STAT 206

Statistical Computing for Business Analytics

Professor Alfonso Landeros

IBM HR Analytics Employee Attrition & Performance

Cindy Miao, Haiying Lin, Susie Liang

March 22, 2024

1. Data Summary

1.1 About Dataset

We use the "IBM HR Analytics Employee Attrition & Performance" dataset from Kaggle, which offers a comprehensive look at various factors that might influence an employee's decision to leave the company. There are 16 numerical and 19 categorical variables in the dataset that encompass a range of factors from demographic details like age and gender to job-specific information such as role and travel frequency, alongside performance indicators.

1.2 Dataset

https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset/data

1.3 Problem Statement

Our business question is based on IBM's employee information records to analyze the factors most affecting employee attrition. We aim to discover the relationship between employee's personal information and performance records with their attrition status. This analysis can help us get strategies to enhance understanding of employee performances, and potentially reduce overall attrition rates. This insight is crucial for creating a supportive work environment that encourages employees to stay.

We aim to leverage this dataset to discover the patterns and insights of employee attrition. We plan to use logistic regression, a statistical method that estimates the probability of a binary outcome based on one or more predictor variables, to find the variables that most affect employee attrition. Besides, classification models will group employees into 'attrited' or 'not attrited' categories based on their characteristics.

1.4 Data Cleaning

We removed the unnecessary categories: "EmployeeCount", "EmployeeNumber", "Over18", and "StandardHours" since all rows in these three variables are the same. We also change Attrition to factors "Yes" = 1 & "No" = 0. Here is our new dataset, with 31 variables of 1470 observations.

```
# remove the unnecessary categories: EmployeeCount, EmployeeNumber,Over18, and StandardHours, left 31 variables
IBM_Employee = select(IBM_Employee, Not([:EmployeeCount, :EmployeeNumber, :Over18, :StandardHours]))
```

```
IBM_Employee.Attrition = map(x -> x == "Yes" ? 1 : 0, IBM_Employee.Attrition)
first(IBM_Employee, 5)
```

✓ 0.0s

5×31 DataFrame

Row	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField
	Int64	Int64	String	Int64	String	Int64	Int64	String
1	41	1	Travel_Rarely	1102	Sales	1	2	Life Sciences
2	49	0	Travel_Frequently	279	Research & Development	8	1	Life Sciences
3	37	1	Travel_Rarely	1373	Research & Development	2	2	Other
4	33	0	Travel_Frequently	1392	Research & Development	3	4	Life Sciences
5	27	0	Travel_Rarely	591	Research & Development	2	1	Medical

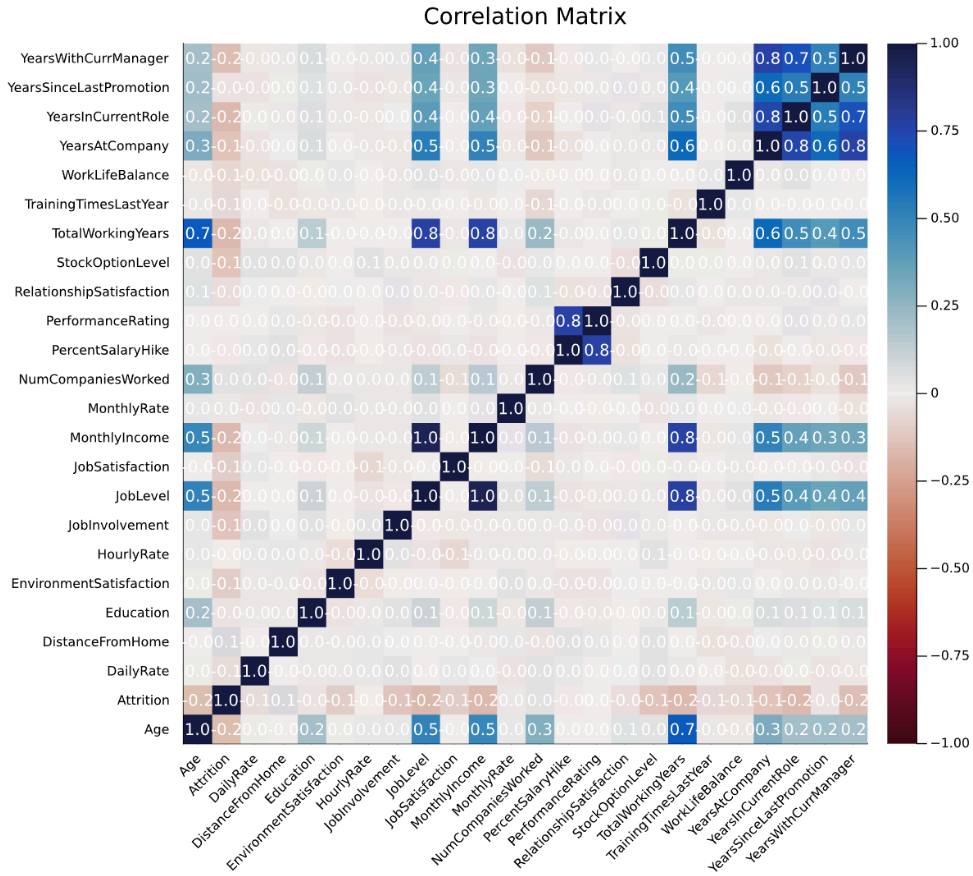
2. Exploratory Data Analysis

2.1 Data Summary

Row	variable	mean	std	min	max	q25	median	q75	nmissing	eltype
	Symbol	Union...	Union...	Any	Any	Union...	Union...	Union...	Int64	DataType
1	Age	36.9238	9.13537	18	60	30.0	36.0	43.0	0	Int64
2	Attrition	0.161224	0.367863	0	1	0.0	0.0	0.0	0	Int64
3	BusinessTravel			Non-Travel	Travel_Rarely				0	String
4	DailyRate	802.486	403.509	102	1499	465.0	802.0	1157.0	0	Int64
5	Department			Human Resources	Sales				0	String
6	DistanceFromHome	9.19252	8.10686	1	29	2.0	7.0	14.0	0	Int64
7	Education	2.91293	1.02416	1	5	2.0	3.0	4.0	0	Int64
8	EducationField			Human Resources	Technical Degree				0	String
9	EnvironmentSatisfaction	2.72177	1.09308	1	4	2.0	3.0	4.0	0	Int64
10	Gender			Female	Male				0	String
11	HourlyRate	65.8912	20.3294	30	100	48.0	66.0	83.75	0	Int64
12	JobInvolvement	2.72993	0.711561	1	4	2.0	3.0	3.0	0	Int64
13	JobLevel	2.06395	1.10694	1	5	1.0	2.0	3.0	0	Int64
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	OverTime			No	Yes				0	String
21	PercentSalaryHike	15.2095	3.65994	11	25	12.0	14.0	18.0	0	Int64
22	PerformanceRating	3.15374	0.360824	3	4	3.0	3.0	3.0	0	Int64
23	RelationshipSatisfaction	2.71224	1.08121	1	4	2.0	3.0	4.0	0	Int64
24	StockOptionLevel	0.793878	0.852077	0	3	0.0	1.0	1.0	0	Int64
25	TotalWorkingYears	11.2796	7.78078	0	40	6.0	10.0	15.0	0	Int64
26	TrainingTimesLastYear	2.79932	1.28927	0	6	2.0	3.0	3.0	0	Int64
27	WorkLifeBalance	2.76122	0.706476	1	4	2.0	3.0	3.0	0	Int64
28	YearsAtCompany	7.00816	6.12653	0	40	3.0	5.0	9.0	0	Int64
29	YearsInCurrentRole	4.22925	3.62314	0	18	2.0	3.0	7.0	0	Int64
30	YearsSinceLastPromotion	2.18776	3.22243	0	15	0.0	1.0	3.0	0	Int64
31	YearsWithCurrManager	4.12313	3.56814	0	17	2.0	3.0	7.0	0	Int64

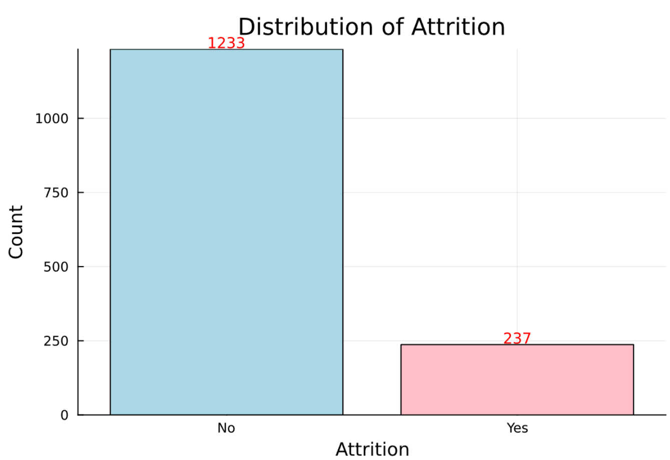
2.2 Correlation Matrix

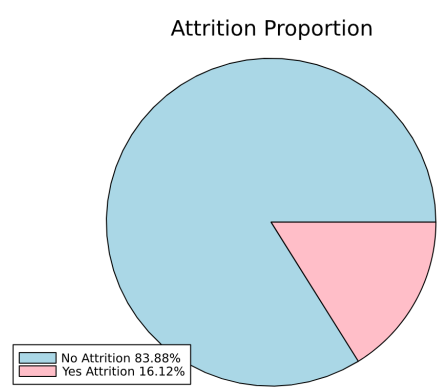
24 variables of numerical data



2.3 Data Visualization

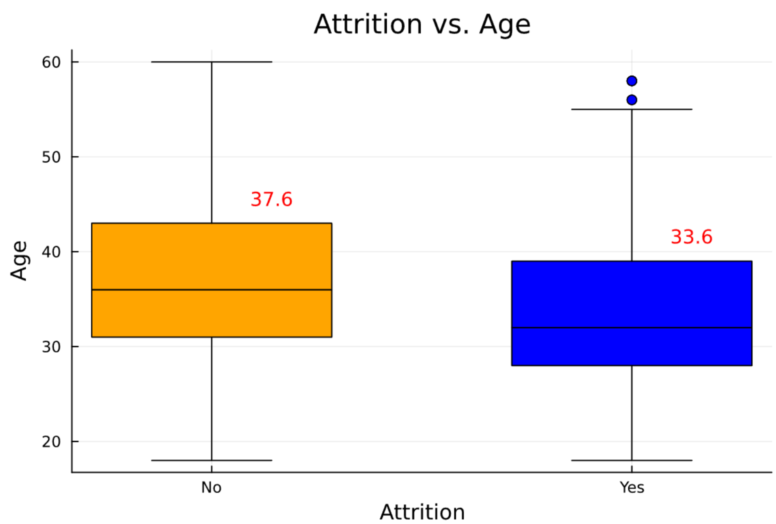
*2.3.1 **Attrition** — Dependent Variable*





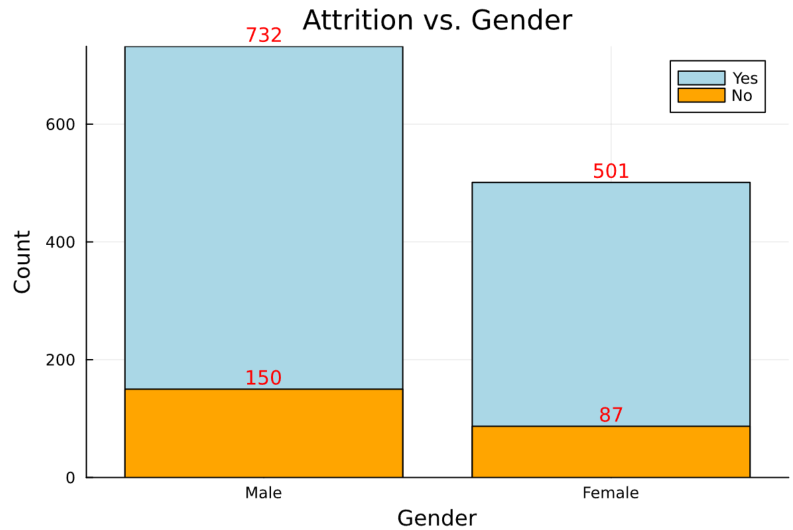
The bar graph shows a total of 1,233 employees did not experience attrition (No), and 237 did (Yes). The pie chart displays the same information as proportions, with 83.9% of employees staying and 16.1% leaving. Because in the original distribution, “Yes” compared with “No” is 83% to 16%, so set the cut-off as 80% for prediction.

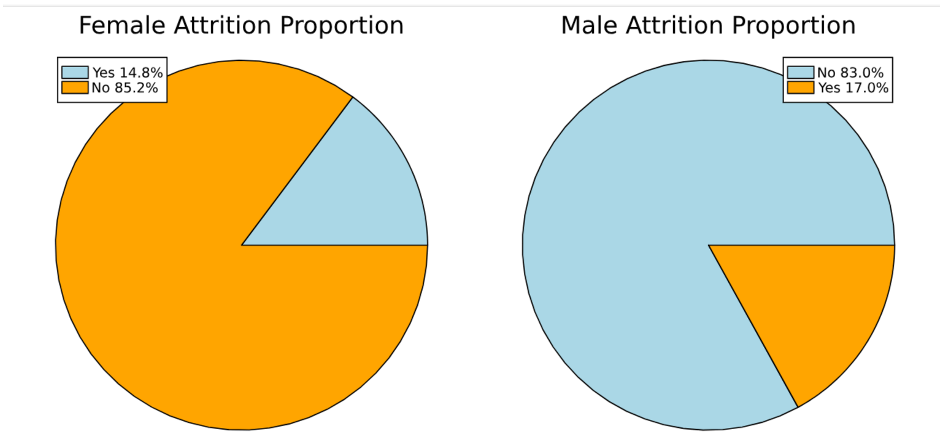
*2.3.2 **Age** — Grouped Box Plot*



Employees who left tend to be younger, with a mean age of about 33.6 years, while those who stayed have a mean age of approximately 37.6 years.

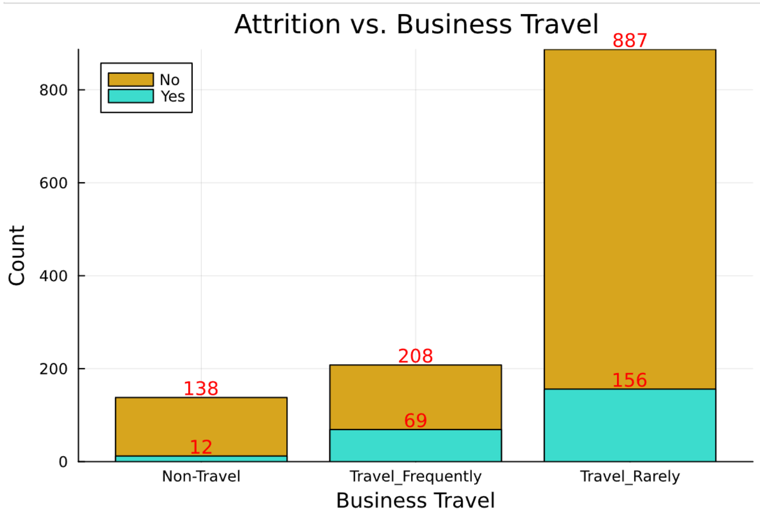
*2.3.3 **Gender** — Grouped Bar & Pie Chart*

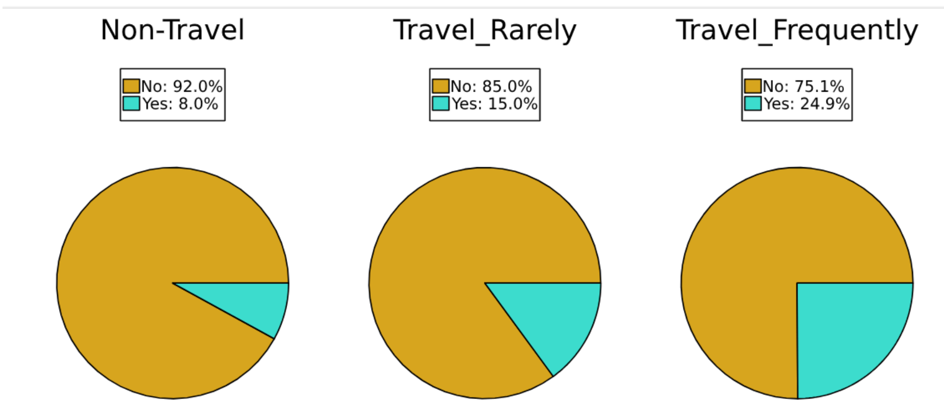




The bar chart shows that out of 501 female employees, 87 left (Yes), and out of 732 male employees, 150 left. The pie charts reflect this in percentages, with 14.8% of female employees and 17.0% of male employees leaving, indicating that attrition is slightly higher among males.

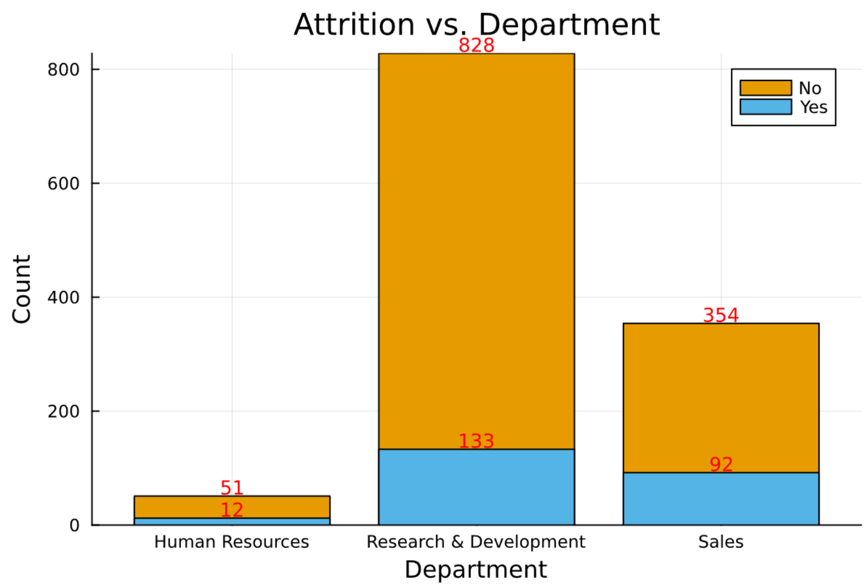
*2.3.4 **BusinessTravel** — Grouped Bar & Pie Chart*





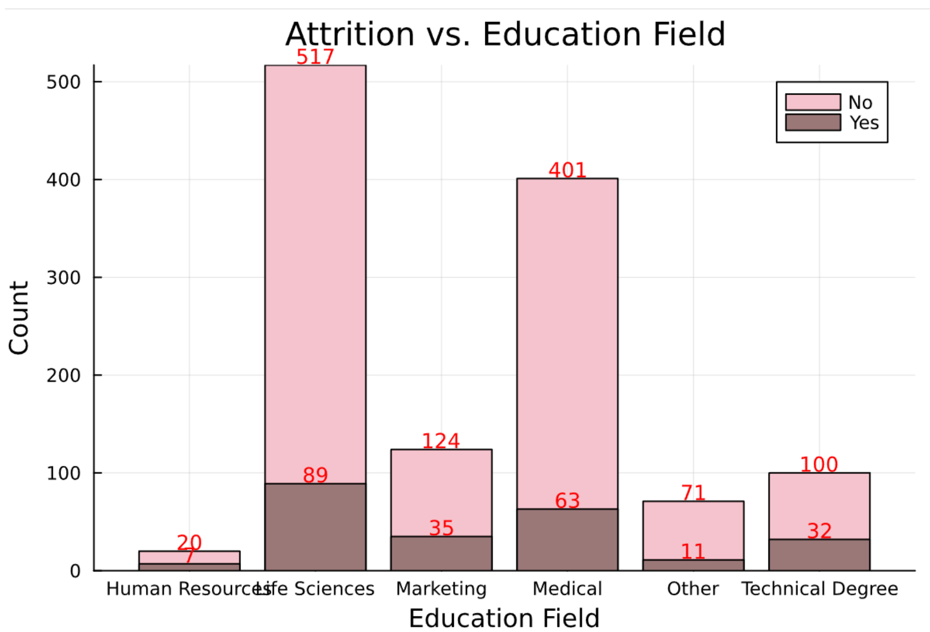
The bar chart and pie charts show the relationship between business travel frequency and employee attrition. Employees who don't travel for business show the lowest attrition rate (8%), those who travel rarely have a higher attrition rate (15%), and employees who travel frequently have the highest attrition rate (24.9%).

2.3.5 ***Department*** *— Grouped Bar Chart*



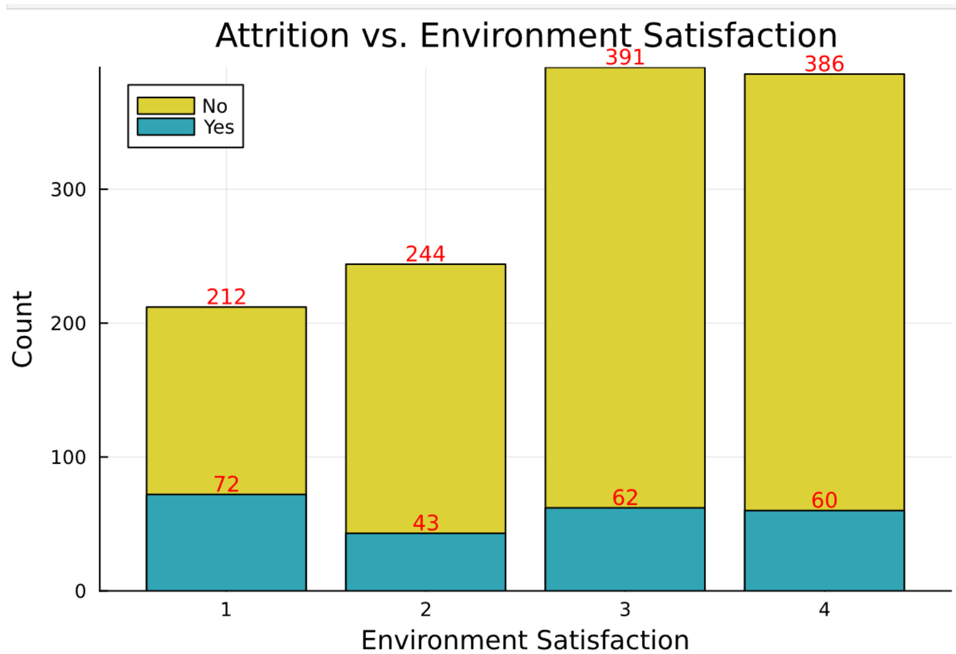
This indicates that the Sales department has the highest rate of employees leaving, followed by Human Resources, and then Research & Development.

2.3.6 ***EducationField*** *— Grouped Bar Chart*



The bar chart shows the number of individuals grouped by their field of education, with a comparison between those who stayed with the company (No attrition) and those who left (Yes attrition). The Life Sciences and Medical fields have higher numbers of individuals who stayed, while the other fields have a more balanced distribution between those who stayed and those who left.

2.3.7 ***Environment Satisfaction*** *— Grouped Bar Chart*



The bar chart indicates that employees with low environmental satisfaction have the highest attrition rate, and as satisfaction increases, attrition rates decrease, suggesting that a more satisfying work environment might lead to lower attrition.

3. **Numerical Data Standardization & Categorical Data Encoding**

To fit the model better, we did data standardization and categorical data encoding(changing categorical variables to numerically coded multi-class type) to make the dataset more suitable and effective for analytical modeling.

3.1 Data Standardization of Numerical Variables

3.1.1 Numerical Data Summary

Row	variable	mean	std	min	median	max	nmissing	eltype
	Symbol	Float64	Float64	Int64	Float64	Int64	Int64	DataType
1	DailyRate	802.486	403.509	102	802.0	1499	0	Int64
2	DistanceFromHome	9.19252	8.10686	1	7.0	29	0	Int64
3	HourlyRate	65.8912	20.3294	30	66.0	100	0	Int64
4	MonthlyIncome	6502.93	4707.96	1009	4919.0	19999	0	Int64
5	MonthlyRate	14313.1	7117.79	2094	14235.5	26999	0	Int64
6	NumCompaniesWorked	2.6932	2.49801	0	2.0	9	0	Int64
7	PercentSalaryHike	15.2095	3.65994	11	14.0	25	0	Int64
8	TotalWorkingYears	11.2796	7.78078	0	10.0	40	0	Int64
9	TrainingTimesLastYear	2.79932	1.28927	0	3.0	6	0	Int64
10	YearsAtCompany	7.00816	6.12653	0	5.0	40	0	Int64
11	YearsInCurrentRole	4.22925	3.62314	0	3.0	18	0	Int64
12	YearsSinceLastPromotion	2.18776	3.22243	0	1.0	15	0	Int64
13	YearsWithCurrManager	4.12313	3.56814	0	3.0	17	0	Int64

data summary of original numerical data

By looking at the data summary of all the numerical variables in our original data, there are a variety of differences in the mean and standard deviations. Applying the standardization helps to scale all data into normalization scales to avoid biased results and numerical instabilities due to different scalings.

3.1.2 Z-score Transformation

```
# use Zscore transform to standarize
dt = StatsBase.fit(ZScoreTransform, numerical_matrix_float; dims=1)
input_standardized = StatsBase.transform!(dt, numerical_matrix_float)

# Round the standardized values to two decimal places
input_standardized = round.(input_standardized, digits=2)
```

Z-score transformation for standardization

To standardize the data, we select all the numerical variables(except variables ordered in numbered levels), change them into a matrix, and use the Z-score transformation method to transform all numerical data into new values of 2 decimals. From the standardized data summary, all averages changed to 0 and standard deviations to 1 for each of the variables, which resulting a normalized feature. Here is our numerical data and summary after standardization.

Row	variable	mean	std	min	median	max	nmissing	eltype
	Symbol	Float64	Float64	Float64	Float64	Float64	Int64	DataType
1	DailyRate	6.80272e-5	0.999839	-1.74	0.0	1.73	0	Float64
2	DistanceFromHome	2.04082e-5	0.999998	-1.01	-0.27	2.44	0	Float64
3	HourlyRate	-0.00014966	1.00027	-1.77	0.01	1.68	0	Float64
4	MonthlyIncome	-0.000170068	1.00004	-1.17	-0.335	2.87	0	Float64
5	MonthlyRate	-4.08163e-5	0.999885	-1.72	-0.01	1.78	0	Float64
6	NumCompaniesWorked	-0.00272109	0.999204	-1.08	-0.28	2.52	0	Float64
7	PercentSalaryHike	0.000557823	1.00063	-1.15	-0.33	2.68	0	Float64
8	TotalWorkingYears	0.000585034	0.999961	-1.45	-0.16	3.69	0	Float64
9	TrainingTimesLastYear	0.00130612	1.00048	-2.17	0.16	2.48	0	Float64
10	YearsAtCompany	0.000598639	0.999726	-1.14	-0.33	5.39	0	Float64
11	YearsInCurrentRole	-0.00214966	1.00068	-1.17	-0.34	3.8	0	Float64
12	YearsSinceLastPromotion	-0.00157823	0.999759	-0.68	-0.37	3.98	0	Float64
13	YearsWithCurrManager	-0.000170068	1.003	-1.16	-0.31	3.61	0	Float64

numerical data summary after standardization

Row	DailyRate	DistanceFromHome	HourlyRate	MonthlyIncome	MonthlyRate	NumCompaniesWorked	PercentSalaryHike	TotalWorkingYears	TrainingTimesLastYear	YearsAtCompany	YearsInCurrentRole	YearsSinceLastPromotion	YearsWithCurrManager
	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64
1	0.74	-1.01	1.38	-0.11	0.73	2.12	-1.15	-0.42	-2.17	-0.16	-0.06	-0.68	0.25
2	-1.3	-0.15	-0.24	-0.29	1.49	-0.68	2.13	-0.16	0.16	0.49	0.76	-0.37	0.81
3	1.41	-0.89	1.28	-0.94	-1.67	1.32	-0.06	-0.55	0.16	-1.14	-1.17	-0.68	-1.16
4	1.46	-0.76	-0.49	-0.76	1.24	-0.68	-1.15	-0.42	0.16	0.16	0.76	0.25	-1.16
5	-0.52	-0.89	-1.27	-0.64	0.33	2.52	-0.88	-0.68	0.16	-0.82	-0.62	-0.06	-0.6

numerical data after standardization

Last, we combine the standardized numerical data and the original categorical data into a new data frame to perform categorical data encoding.

3.2 Categorical Data Encoding

On the other hand, we use the “OneHotEncoder” machine to encode all categorical variables that are not ordered in numbers to perform the Lasso and Classification methods.

```julia
# change "Texual to Multiclass" for all categorical variables
update_IBM = coerce(combined_IBM, Dict(:BusinessTravel => Multiclass, :Department => Multiclass, :EducationField => Multiclass,
                                       :Gender => Multiclass, :JobRole => Multiclass, :MaritalStatus => Multiclass, :OverTime => Multiclass))
MLJ.schema(update_IBM)
```

names	scitypes	types
DailyRate	Continuous	Float64
DistanceFromHome	Continuous	Float64
HourlyRate	Continuous	Float64
MonthlyIncome	Continuous	Float64
MonthlyRate	Continuous	Float64
NumCompaniesWorked	Continuous	Float64
PercentSalaryHike	Continuous	Float64
TotalWorkingYears	Continuous	Float64
TrainingTimesLastYear	Continuous	Float64
YearsAtCompany	Continuous	Float64
YearsInCurrentRole	Continuous	Float64
YearsSinceLastPromotion	Continuous	Float64
YearsWithCurrManager	Continuous	Float64
Age	Count	Int64
Attrition	Textual	String
BusinessTravel	Multiclass{3}	CategoricalValue{String, UInt32}
⋮	⋮	⋮

change all categorical variables to Multiclass-type

```julia
# Encoding categorical variables
mach = machine(OneHotEncoder(), update_IBM) |> fit!
update_IBM = MLJ.transform(mach)
```

encode categorical data using “OneHotEncoder”

Some of our categorical variables have sci types of “Textual” with multiple categories, such as “BusinessTravel” which includes three levels: non-travel, rarely, and frequently. To use “OneHotEncoder”, we need to change the variable to sci type of “Multiclass{numbers of categories}” first. Then we can use the “OneHotEncoder” machine to encode all categorical data by classifying the categories in each variable as a new column with “0” being “No Attrition” and “1” being “Yes Attrition”. This process does not mutate the numerical variables.

BusinessTravel__Non-Travel	BusinessTravel__Travel_Frequently	BusinessTravel__Travel_Rarely	Department__Human Resources	Department__Research & Development	Department__Sales	Education	EducationField__Human Resources	EducationField__Life Sciences
Float64	Float64	Float64	Float64	Float64	Float64	Int64	Float64	Float64
0.0	0.0	1.0	0.0	0.0	1.0	2	0.0	1.0
0.0	1.0	0.0	0.0	1.0	0.0	1	0.0	1.0
0.0	0.0	1.0	0.0	1.0	0.0	2	0.0	0.0
0.0	1.0	0.0	0.0	1.0	0.0	4	0.0	1.0
0.0	0.0	1.0	0.0	1.0	0.0	1	0.0	0.0

encoded categorical data example

4. Lasso Variable Selection

We decided to perform Lasso Variable Selection due to too many variables in our data. Therefore, we want to eliminate the independent variables to only the ones that are strongly correlated with employee attrition. The package here we are using is the GLMNet package.

4.1 Find the Best Lambda by Cross-Validation

```
# Specify the proportion of data to use for training
train_proportion = 0.7
# Split the data into train and test sets
(train_data, test_data) = splitobs(shuffleobs(update_IBM), at = train_proportion)

# predict and response variables of train data
X_train = Matrix(select(train_data, Not([:Attrition])))
y_train = convert(Vector, train_data.Attrition)

# predict and response variables of test data
X_test = Matrix(select(test_data, Not([:Attrition])))
y_test = convert(Vector, test_data.Attrition);
```

split data and separate predict and response variables

Splitting our standardized and encoded data first into 70% training and 30% testing subset. Then for both training and testing data, we change our 51 predictors into an absolute matrix format called X train & test, and the response variable “Attrition” into a vector called y train & test.

```
# find the best value of λ by cross-validation
cv = glmnetcv(X_train, y_train)
```

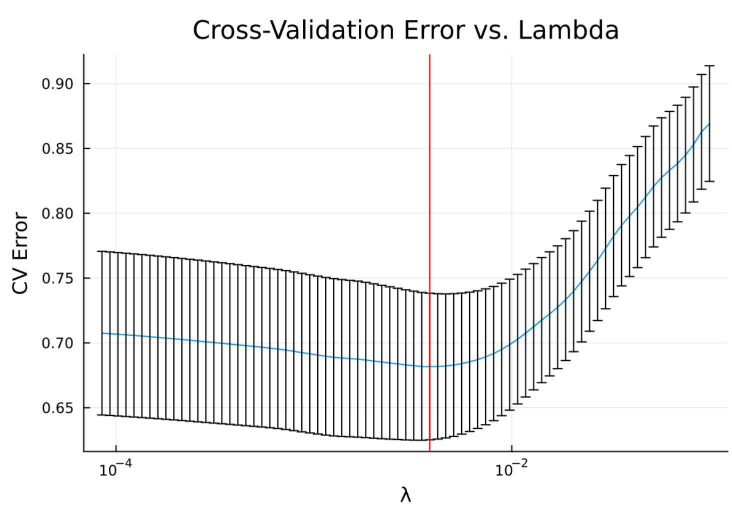
```
Logistic GLMNet Cross Validation
77 models for 51 predictors in 10 folds
Best λ 0.004 (mean loss 0.682, std 0.057)
```

```
# best value of lambda
best_lambda = cv.lambda[argmin(cv.meanloss)]
```

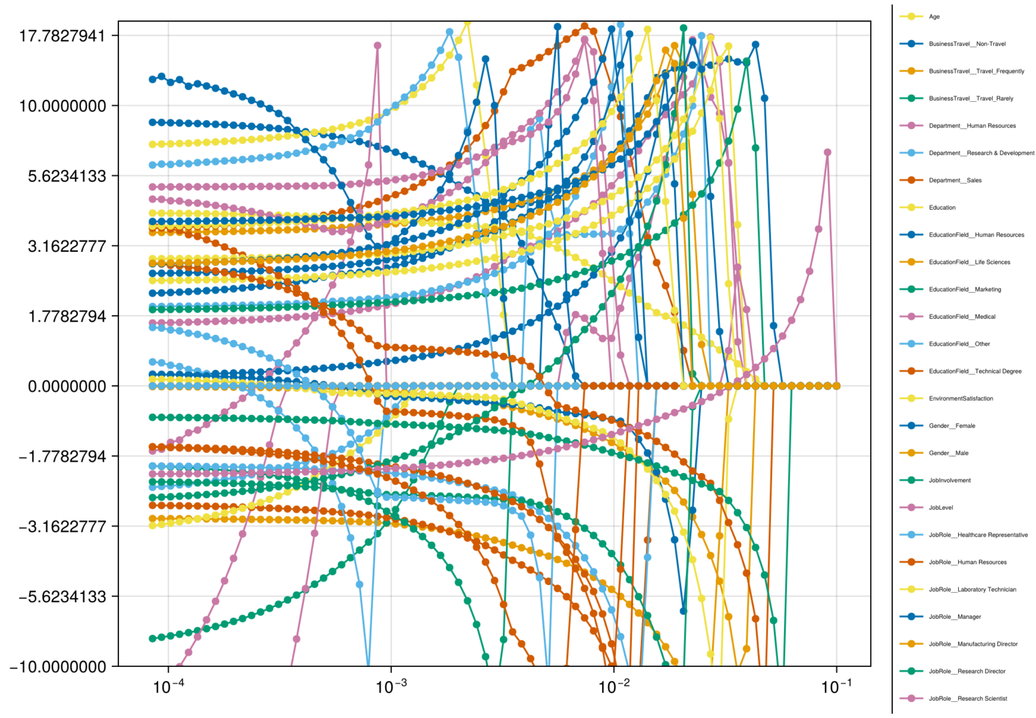
```
0.003849211258865363
```

find the best lambda with minimum cv error

We use the glmnetcv function in the GLMNet package to perform the logistic 10-fold cross-validation, which by passing the predictor matrix and the response vector we created above will result in the best lambda that minimizes the cross-validation error. After fitting 77 different variable models, we find the best value of λ to be 0.0038.



cv error vs. log of lambda



4.2 Best Model

Best Model

```
# best model with the lowest lambda
fitted = glmnet(X_train, y_train, lambda = [best_lambda])
```

```
Logistic GLMNet Solution Path (1 solutions for 51 predictors in 110 passes):
──────────────────────────────
      df    pct_dev           λ
──────────────────────────────
[1]   35   0.312803  0.00384921
──────────────────────────────
```

best model result

From our lambda 0.0038 which minimizes the cross-validation error, we find the best model has 35 correlated independent variables.

```
# DataFrame with all variables with corresponding coefficients
cv_df = DataFrame(predictors = predictors,
                  coef = GLMNet.coef(cv))

# Filter DataFrame to select only columns where coef != 0.0
filter!(row -> row.coef != 0.0, cv_df)

@show cv_df.predictors
```

```
cv_df.predictors = ["DailyRate", "DistanceFromHome", "NumCompaniesWorked", "TotalWorkingYears", "TrainingTimesLastYear", "YearsInCurrentRole", "YearsSinceLastPromotion",
"YearsWithCurrManager", "Age", "BusinessTravel__Non-Travel", "BusinessTravel__Travel_Frequently", "EducationField__Human Resources", "EducationField__Marketing", "Educatio
nField__Medical", "EducationField__Other", "EducationField__Technical Degree", "EnvironmentSatisfaction", "Gender__Female", "Gender__Male", "JobInvolvement", "JobRole__Hum
an Resources", "JobRole__Laboratory Technician", "JobRole__Research Director", "JobRole__Research Scientist", "JobRole__Sales Executive", "JobRole__Sales Representative",
"JobSatisfaction", "MaritalStatus__Divorced", "MaritalStatus__Single", "OverTime__No", "OverTime__Yes", "PerformanceRating", "RelationshipSatisfaction", "StockOptionLeve
l", "WorkLifeBalance"]
```

final selected variables from the best model

After we find the best model, since these 35 variables include those dummy variables in the categorical data, we manually select the variables. There are 22 independent variables selected.

```
# filter selected variables in the best model
final_IBM = select(combined_IBM, [:Attrition, :DailyRate, :DistanceFromHome, :NumCompaniesWorked, :TotalWorkingYears,
                                  :TrainingTimesLastYear, :YearsInCurrentRole, :YearsSinceLastPromotion, :YearsWithCurrManager, :Age, :BusinessTravel,
                                  :EducationField, :EnvironmentSatisfaction, :Gender, :JobInvolvement, :JobRole, :JobSatisfaction,
                                  :MaritalStatus, :OverTime, :PerformanceRating, :RelationshipSatisfaction, :StockOptionLevel, :WorkLifeBalance])
```

final selected data

4.3 Model Performance

```
yhat = ifelse.(probs .>= 0.8, "Yes", "No")
# Accuracy of Train Model
100 * mean(yhat .== y_train)
```

```
85.32555879494656
```

```
# Accuracy of Test Model
yhat_test = ifelse.(GLMNet.predict(fit, X_test) .>= 0.8, "Yes", "No")
100 * mean(yhat_test .== y_test)
```

```
85.26077097505669
```

model accuracy on prediction of train and test

From the prediction made by the best model, we use an 80% cut-off to classify the prediction on attrition. We end up having the accuracy of the train model is 85.33%, while the testing model's accuracy is 85.26%. Overall we can see this model has a pretty high accuracy.

5. Logistic Regression

5.1 **Original Data** - Full Model

We split the training data and testing dataset into 70% and 30%. Using training data to train the logistic regression and using the testing dataset to predict the model and evaluate the accuracy of the model. We input all 31 variables into the logistic regression model to test the performance of the full model.

5.1.1 Regression Summary

	Coef.	Std. Error	z	Pr(>|z|)	Lower 95%	Upper 95%
(Intercept)	-3.53393	16.4979	-0.21	0.8304	-35.8691	28.8013
Age	-0.00766502	0.00908964	-0.84	0.3991	-0.0254804	0.0101503
BusinessTravel: Travel_Frequently	1.05946	0.257327	4.12	<1e-04	0.555112	1.56382
BusinessTravel: Travel_Rarely	0.565585	0.233159	2.43	0.0153	0.108602	1.02257
DailyRate	-8.15487e-5	0.000144668	-0.56	0.5730	-0.000365093	0.000201995
Department: Research & Development	3.69311	16.4746	0.22	0.8226	-28.5966	35.9828
Department: Sales	2.82107	16.4817	0.17	0.8641	-29.4825	35.1246
DistanceFromHome	0.0216475	0.00705052	3.07	0.0021	0.00782875	0.0354663
Education	0.0098396	0.0582332	0.17	0.8658	-0.104295	0.123974
EducationField: Life Sciences	-0.74801	0.519115	-1.44	0.1496	-1.76546	0.269437
EducationField: Marketing	-0.529457	0.55178	-0.96	0.3373	-1.61093	0.552012
EducationField: Medical	-0.816324	0.51797	-1.58	0.1150	-1.83153	0.198878
EducationField: Other	-1.02763	0.563558	-1.82	0.0682	-2.13218	0.0769277
EducationField: Technical Degree	-0.368191	0.538569	-0.68	0.4942	-1.42377	0.687384
EnvironmentSatisfaction	-0.228288	0.054057	-4.22	<1e-04	-0.334238	-0.122339
Gender: Male	0.187653	0.120217	1.56	0.1185	-0.0479685	0.423274
HourlyRate	3.52376e-5	0.00293766	0.01	0.9904	-0.00572248	0.00579295
JobInvolvement	-0.28713	0.0813221	-3.53	0.0004	-0.446518	-0.127741
JobLevel	0.0658274	0.211105	0.31	0.7552	-0.347931	0.479586
JobRole: Human Resources	4.12415	16.4758	0.25	0.8023	-28.1678	36.4161
JobRole: Laboratory Technician	0.640128	0.283399	2.26	0.0239	0.0846764	1.19558
JobRole: Manager	-0.145454	0.503047	-0.29	0.7725	-1.13141	0.8405
JobRole: Manufacturing Director	0.0501855	0.295518	0.17	0.8651	-0.529019	0.62939
JobRole: Research Director	-1.34068	0.693355	-1.93	0.0532	-2.69963	0.0182737
JobRole: Research Scientist	-0.0746479	0.296346	-0.25	0.8011	-0.655476	0.50618
JobRole: Sales Executive	1.14345	0.750189	1.52	0.1275	-0.32689	2.6138
JobRole: Sales Representative	1.79254	0.794939	2.25	0.0241	0.234488	3.35059
JobSatisfaction	-0.184959	0.05289	-3.50	0.0005	-0.288621	-0.0812962
MaritalStatus: Married	0.304938	0.177329	1.72	0.0855	-0.0426196	0.652497
MaritalStatus: Single	0.751413	0.232187	3.24	0.0012	0.296336	1.20649
MonthlyIncome	1.24332e-5	5.33969e-5	0.23	0.8159	-9.22229e-5	0.000117089
MonthlyRate	1.48694e-6	8.29464e-6	0.18	0.8577	-1.47702e-5	1.77441e-5
NumCompaniesWorked	0.0804375	0.0263556	3.05	0.0023	0.0287814	0.132093
OverTime: Yes	1.12684	0.126143	8.93	<1e-18	0.879602	1.37407
PercentSalaryHike	-0.0224583	0.0261024	-0.86	0.3896	-0.0736181	0.0287015
PerformanceRating	0.388423	0.263478	1.47	0.1404	-0.127984	0.904831
RelationshipSatisfaction	-0.14041	0.0546056	-2.57	0.0101	-0.247435	-0.0333849
StockOptionLevel	-0.0907072	0.104157	-0.87	0.3838	-0.294851	0.113436
TotalWorkingYears	-0.0397835	0.0189585	-2.10	0.0359	-0.0769414	-0.00262557
TrainingTimesLastYear	-0.124108	0.0482711	-2.57	0.0101	-0.218718	-0.0294984
WorkLifeBalance	-0.184011	0.083802	-2.20	0.0281	-0.348259	-0.0197617
YearsAtCompany	0.0598197	0.0236961	2.52	0.0116	0.0133763	0.106263
YearsInCurrentRole	-0.088118	0.0292686	-3.01	0.0026	-0.145483	-0.0307526
YearsSinceLastPromotion	0.0807535	0.025498	3.17	0.0015	0.0307784	0.130729
YearsWithCurrManager	-0.0510224	0.0292024	-1.75	0.0806	-0.108258	0.0062133

5.1.2 Model Performance on Prediction

```
# Converting probability score to classes with cut of score of 0.8
prediction_class = [if i < 0.8 0 else 1 end for i in prediction]
```

A threshold of 0.8 was selected based on the original attrition rate distribution, where the proportion of "no" responses amounted to 83%.

```
# Accuracy Score
accuracy_score = GLM.mean(prediction_df.correctly_classified)*100
```

```
84.58049886621315
```

model accuracy of training data predict testing data

Row	Class	Yes	No
	String	Int64	Int64
1	Yes	8	0
2	No	68	365

confusion matrix of the model performance

The confusion matrix indicates that the comprehensive model performs satisfactorily; however, it is burdened by an excessive number of variables, leading to considerable time and space consumption during execution. Consequently, we aim to refine the model by selecting statistically significant variables via LASSO. This approach is intended to preserve the model's accuracy while enhancing its computational efficiency, thereby optimizing it for more effective deployment in data analysis projects.

5.2 **Selected Data from Lasso** - Significant Model

After using Lasso variable selection, we selected **22** statistically significant independent variables to estimate the significant model.

5.2.1 Regression Summary

	Coef.	Std. Error	z	Pr(>|z|)	Lower 95%	Upper 95%
(Intercept)	0.264329	0.882192	0.30	0.7645	-1.46474	1.99339
Age	-0.0103409	0.00880738	-1.17	0.2403	-0.0276031	0.00692121
BusinessTravel: Travel_Frequently	1.02616	0.249312	4.12	<1e-04	0.537517	1.5148
BusinessTravel: Travel_Rarely	0.535678	0.226665	2.36	0.0181	0.0914223	0.979934
DailyRate	-0.0468263	0.0571063	-0.82	0.4122	-0.158753	0.0650999
DistanceFromHome	0.170194	0.0560024	3.04	0.0024	0.0604308	0.279956
EducationField: Life Sciences	-0.589876	0.484148	-1.22	0.2231	-1.53879	0.359037
EducationField: Marketing	-0.385372	0.516269	-0.75	0.4554	-1.39724	0.626496
EducationField: Medical	-0.64835	0.483054	-1.34	0.1795	-1.59512	0.298418
EducationField: Other	-0.91972	0.533788	-1.72	0.0849	-1.96593	0.126485
EducationField: Technical Degree	-0.222354	0.503986	-0.44	0.6591	-1.21015	0.76544
EnvironmentSatisfaction	-0.217905	0.0529274	-4.12	<1e-04	-0.321641	-0.114169
Gender: Male	0.176166	0.118338	1.49	0.1366	-0.0557727	0.408105
JobInvolvement	-0.289477	0.0793625	-3.65	0.0003	-0.445025	-0.133929
JobRole: Human Resources	0.471017	0.387847	1.21	0.2246	-0.289149	1.23118
JobRole: Laboratory Technician	0.554584	0.24717	2.24	0.0248	0.0701405	1.03903
JobRole: Manager	-0.274375	0.373624	-0.73	0.4627	-1.00666	0.457914
JobRole: Manufacturing Director	0.040614	0.290257	0.14	0.8887	-0.52828	0.609508
JobRole: Research Director	-0.974531	0.561697	-1.73	0.0827	-2.07544	0.126374
JobRole: Research Scientist	-0.1515	0.259546	-0.58	0.5594	-0.660202	0.357201
JobRole: Sales Executive	0.2698	0.254611	1.06	0.2893	-0.229228	0.768829
JobRole: Sales Representative	0.802424	0.305677	2.63	0.0087	0.203308	1.40154
JobSatisfaction	-0.172782	0.0515525	-3.35	0.0008	-0.273823	-0.0717405
MaritalStatus: Married	0.27572	0.173195	1.59	0.1114	-0.0637367	0.615176
MaritalStatus: Single	0.71254	0.226326	3.15	0.0016	0.268949	1.15613
NumCompaniesWorked	0.158296	0.0637469	2.48	0.0130	0.0333539	0.283237
OverTime: Yes	1.0947	0.123512	8.86	<1e-18	0.85262	1.33678
PerformanceRating	0.173326	0.156177	1.11	0.2671	-0.132775	0.479427
RelationshipSatisfaction	-0.128798	0.0535377	-2.41	0.0161	-0.23373	-0.0238655
StockOptionLevel	-0.0849464	0.101397	-0.84	0.4022	-0.283681	0.113788
TotalWorkingYears	-0.097953	0.117352	-0.83	0.4039	-0.32796	0.132054
TrainingTimesLastYear	-0.147416	0.0610392	-2.42	0.0157	-0.26705	-0.0277812
WorkLifeBalance	-0.172372	0.0821055	-2.10	0.0358	-0.333296	-0.0114485
YearsInCurrentRole	-0.221684	0.0990804	-2.24	0.0253	-0.415878	-0.0274899
YearsSinceLastPromotion	0.3144	0.0789169	3.98	<1e-04	0.159726	0.469074
YearsWithCurrManager	-0.0785268	0.0939654	-0.84	0.4033	-0.262696	0.105642

5.2.2 Model Performance on Prediction

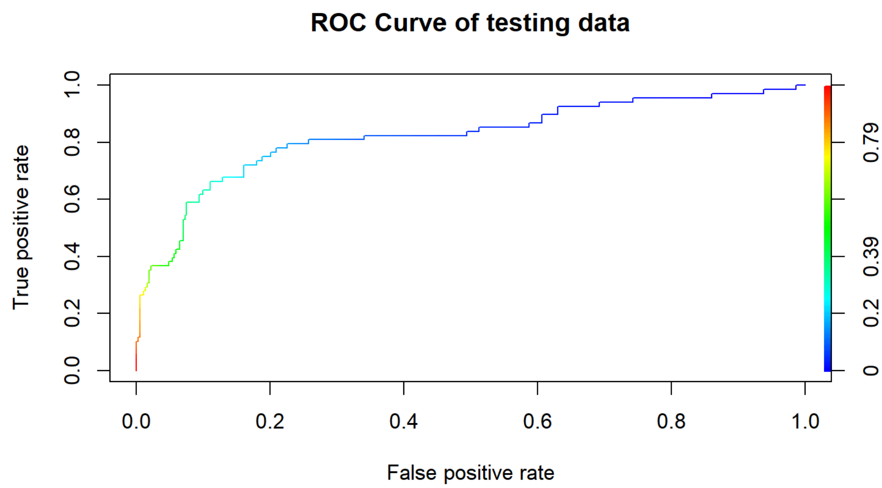
```
# Accuracy Score
accuracy_score = GLM.mean(prediction_df.correctly_classified)*100
```

```
84.12698412698413
```

model accuracy of training data predict testing data

Row	Class	Yes	No
	String	Int64	Int64
1	Yes	6	0
2	No	70	365

confusion matrix of the model performance



ROC Curve of the model performance

The performance of the significant model, as measured by the accuracy score, closely mirrors that of the full model, despite a reduction in the number of variables from 31 to 22. Given the comparable accuracy and improved efficiency, we advocate for the adoption of a significant model for predicting attrition rates. Additionally, we recommend a focused analysis of the impact of the variables retained in the significant model to further understand their contributions to the model's predictive capabilities.

6. Classification (*KNNClassifier, LDA, NeuralNetworkClassifier, MultinomialClassifier)*

We split the data and fit each model using our best-selected data and evaluate the **accuracy** which measures the percentage of the correct predictions, **precision** which measures the percentage of true positive instances out of the total instances predicted as positive, **recal**l which measures the true positive instances out of the total actual positive instances, and **F1** which provide a balanced measure of precision and recall.

6.1 Confusion Matrix for Each Model

```
# ConfusionMatrix for each model
mat[1]  # KNNClassifier
```

	Ground Truth	
Predicted	Yes	No
Yes	8	4
No	64	365

```
mat[2]  # LDA
```

	Ground Truth	
Predicted	Yes	No
Yes	55	72
No	17	297

```
mat[3]  # NeuralNetworkClassifier
```

	Ground Truth	
Predicted	Yes	No
Yes	18	3
No	54	366

```
mat[4]  # MultinomialClassifier
```

	Ground Truth	
Predicted	Yes	No
Yes	24	8
No	48	361

From the matrices that result in the model performance of the KNNClassifier, LDA, NeuralNetworkClassifier, and MultinomialClassifier method, the data results are imbalanced.

6.2 Model Performance

Row	Model	Accuracy	Precision	Recall	F1
	DataType	Float64	Float64	Float64	Float64
1	KNNClassifier	0.845805	0.758741	0.550136	0.552632
2	LDA	0.798186	0.689465	0.784383	0.711228
3	NeuralNetworkClassifier{Short, typeof(softmax), Adam, typeof(crossentropy)}	0.870748	0.864286	0.620935	0.657427
4	MultinomialClassifier	0.873016	0.81632	0.655827	0.69478

Based on our attrition distribution, we have identified an imbalance between the 'Yes' and 'No' categories. Therefore, we aim to find the model with the highest accuracy while maintaining a fairly good F1 score.

Looking at each model's performance, the KNN Classifier has an overall high accuracy with the lowest F1 score, indicating that the model is performing well on the majority class but struggling with the minority class. This matches our imbalanced datasets, where the model may be biased towards the majority class. The LDA model has the lowest accuracy but the highest F1 score, indicating that the model is performing well on the minority class but misclassifying a significant portion of the majority class instances. The Neural Network Classifier and the Multinomial Classifier have very similar performances with an overall balanced accuracy and F1 score, which maintains a balance between the overall correctness of the prediction and optimizing the correct classification.

Comparing the Neural Network Classifier and the Multinomial Classifier models' performance, the Multinomial Classifier has a better performance. Both the accuracy and F1 score of the Multinomial are higher than the Neural Network Classifier. This result matches the characteristics of our data since most of our categorical variables are multiclassed, which have subcategories with 2 or more.

7. Conclusion

Throughout this project, we encountered numerous challenges, with variable selection posing the most significant obstacle. Our dataset comprises 35 variables, predominantly categorical in nature, necessitating format conversion through one-hot encoding. Additionally, numerical data required standardization to ensure uniformity prior to analysis. The integration of these preprocessed variables into a Lasso regression model for the identification of statistically significant predictors represented the project's most complex aspect. Employing Lasso regularization within the context of logistic regression was particularly challenging, demanding a nuanced understanding of both the mathematical principles involved and the practical considerations of their application in data analysis.

In this project, we uncovered several key insights that can assist the Human Resources (HR) department in mitigating employee turnover and retaining valuable staff. These insights revolve around factors such as the amount of overtime work, job satisfaction, job involvement, years since the last promotion, tenure in the current role, and the extent of business travel. Below is a detailed exploration of how each factor influences attrition rates, intended to serve as a comprehensive conclusion to our analysis:

- *Overtime Work*: Our findings indicate that excessive overtime can significantly elevate employee turnover rates. Employees subjected to prolonged hours of work beyond their regular schedule are more likely to experience burnout and decreased work-life balance. A policy to monitor and limit overtime could foster a healthier work environment and reduce attrition.
- *Job Satisfaction and Involvement*: Job satisfaction emerged as a critical determinant of employee retention. Satisfaction levels correlate directly with employees' sense of value and their engagement with the organization. High job satisfaction enhances the loyalty of employees. Job involvement is linked to a sense of purpose and the perception that one's work is meaningful. Cultivating an environment that promotes involvement can lead to increased retention by making employees feel integral to the company's success.
- *Years Since Last Promotion*: The duration since an employee's last promotion plays a significant role in their decision to stay with or leave the company. Longer intervals without recognition or advancement can lead to frustration and diminished motivation. Streamlining promotion and recognition processes to acknowledge and reward deserving employees timely may help in curbing attrition rates.
- *Minimizing Business Travel*: Frequent business travel can be a source of stress and dissatisfaction for employees, contributing to higher attrition rates. The physical and emotional toll of regular travel can impact work-life balance adversely. Therefore, optimizing travel schedules and exploring alternatives such as virtual meetings could help in retaining staff.

In conclusion, by addressing these factors, the HR department can implement targeted strategies aimed at reducing employee turnover. The emphasis should be on creating a supportive and fulfilling work environment that encourages staff to remain engaged and committed to the organization.