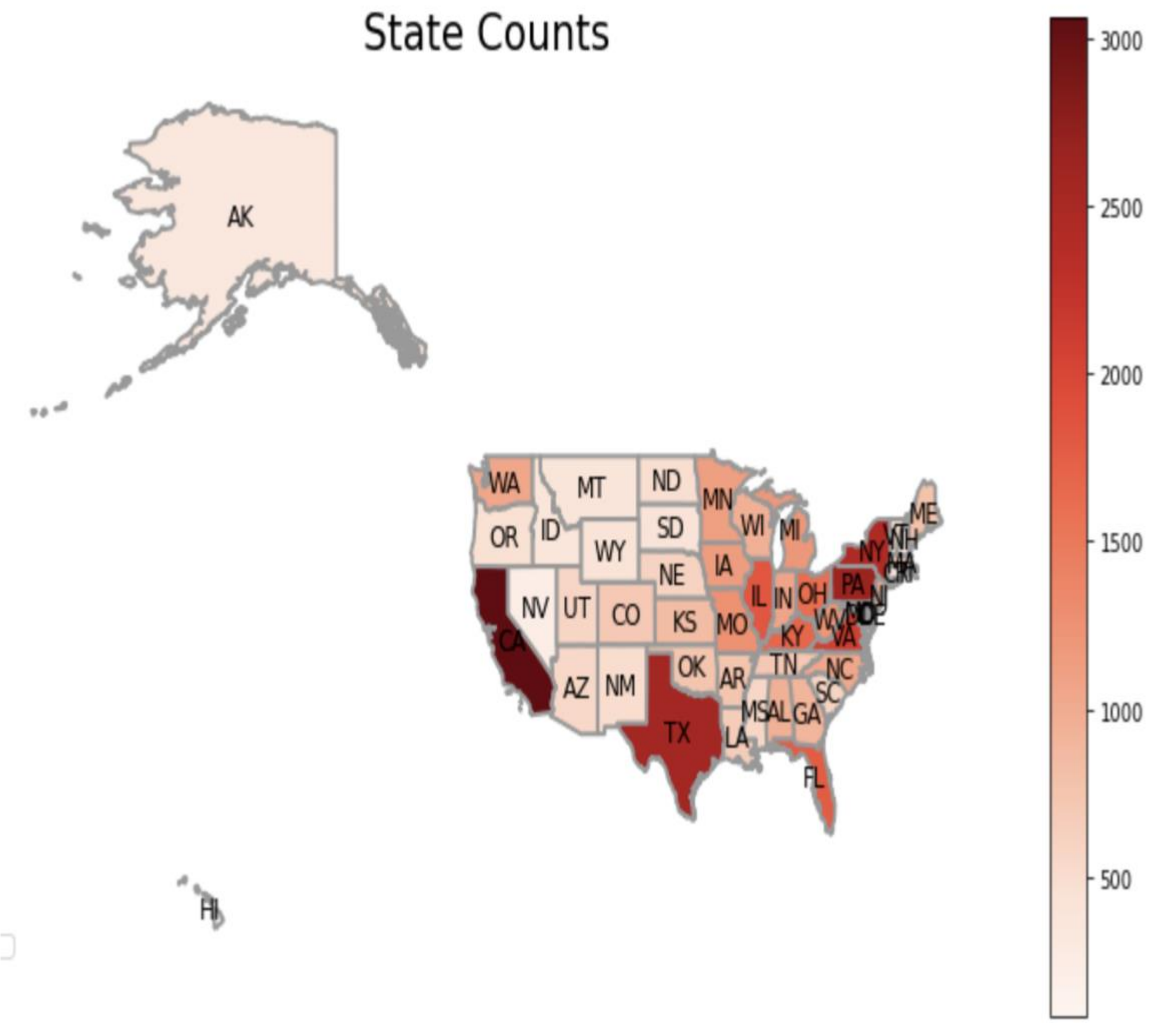
The word cloud suggests that the most popular job profile is a consultant. Additionally, business analysts and store managers are also among the popular profiles.

The word cloud visually displays the frequency of different job profiles mentioned. The most commonly cited job profile is a consultant, indicating that it is a popular choice among individuals. Other popular job profiles mentioned in the word cloud include business analyst and store manager.

GIS Vector Graph

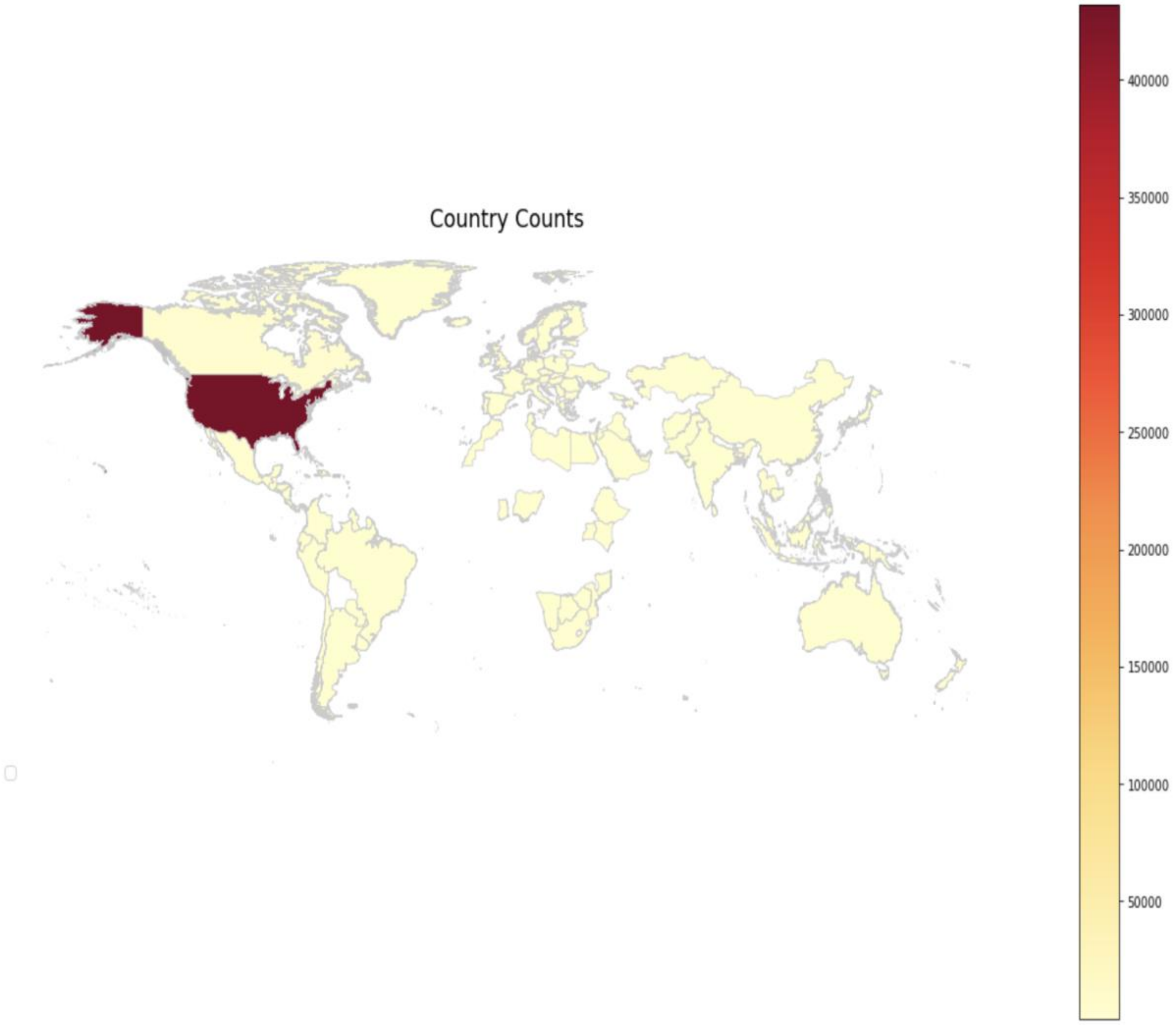
Q 7. Which cities are the most popular for job seekers in the USA?

- **shapefile:** Utilized .shp file that stores the geometric data, such as points, lines, or polygons, representing the spatial features to create this graph.



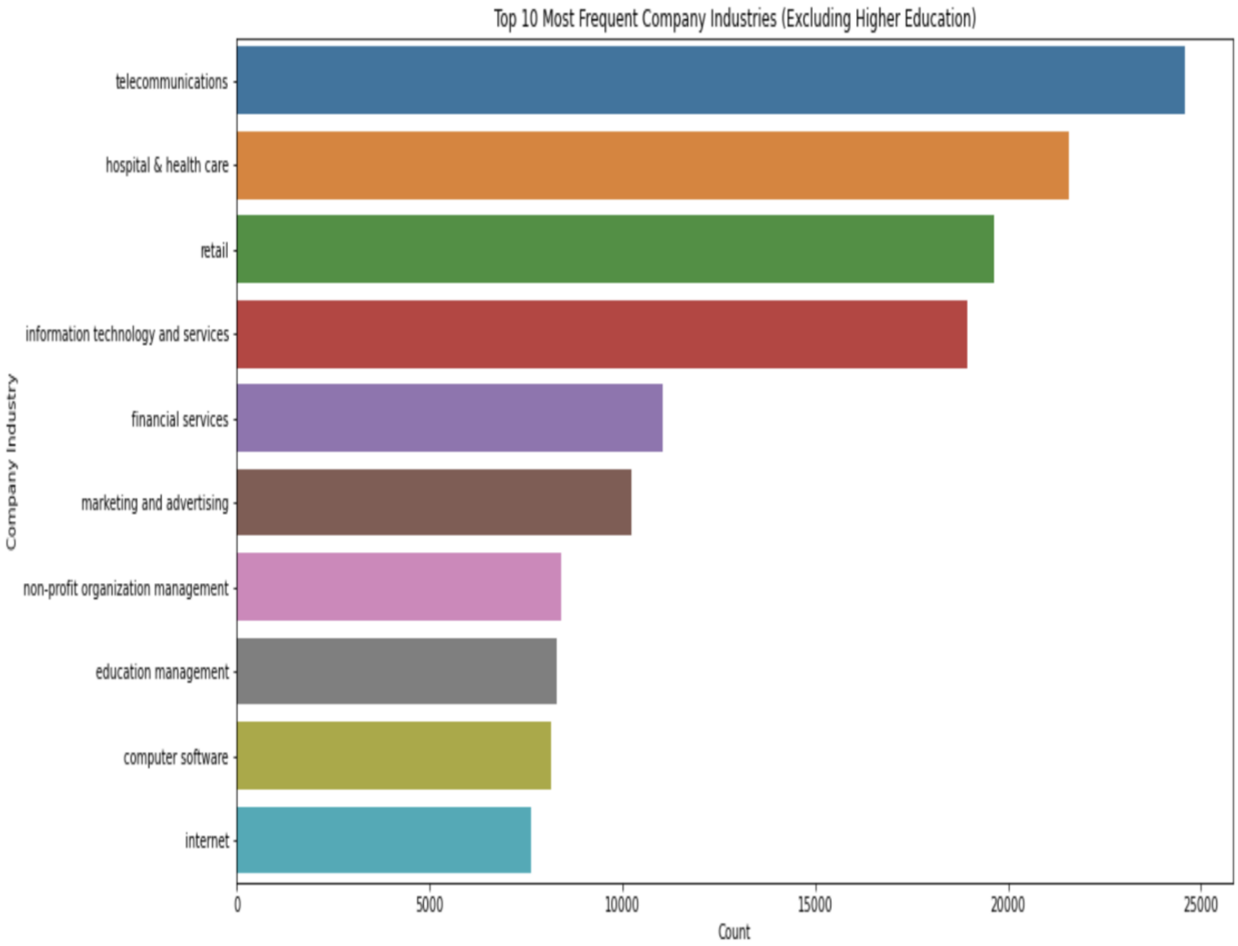
The geographical data graph represents the distribution of job openings across different cities. Based on the graph, it is evident that Texas and California are the top choices in terms of the highest number of job openings. These two states have a significant concentration of job opportunities compared to other cities or states.

Q 8. Which countries have the highest job demand?



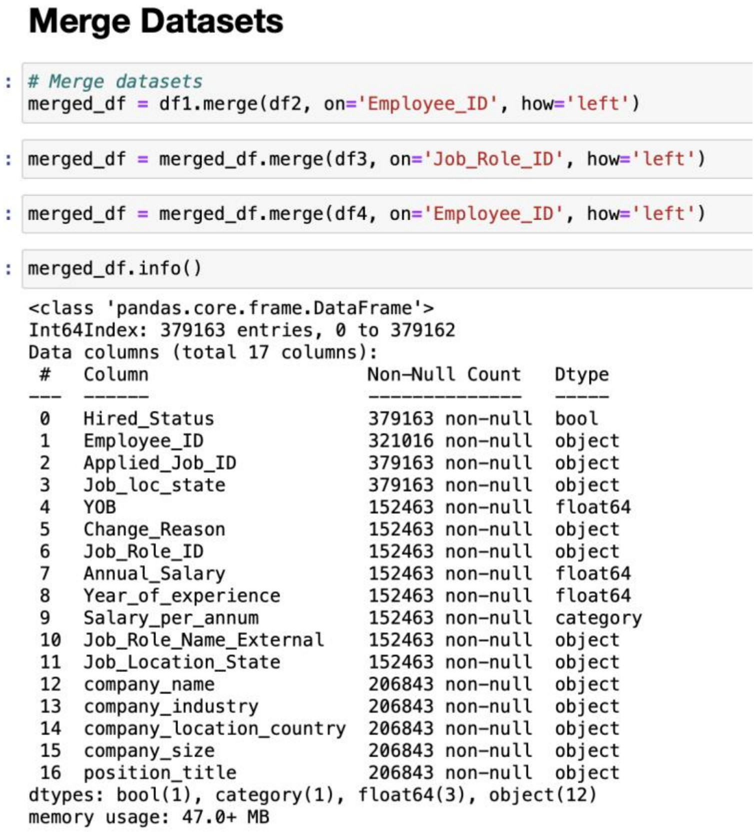
Among the countries, the USA stands out as having a higher frequency of hiring compared to other countries. This implies that the USA has a larger number of job openings and a more active job market compared to other countries represented in the data.

Q 9. Which Industry/Sector has the most number of Jobs?



The bar chart provides information about the number of jobs in different sectors. From the chart, it is clear that the telecommunications and healthcare sectors have the highest number of jobs. These sectors are represented by the tallest bars in the chart, indicating a high frequency of job openings in those industries. This suggests that there are ample opportunities for employment within the telecommunications and healthcare sectors.

MERGE DATASET



Merge Datasets

```
# Merge datasets
merged_df = df1.merge(df2, on='Employee_ID', how='left')
```

```
merged_df = merged_df.merge(df3, on='Job_Role_ID', how='left')
```

```
merged_df = merged_df.merge(df4, on='Employee_ID', how='left')
```

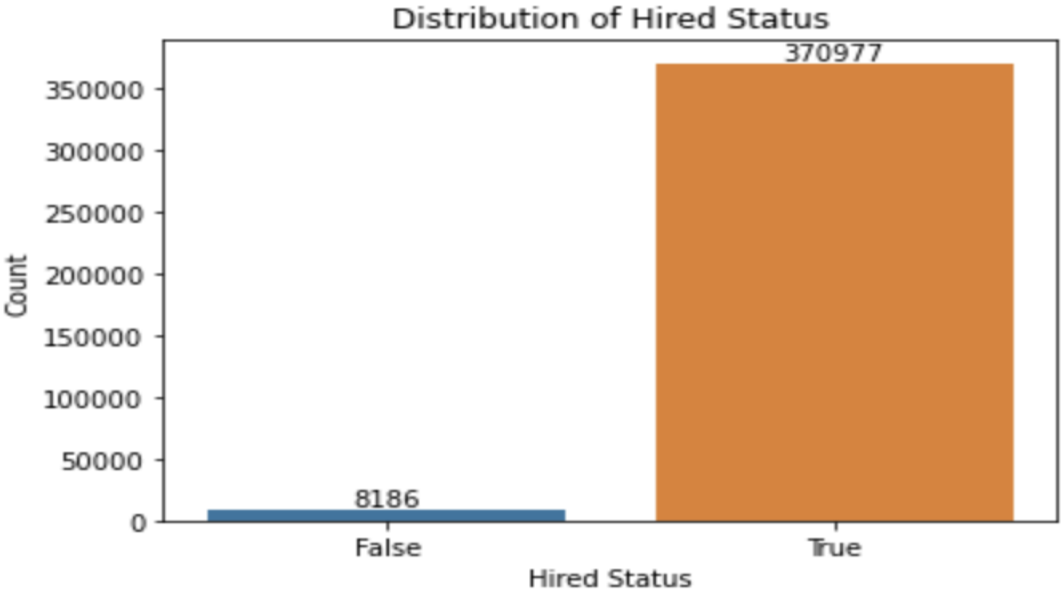
```
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 379163 entries, 0 to 379162
Data columns (total 17 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   Hired_Status             379163 non-null  bool
 1   Employee_ID              321016 non-null  object
 2   Applied_Job_ID           379163 non-null  object
 3   Job_loc_state            379163 non-null  object
 4   YOB                      152463 non-null  float64
 5   Change_Reason            152463 non-null  object
 6   Job_Role_ID              152463 non-null  object
 7   Annual_Salary            152463 non-null  float64
 8   Year_of_experience       152463 non-null  float64
 9   Salary_per_annum         152463 non-null  category
 10  Job_Role_Name_External   152463 non-null  object
 11  Job_Location_State       152463 non-null  object
 12  company_name             206843 non-null  object
 13  company_industry         206843 non-null  object
 14  company_location_country 206843 non-null  object
 15  company_size             206843 non-null  object
 16  position_title           206843 non-null  object
dtypes: bool(1), category(1), float64(3), object(12)
memory usage: 47.0+ MB
```

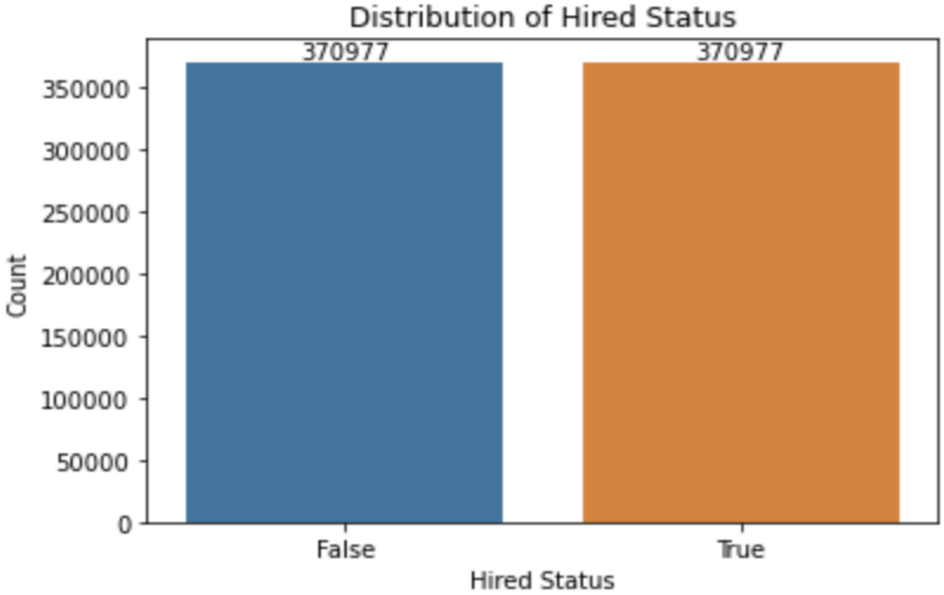
We have merged multiple datasets using Python's merge function and a left join, resulting in a merged dataset with 379,162 data entities and 17 columns. The next step involved applying one-hot encoding to categorical variables and standardization to numerical variables. One-hot encoding converted categorical variables into binary columns, representing unique categories. Standardization scaled numerical variables to have zero mean and unit variance. By performing these transformations, the merged dataset was prepared for further analysis or machine learning tasks. The final dataset, named df_final, incorporated one-hot encoded categorical variables and standardized numerical variables, providing a consolidated and processed dataset for subsequent steps.

Oversampling(SMOTE):-Handling Imbalance Dataset

Before Sampling: -



After Sampling: -



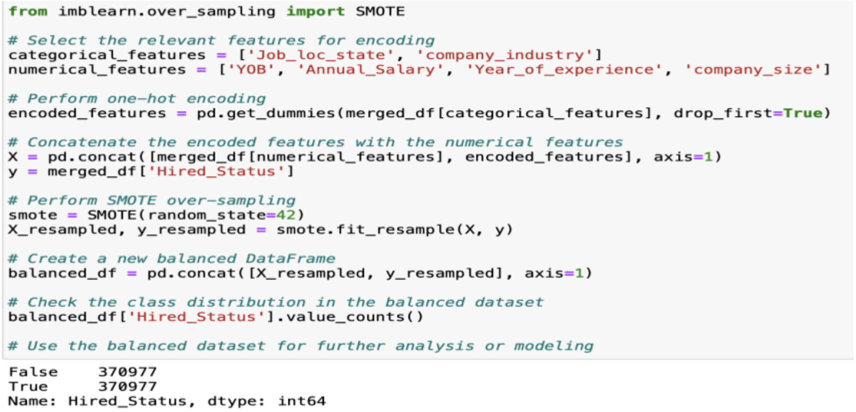
We have a dataset with two classes: "True" and "False." Each class has a certain number of rows associated with it. The task at hand is to perform oversampling using the SMOTE (Synthetic Minority Over-sampling Technique) algorithm and observe how the number of rows changes before and after sampling.

Before oversampling, the dataset consists of 370,977 rows labeled as "True" and 8,186 rows labeled as "False." This class imbalance indicates that the "False" class is underrepresented compared to the "True" class.

To address this class imbalance, we apply the SMOTE algorithm, which generates synthetic samples for the minority class (in this case, the "False" class) based on the existing samples. The algorithm creates new instances by interpolating between similar instances, effectively increasing the representation of the minority class.After performing oversampling with SMOTE, the number of rows in the "True" class remains the same at 370,977, as there is no need to create additional samples for the majority class.

However, the number of rows in the "False" class changes significantly. By applying SMOTE, the "False" class is augmented with synthetic samples, resulting in a new count of 370,977 rows. This equalizes the representation of both classes in the dataset, alleviating the class imbalance issue.

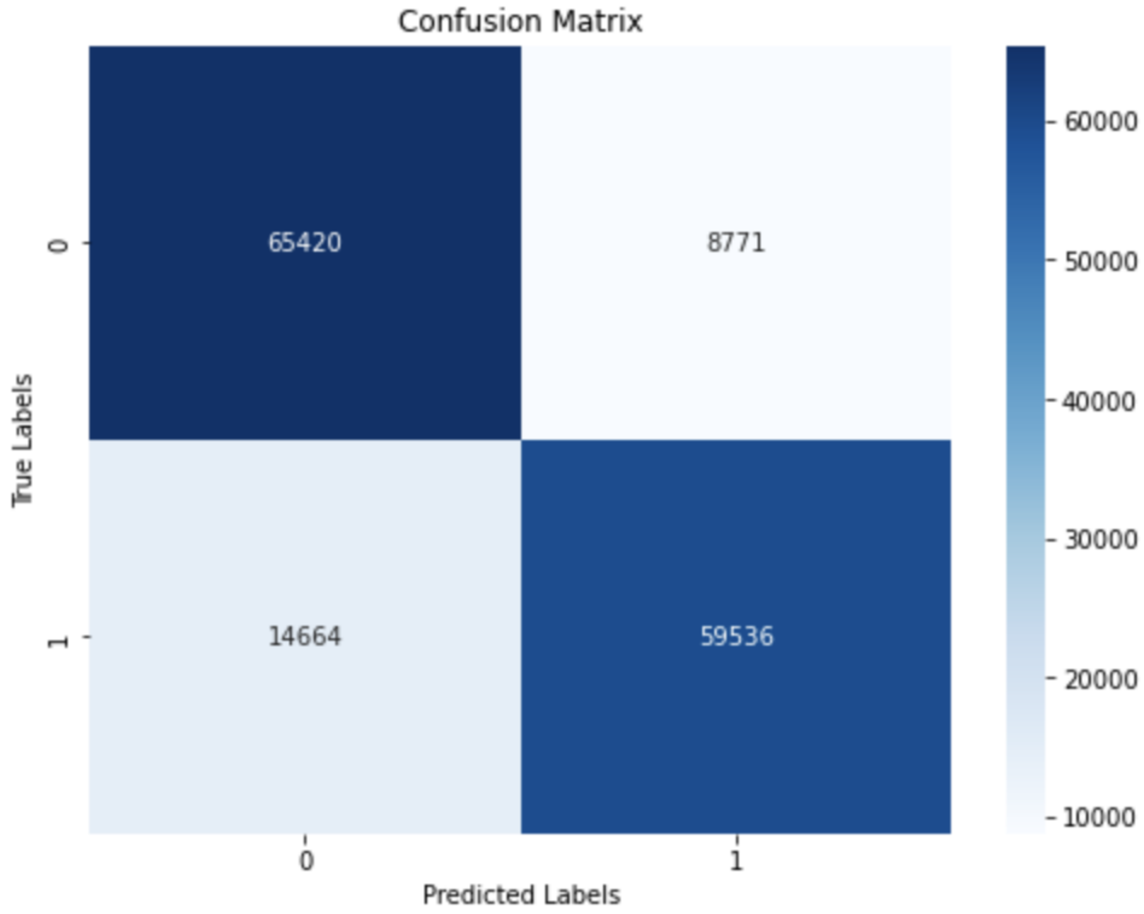
The oversampling process using SMOTE has effectively balanced the dataset, allowing for better modeling and prediction of both classes. This technique can help mitigate the negative impact of imbalanced data on machine learning algorithms, enabling them to learn from and generalize to minority class instances more effectively.

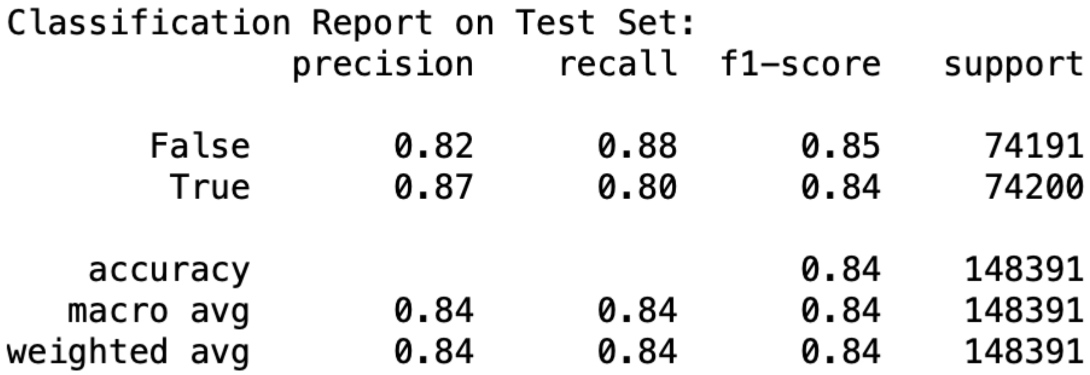


```
from imblearn.over_sampling import SMOTE

# Select the relevant features for encoding
categorical_features = ['Job_loc_state', 'company_industry']
numerical_features = ['YOB', 'Annual_Salary', 'Year_of_experience', 'company_size']

# Perform one-hot encoding
encoded_features = pd.get_dummies(merged_df[categorical_features], drop_first=True)

# Concatenate the encoded features with the numerical features
X = pd.concat([merged_df[numerical_features], encoded_features], axis=1)
y = merged_df['Hired_Status']

# Perform SMOTE over-sampling
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)

# Create a new balanced DataFrame
balanced_df = pd.concat([X_resampled, y_resampled], axis=1)

# Check the class distribution in the balanced dataset
balanced_df['Hired_Status'].value_counts()

# Use the balanced dataset for further analysis or modeling
```

```
False    370977
True     370977
Name: Hired_Status, dtype: int64
```

Model Building – Logistic





```
Classification Report on Test Set:
              precision    recall  f1-score   support

       False       0.82      0.88      0.85     74191
        True       0.87      0.80      0.84     74200

    accuracy                           0.84    148391
   macro avg       0.84      0.84      0.84    148391
weighted avg       0.84      0.84      0.84    148391
```

Based on the provided classification report for a logistic regression model on the test set, we can interpret the results as follows:

Accuracy: The overall accuracy of the model is 84%. Accuracy measures the proportion of correctly classified instances out of the total instances in the dataset. In this case, the model is correctly predicting the class label for 84% of the instances in the test set.

Precision: Precision is a measure of the model's ability to correctly identify positive instances out of the total instances it predicted as positive. For the "False" class, the precision is 0.82, indicating that out of all instances predicted as "False," 82% of them are actually "False." Similarly, for the "True" class, the precision is 0.87, meaning that out of all instances predicted as "True," 87% of them are actually "True." Higher precision values indicate better performance in correctly identifying positive instances.

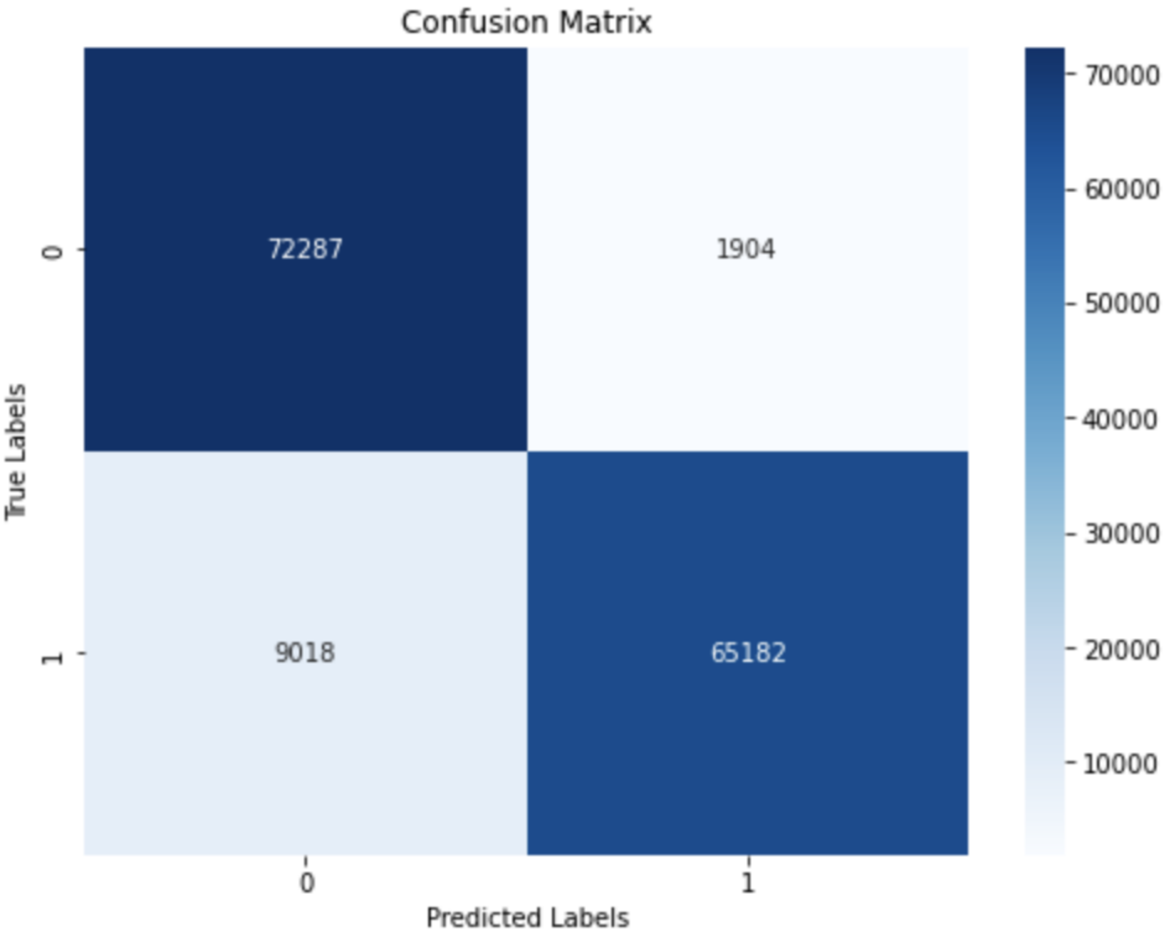
Recall: Recall, also known as sensitivity or true positive rate, measures the model's ability to correctly identify positive instances out of the total actual positive instances in the dataset. For the "False" class, the recall is 0.88, indicating that the model is able to correctly identify 88% of the actual "False" instances. Similarly, for the "True" class, the recall is 0.80, meaning that the model can correctly identify 80% of the actual "True" instances. Higher recall values indicate better performance in capturing positive instances.

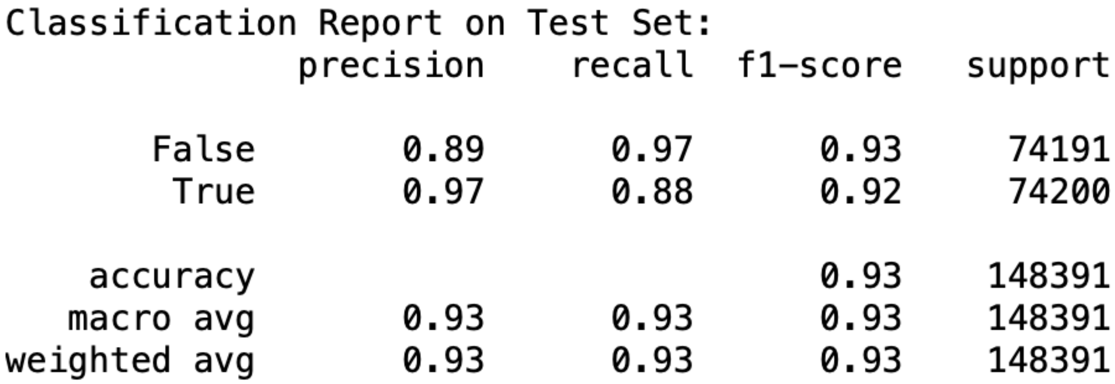
F1-score: It considers both precision and recall, making it useful when the dataset is imbalanced. For the "False" class, the F1-score is 0.85, while for the "True" class, it is 0.84. Higher F1-scores indicate better overall performance in terms of both precision and recall.

Support: Support refers to the number of instances in each class in the test set. In this case, the "False" class has 74,191 instances, while the "True" class has 74,200 instances.

Considering the overall results, the logistic regression model demonstrates reasonably good performance with an accuracy of 84%. It shows relatively balanced precision, recall, and F1-scores for both classes, indicating that the model is performing well for both "False" and "True" instances.

Model Building – XG Boost





```
Classification Report on Test Set:
              precision    recall  f1-score   support

       False       0.89      0.97      0.93     74191
        True       0.97      0.88      0.92     74200

    accuracy                           0.93    148391
   macro avg       0.93      0.93      0.93    148391
weighted avg       0.93      0.93      0.93    148391
```

Based on the provided classification report for an XGBoost model on the test set, we can interpret the results as follows:

Accuracy: The overall accuracy of the model is 92%. Accuracy measures the proportion of correctly classified instances out of the total instances in the dataset. In this case, the model is correctly predicting the class label for 92% of the instances in the test set.

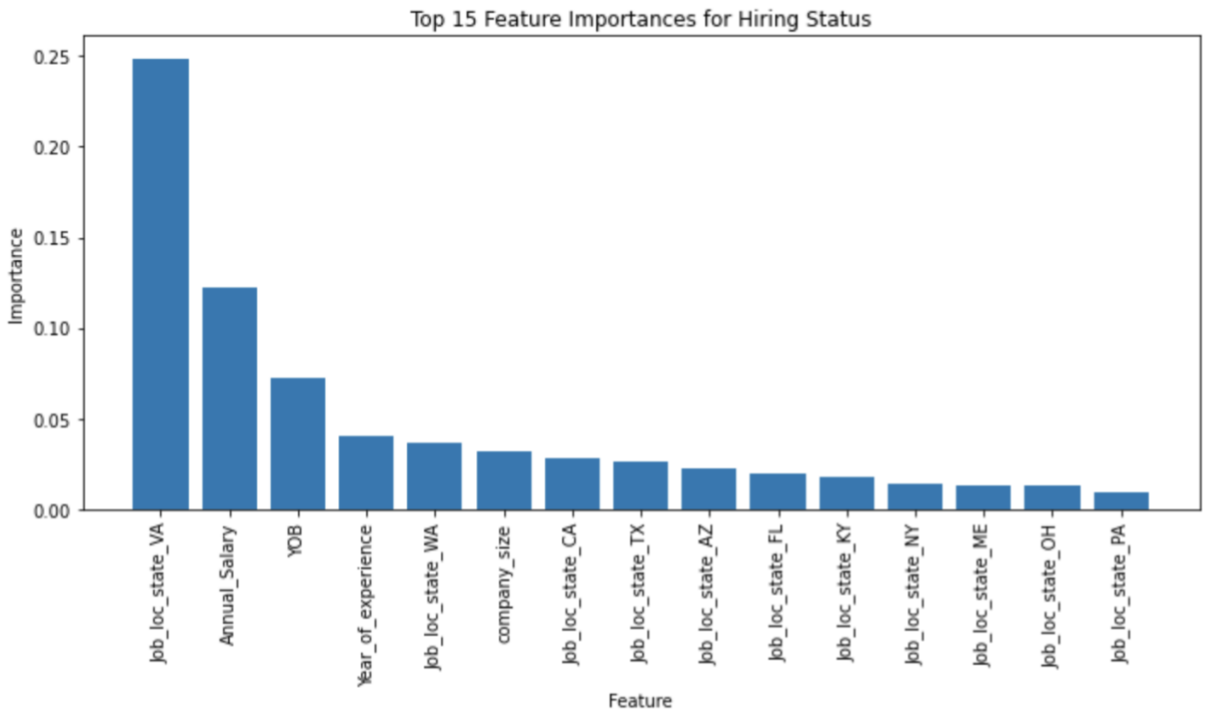
Precision: Precision is a measure of the model's ability to correctly identify positive instances out of the total instances it predicted as positive. For the "False" class, the precision is 0.89, indicating that out of all instances predicted as "False," 89% of them are actually "False." Similarly, for the "True" class, the precision is 0.97, meaning that out of all instances predicted as "True," 97% of them are actually "True." Higher precision values indicate better performance in correctly identifying positive instances.

Recall: Recall, also known as sensitivity or true positive rate, measures the model's ability to correctly identify positive instances out of the total actual positive instances in the dataset. For the "False" class, the recall is 0.97, indicating that the model is able to correctly identify 97% of the actual "False" instances. Similarly, for the "True" class, the recall is 0.88, meaning that the model can correctly identify 88% of the actual "True" instances. Higher recall values indicate better performance in capturing positive instances.

F1-score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. It considers both precision and recall, making it useful when the dataset is imbalanced. For the "False" class, the F1-score is 0.93, while for the "True" class, it is 0.92. Higher F1-scores indicate better overall performance in terms of both precision and recall. Support: Support refers to the number of instances in each class in the test set. In this case, the "False" class has 74,191 instances, while the "True" class has 74,200 instances.

Considering the overall results, the XGBoost model demonstrates excellent performance with an accuracy of 92%. It shows high precision, recall, and F1-scores for both classes, indicating that the model is performing well for both "False" and "True" instances. The model shows a slightly higher recall for the "False" class, indicating that it is better at capturing the actual "False" instances. However, overall, the model achieves a balanced performance between precision and recall for both classes.

FEATURE IMPORTANCE – RANDOM FOREST



Feature importance is a technique used in machine learning, particularly with the Random Forest algorithm, to determine the relevance or significance of input features in predicting the target variable. Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. The importance of each feature is calculated based on how much each feature decreases the impurity in the decision trees. The impurity reduction caused by a particular feature across all the trees in the Random Forest model is averaged to determine its importance.

There are different methods to measure feature importance in Random Forests, including Gini importance and Mean Decrease Impurity (MDI). Gini importance measures the total reduction in the Gini index when a particular feature is used for splitting the data. A higher reduction in the Gini index indicates a more important feature. MDI computes the average impurity reduction across all trees for each feature.

Annual Salary, YOB, Year of Experience, Job_loc_state, and Company Size is the most important features to predict the hiring status of the employee.

AutoML

AutoML (Automated Machine Learning) is a powerful technology that has gained significant attention and adoption in various industries, including talent acquisition. AutoML refers to the automated process of building, optimizing, and deploying machine learning models without requiring extensive manual intervention or expertise in data science. In the talent acquisition sector, AutoML offers numerous benefits and has the potential to revolutionize the way organizations attract, assess, and hire talent.

Streamlined Recruitment Process: AutoML simplifies and accelerates the recruitment process by automating various time-consuming tasks. It can automatically analyze and process large volumes of candidate data, including resumes, cover letters, and application forms, extracting relevant information and creating structured profiles for each candidate.

Enhanced Candidate Screening: AutoML algorithms can be trained to evaluate candidate profiles against specific job requirements, identifying the most suitable candidates for further consideration. By leveraging machine learning techniques, AutoML models can learn from historical hiring data and make predictions about a candidate's potential fit for a particular role.

Bias Mitigation: One of the critical challenges in talent acquisition is bias, both conscious and unconscious. AutoML systems can be designed to minimize bias by using diverse training data and employing fairness metrics during model development. This helps promote fairness, diversity, and inclusion in the candidate selection process.

Personalized Candidate Experience: AutoML can improve the candidate experience by personalizing interactions and recommendations. It can leverage natural language processing to analyze candidate preferences, career goals, and past experiences, allowing organizations to provide tailored job recommendations and relevant content to engage candidates effectively.

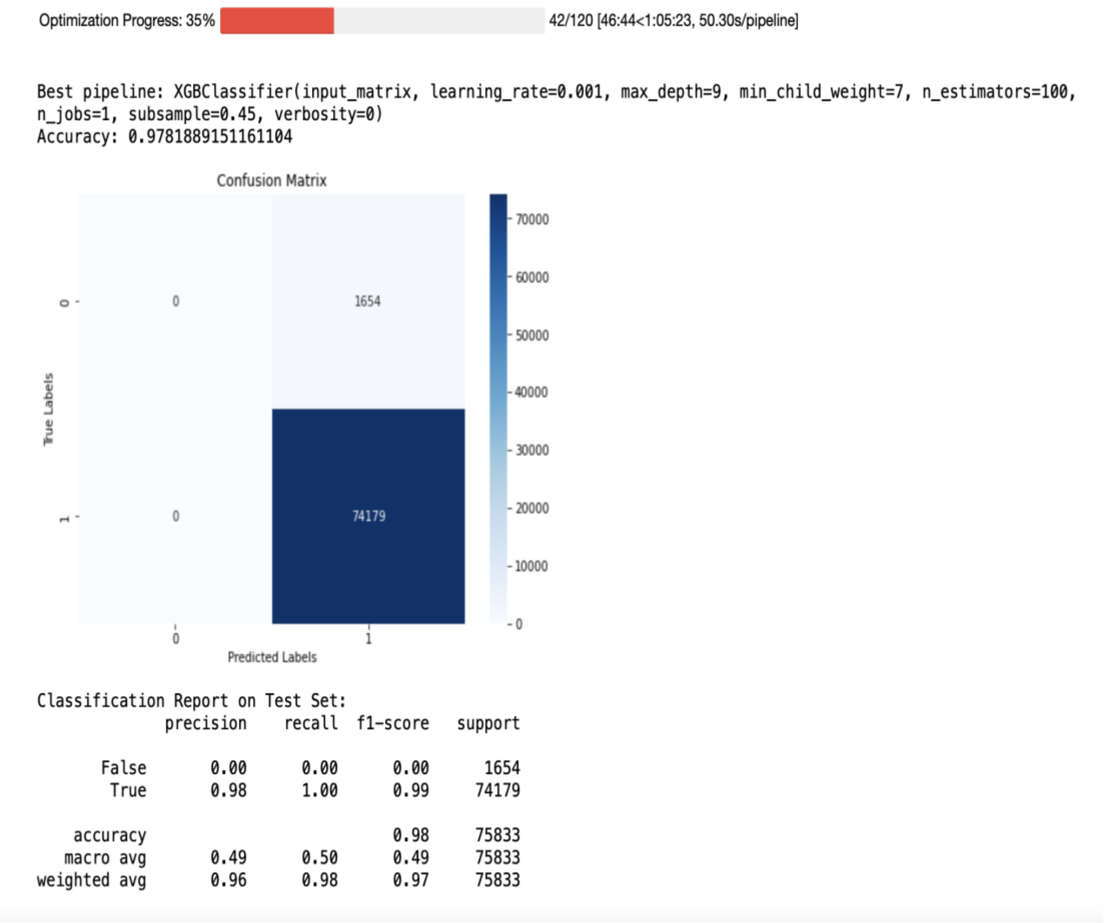
Reducing Time-to-Hire: AutoML can significantly reduce the time-to-hire by automating manual tasks, such as resume screening and candidate ranking. This efficiency improvement allows recruiters and hiring managers to focus on more strategic activities, such as building.

Scalability and Cost Efficiency: AutoML platforms provide scalability, allowing organizations to handle a large number of job applications and candidate profiles without compromising efficiency. Additionally, the automation of repetitive tasks reduces manual effort and associated costs, enabling recruiters to handle a higher volume of hiring processes within limited resources.

In conclusion, AutoML is transforming the talent acquisition sector by automating and augmenting various aspects of the recruitment process. By leveraging machine learning and data-driven insights, AutoML enables organizations to make more informed hiring decisions, reduce bias, improve candidate experiences, and optimize resource allocation. As the field of AutoML continues to advance, its potential to revolutionize talent acquisition and drive better hiring outcomes is becoming increasingly apparent.

Without Importing Torch Library (No consideration of deep learning model): -

- XG Boost is the Winner 🏆. XG Boost Classifier is the best model to solve such talent management problems with the max_depth=9, minimum child weight=7 and learning_rate=0.001.



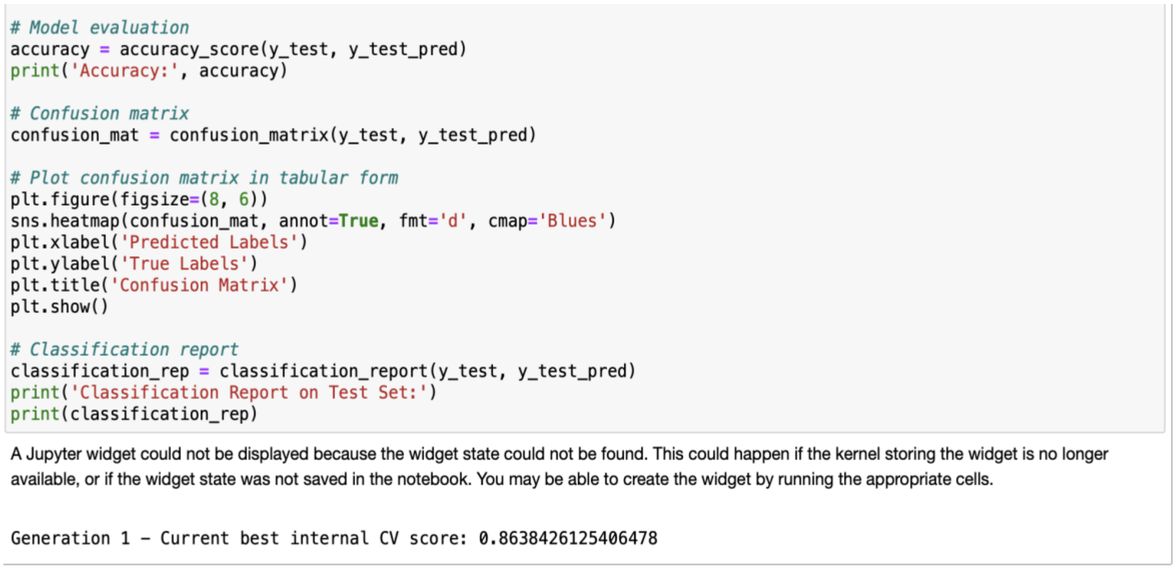
```
Optimization Progress: 35%  42/120 [46:44<1:05:23, 50.30s/pipeline]

Best pipeline: XGBClassifier(input_matrix, learning_rate=0.001, max_depth=9, min_child_weight=7, n_estimators=100,
n_jobs=1, subsample=0.45, verbosity=0)
Accuracy: 0.9781889151161104
```

```
Classification Report on Test Set:
              precision    recall  f1-score   support

       False       0.00      0.00      0.00      1654
        True       0.98      1.00      0.99     74179

    accuracy                           0.98     75833
   macro avg       0.49      0.50      0.49     75833
weighted avg       0.96      0.98      0.97     75833
```

With Consideration of Deep Learning Models (Imported Torch library): -

The result you provided indicates the performance of a model generated by an AutoML process during the first generation. The internal CV score, which stands for cross-validation score, is a measure of how well the model performs on the training data.

In this case, the current best internal CV score is 0.8638426125406478. This score typically ranges between 0 and 1, with higher values indicating better performance. It suggests that the model achieved a relatively high accuracy or predictive capability on the training data during the first generation.



```
# Model evaluation
accuracy = accuracy_score(y_test, y_test_pred)
print('Accuracy:', accuracy)

# Confusion matrix
confusion_mat = confusion_matrix(y_test, y_test_pred)

# Plot confusion matrix in tabular form
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_mat, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Confusion Matrix')
plt.show()

# Classification report
classification_rep = classification_report(y_test, y_test_pred)
print('Classification Report on Test Set:')
print(classification_rep)
```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

```
Generation 1 - Current best internal CV score: 0.8638426125406478
```

CONCLUSION & FUTURE WORK

In conclusion, the findings of this project highlight the importance of key numerical features, such as Annual Salary, Year of Birth (YOB), Year of Experience, and Job location, in predicting the hiring status of employees. The XG Boost model, specifically configured with a max_depth of 9, minimum child weight of 7, and learning_rate of 0.001, emerged as the most effective model for this prediction task. The integration of AI technology holds immense potential in enhancing the candidate selection process by automating repetitive tasks and efficiently identifying and ranking the best matches between resumes and job roles. Furthermore, the analysis reveals that the telecommunication and healthcare sectors exhibit the highest demand for jobs, while cities such as Texas, California, New York, and Pennsylvania are particularly popular among job seekers. Finally, the United States emerges as the leading country for job seekers, offering a multitude of employment opportunities. These insights provide valuable guidance for organizations and job seekers alike, facilitating informed decision-making in talent management and job search strategies.

In future work, there are several avenues to explore based on the findings of this project. One potential direction is to utilize other talent management datasets using APIs, incorporating additional features such as skills, university information, and resume details. This expanded dataset can provide a more comprehensive understanding of candidate profiles and further improve the accuracy of predictions. Deep learning techniques, coupled with Natural Language Processing (NLP), can be applied to extract meaningful insights from these datasets, enabling a deeper understanding of candidate attributes and preferences.To gain a deeper understanding of the business implications and context surrounding talent management data, it would be valuable to delve into the intricacies of Applicant Tracking Systems (ATS), Human Resource Information Systems (HRIS), and other talent management systems. Conducting a survey or interviews with professionals in the field of people analytics and HR analytics can provide valuable insights into the challenges, best practices, and emerging trends in talent management. This knowledge can further inform the development of more robust and contextually relevant predictive models.

REFERENCES:

- ❑ *Altamiranda, D. (2022, August 18). Artificial Intelligence in Recruitment and Talent Management. Avature. https://www.avature.net/blogs/artificial-intelligence-in-recruitment-and-talent-management/*
- ❑ *AI, E. (2022, January 18). How ai fuels an integrated approach to talent management. Eightfold. https://eightfold.ai/blog/ai-talent-management/*
- ❑ *Sydell, E. (2022, April 14). Council Post: Rethinking Talent Management Through AI. Forbes. https://www.forbes.com/sites/forbeshumanresourcescouncil/2022/04/14/rethinking-talent-management-through-ai/*
- ❑ *Mastrogiovanni, S. (2021, August 30). AI and the Future of Talent Management | International Association for Human Resources Information Management. International Association for Human Resources Information Management. https://www.ihrim.org/2021/08/ai-and-the-future-of-talent-management-by-sergio-mastrogiovanni/*
- ❑ *The Power of AI to Revolutionize Talent Management - Spiceworks. (n.d.). Spiceworks. https://www.spiceworks.com/tech/artificial-intelligence/guest-article/the-power-of-ai-to-revolutionize-talent-management/*
- ❑ D. (2022, April 27). *Transforming Talent Management Experience with Job Market Data—GoodPeople*. Medium. https://medium.com/sfu-cspmp/transforming-talent-management-experience-with-job-market-data-goodpeople-8df9d759e31c
- ❑ Blair, C. (2023, April 12). *Researching Scalable Talent Management Datasets*. Medium. https://curtisblair.medium.com/researching-scalable-talent-management-datasets-e119602355fd
- ❑ Vulpen, E. V. (2019, September 30). *7 HR Data Sets for People Analytics*. AIHR. https://www.aihr.com/blog/hr-data-sets-people-analytics/