CAL STATE EAST BAY

Application of Machine Learning in predicting Employee Attrition

SPRING SEMESTER 2023

Submitted to,

Dr. Farnaz Ganjeizadeh
ENGR 693A

Apr 27, 2023

Team members,

Radhika Vijayaraghavan

Chandnee Das

Rishikesh Reddy

AGENDA

- Introduction
- Objective
- Responsibility Matrix
- Executive Summary
- Literature review
- Descriptive Analysis
- Data cleaning
- Exploratory Analysis
- Feature Selection
- Model Building/Comparison
- Model Evaluation
- Future work
- References

PROBLEM BACKGROUND

U.S. employee annual voluntary turnover is likely to jump nearly 20% this year, from 31.9 million employees quitting their jobs to 37.4 million quitting in 2022, according to Gartner, Inc

---

The SHRM Human Capital Benchmarking report found that the average employee turnover rate in 2017 was 18%, and less than 50% of organizations have a succession plan.

Reports say that, if a company's employee turnover rate is more than 15% per year, the company has a high employee turnover rate.

INTRODUCTION

Source

We use a dataset put up by IBM data scientists for our analysis.

Source of data is from Kaggle.com

Goal

Building and assessing predictive models using various ML techniques

Data

- 23434 observations on 37 variables
- Contains data related to employee performance measures and attrition.

OBJECTIVE

The goal is to find how the company's objective factors influence in attrition of employees, and what kind of working environment is most likely to cause attrition.

Provide recommendations to prevent employees from attrition.

To create industry-specific predictive algorithms, use it to check the accuracy and compare it with the accuracy of the existing models

RESPONSIBILITY MATRIX

- *R - Responsible , C - Consulted , I - Informed , A - Accountable*

Tasks	Chandnee	Radhika	Rishikesh
Brainstorming Project Ideas	**A**	**A**	**A**
Literature Review	**R/A**	**C/A**	**R/C**
Data Collection	**R/A**	**R/A**	**R/A**
Coding	**R/A**	**R/A**	**R/A**
Methodology	**R**	**R/A**	**R**
Model Evaluation	**A/I**	**A/I**	**C/I**
EDA	**C/I**	**C/I**	**R/A**
Conclusion	**A**	**A**	**A**

EXECUTIVE SUMMARY

Language used: R

IDE used : JupyterLab

Models built for this research:

1) Logistic Regression using regularization

2) Random Forest

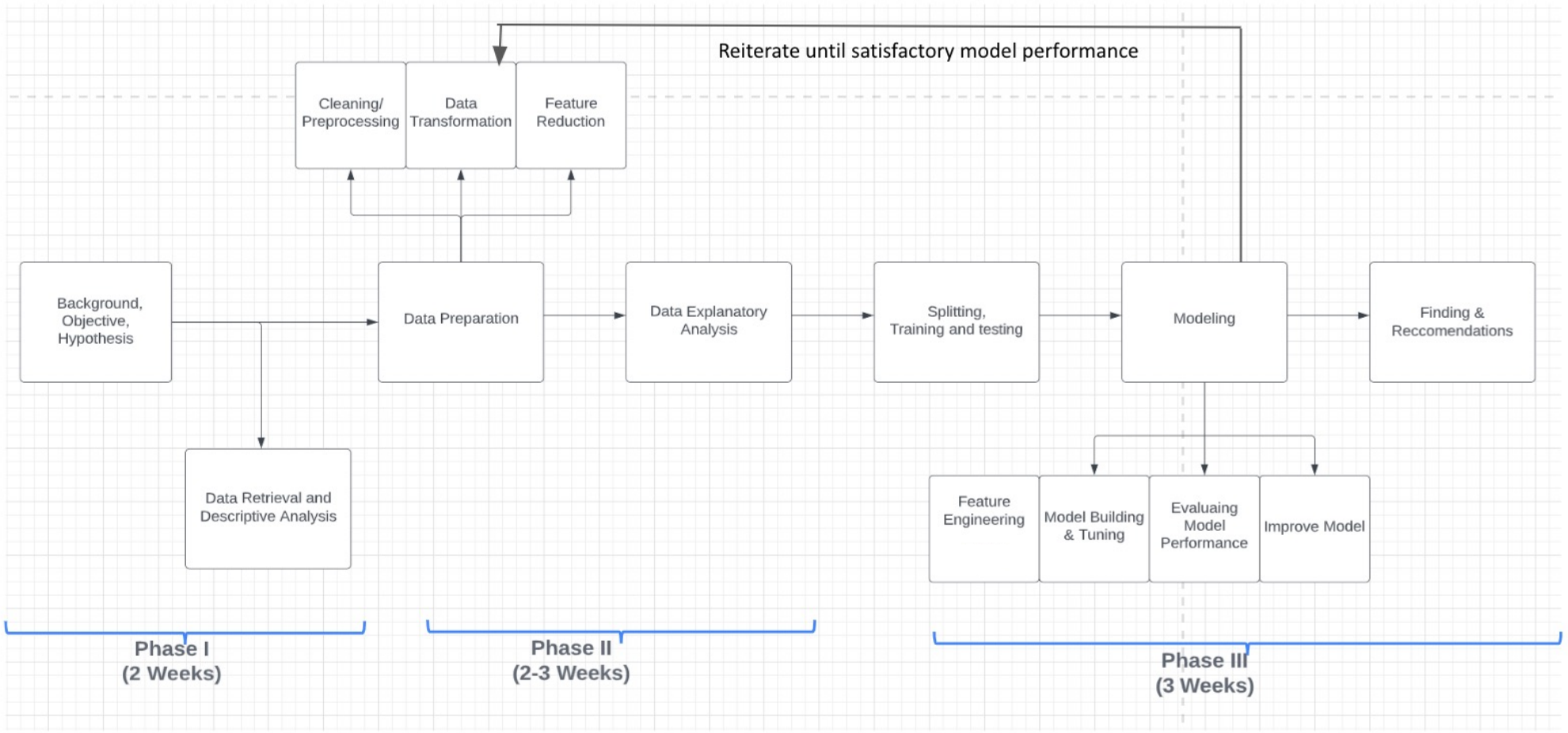
3) Gradient Boosting(xgBoost)

Performance evaluation techniques used:- Accuracy score, roc_auc curve, precision, recall, F1-score

LITERATURE REVIEW

REFERENCE	TOPIC	METHOD	STRENGTH	WEAKNESS
Raman 2019	Predictive Analytics	Correlation analysis, Sentimental Analysis, Word Count Analysis	Good for business school model.	Just specific for one sector area, like the academic sector only. It doesn't apply to other sectors
Mansur, Norshuda 2021	Machine learning for predicting Employee attrition	Classification method, SVM, J48, MLP	Choose one classification method among others to give the better result.	Very few factors used. Not used wide range of predictors to get the perfect response accuracy.
Setiawan 2020	HR analytics: Employee attrition analysis using logistic regression	Logistic regression.	They worked with the improvement of HR for evaluating some factors which makes employee attrition. And showed great visualizations.	Did not use different techniques for getting better accuracy and did not describe which factors are most significant.
Sriram 2019	Factors affecting High Employee Attrition in Manufacturing Firms – A Case Study	ANOVA, Multiple regression	They used ANOVA for predicting employee attrition based on just two main factors.	Did not use wide range of factors to get true result. Did not give a clear picture on the regression result.
Narayana Darapaneni; Raghavendra Naga Turaga; A detailed analysis of AI models..	A Detailed Analysis of AI Models for Predicting Employee Attrition Risk	SVM, Random Forest, Logistic regression, Boosting, Ensemble Average	The robust solution uses xTreme Gradient Boosting which helps to identify the usefulness of each feature in the model.	The models in this paper doesn't adapt well to influencing factors. Imbalance in the data pertaining to each target variable. Needs bigger data sample for higher accuracy.

PROJECT TIMELINE



PREDICTS BINARY OUTCOME

PREDICTS DEPENDENT DATA VARIABLE

DOESN'T REQUIRE NORMALIZING/ SCALING

BASED ON STATISTICAL ANALYSIS METHOD

REDUCE THE COMPLEXITY BY IMPOSING A PENALTY

1. LOGISTIC REGRESSION USING REGULARIZATION

PERFORMS BOOTSTRAPPING AND AGGREGATION

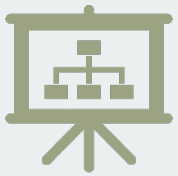
TAKES MAJORITY VOTE FOR FINAL PREDICTION

PREVENTS OVERFITTING OF DATA

II. RANDOM FOREST

USED FOR BOTH CLASSIFICATION & REGRESSION

HANDLES HUGE DATA WITH MANY VARIABLES



ITERATIVE PROCESS CORRECTING EARLIER TREES MISTAKES

III. XGBOOST

PHASE 1, STEP 1 – DESCRIPTIVE ANALYSIS

```
> str(df_cleaned_new)
'data.frame':   5699 obs. of  21 variables:
 $ attrition                : Factor w/ 2 levels "No","Yes": 2 2
 $ business_travel          : Factor w/ 3 levels "Non-Travel","Tr
 $ department               : Factor w/ 3 levels "R&D","HR","Sale
 $ distance_from_home       : int  1 6 6 1 1 6 8 2 2 8 ...
 $ education                : Factor w/ 5 levels "Bachelors","Bel
 $ education_field          : Factor w/ 5 levels "Human Resources
 $ environment_satisfaction : int  2 1 1 4 1 1 3 1 1 3 ...
 $ gender                   : Factor w/ 2 levels "Female","Male":
 $ job_involvement          : int  3 3 3 3 3 2 2 2 2 2 ...
 $ job_role                 : Factor w/ 9 levels "Healthcare Repr
 $ marital_status           : Factor w/ 3 levels "Divorced","Marr
 $ monthly_income           : int  5993 5993 5993 14756 19566 234
 $ over_time                : Factor w/ 2 levels "No","Yes": 2 2
 $ percent_salary_hike      : int  11 11 11 14 11 21 23 23 23 23
 $ relationship_satisfaction : int  1 1 1 3 4 1 4 4 4 4 ...
 $ stock_option_level       : int  0 0 0 3 0 0 1 1 1 1 ...
```

```
> table(df_cleaned_new$attrition)

  No  Yes
4744  955
> perc_attrition_rate <- (sum(df_cleaned_new$attrition== "Yes")
+ /length(df_cleaned_new$attrition))*100
> print(perc_attrition_rate)
[1] 16.75733
>
```

- 5699 employees
- 21 feature variables
- No NaN values
- Turnover rate of 16.75%

PHASE 1, STEP 1 – DESCRIPTIVE ANALYSIS

Overview of Target variable – "Attrition"

Category	job_satisfaction	percent_salary_hike	monthly_income	distance_from_home	years_at_company	years_since_last_promotion
No	2.752308	15.22584	6615.188	8.586998	7.112920	2.205518
Yes	2.586370	15.01748	5512.757	10.069630	6.013926	2.006815

Stayed →

Left the company →

Lower monthly income

Higher # of travel hours

Less Tenured

'Yes' indicates employees who are attrited
'No' indicates not attrited employees

PHASE II – STEP 1
DATA CLEANING

Changed inappropriate values to NAs

- Eg:- “”, “missing”, “na”, ”Test”, ”TEST”, “TESTING”, “Other”

Removed duplicated rows

- 1000 duplicate rows removed

Remove Redundant Columns

- Employee Count
- Over18
- StandardHours

} Constant values

- DailyRate
- HourlyRate
- MonthlyRate

} Poor correlation, deemed unnecessary

Data Type Conversion

Factor

Attrition
BusinessTravel
Department
Education
EducationField
MaritalStatus
Gender
Department
EmployeeSource

Integer

DistanceFromHome
HourlyRate
MonthlyIncome
PercentSalaryHike

Ordinal

PerformanceRating
RelationshipSatisfaction
JobInvolvement

SPECIFIC DATA CLEANING

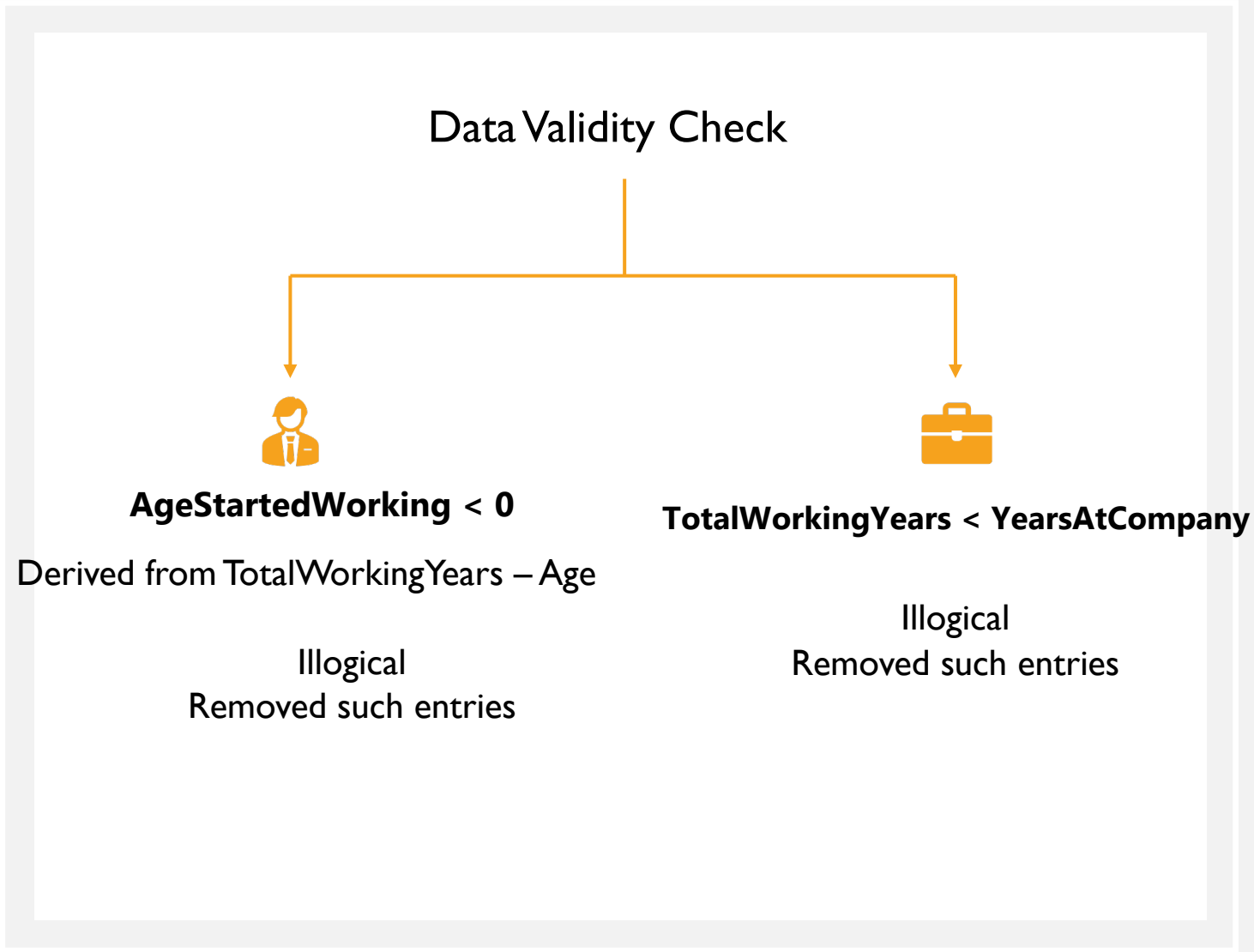
- From the data summary, we see that:
  - Dropped the unnecessary levels
  - Dropped NA values
  - Re-factored levels

```r
# Identify factor columns in df_cleaned
factor_cols <- sapply(df_cleaned, is.factor)

# Remove "Other" and "Test" levels from all factor columns
df_cleaned[factor_cols] <- lapply(df_cleaned[factor_cols],
                                  function(x) droplevels(x,
                                                         exclude = c("Other", "Test", "NA")))

# Convert education_field to character
df_cleaned$education_field <- as.character(df_cleaned$education_field)

# Remove rows containing "NA"
df_cleaned_new <- df_cleaned[!df_cleaned$education_field %in% "NA", ]

# Remove rows with NAs
df_cleaned_new <- drop_na(df_cleaned_new) %>%
  mutate(education_field = as.factor(education_field)) %>%
  select(-c(num_companies_worked, total_working_years, years_with_curr_manager, years_in_current
years_at_company, age_started_working, age, job_level, job_satisfaction, performance_rating))
```

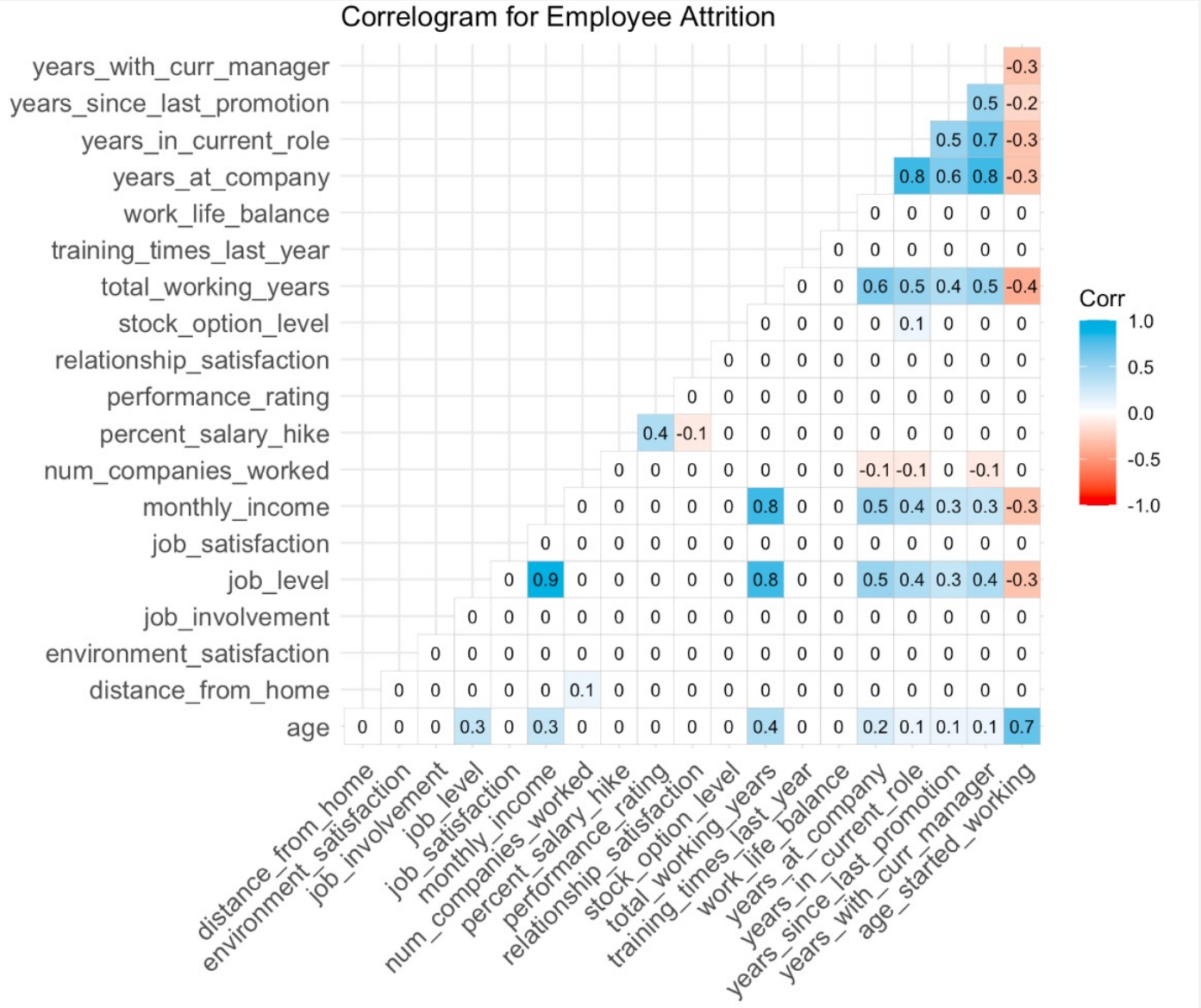
```
                    department
1296                     :    1
Human Resources          : 1019
Research & Development   :15350
Sales                    : 7151
NA's                     :   11
```

```
     gender
1      :    1
2      :    1
Female: 9400
Male   :14120
NA's   :   10
```

SPECIFIC DATA CLEANING

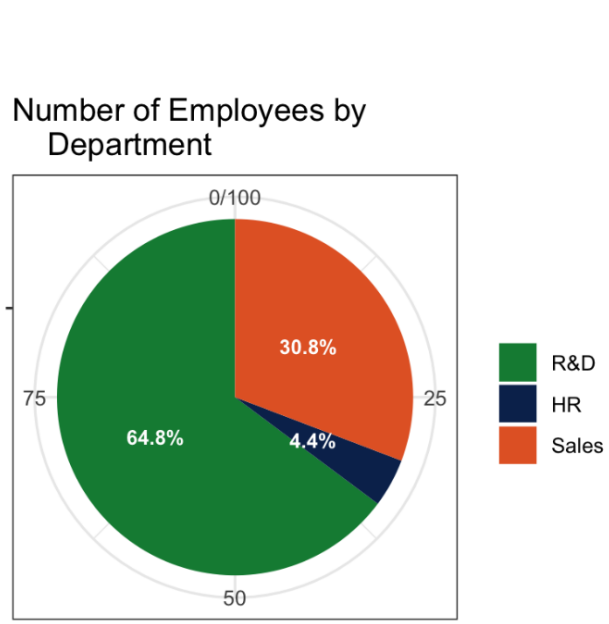


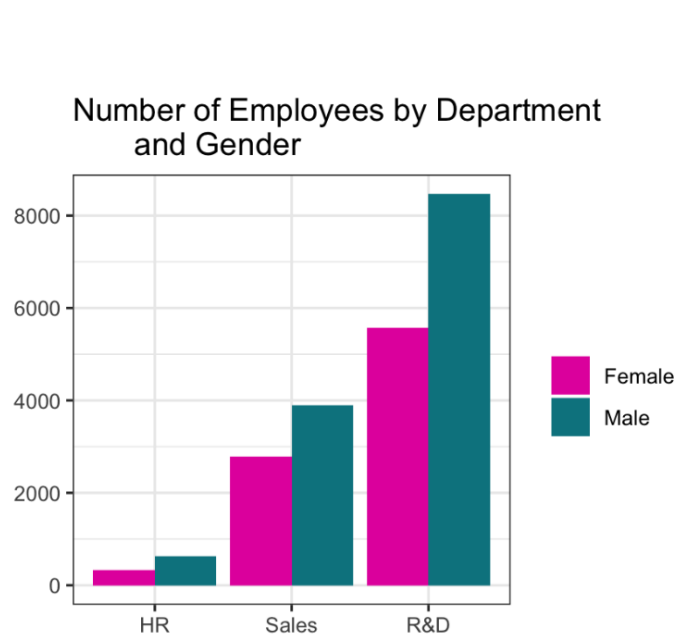
PHASE II – STEP II – EDA

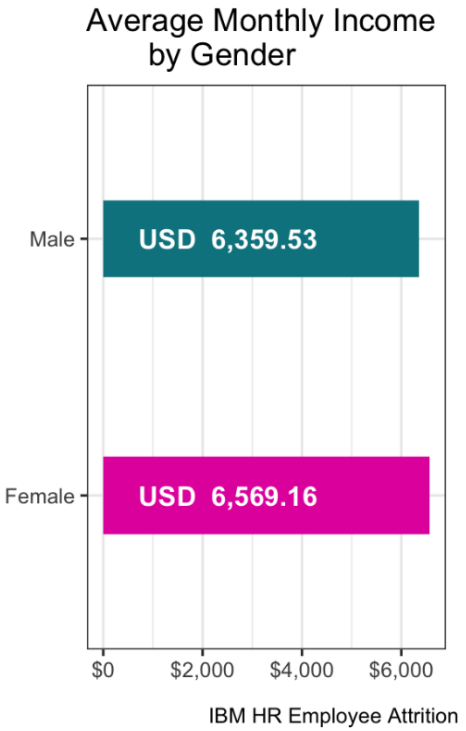


PHASE II – STEP II – EDA

- # of Male & Female employees in R&D are higher than other departments
- Female employees have higher average monthly comparatively

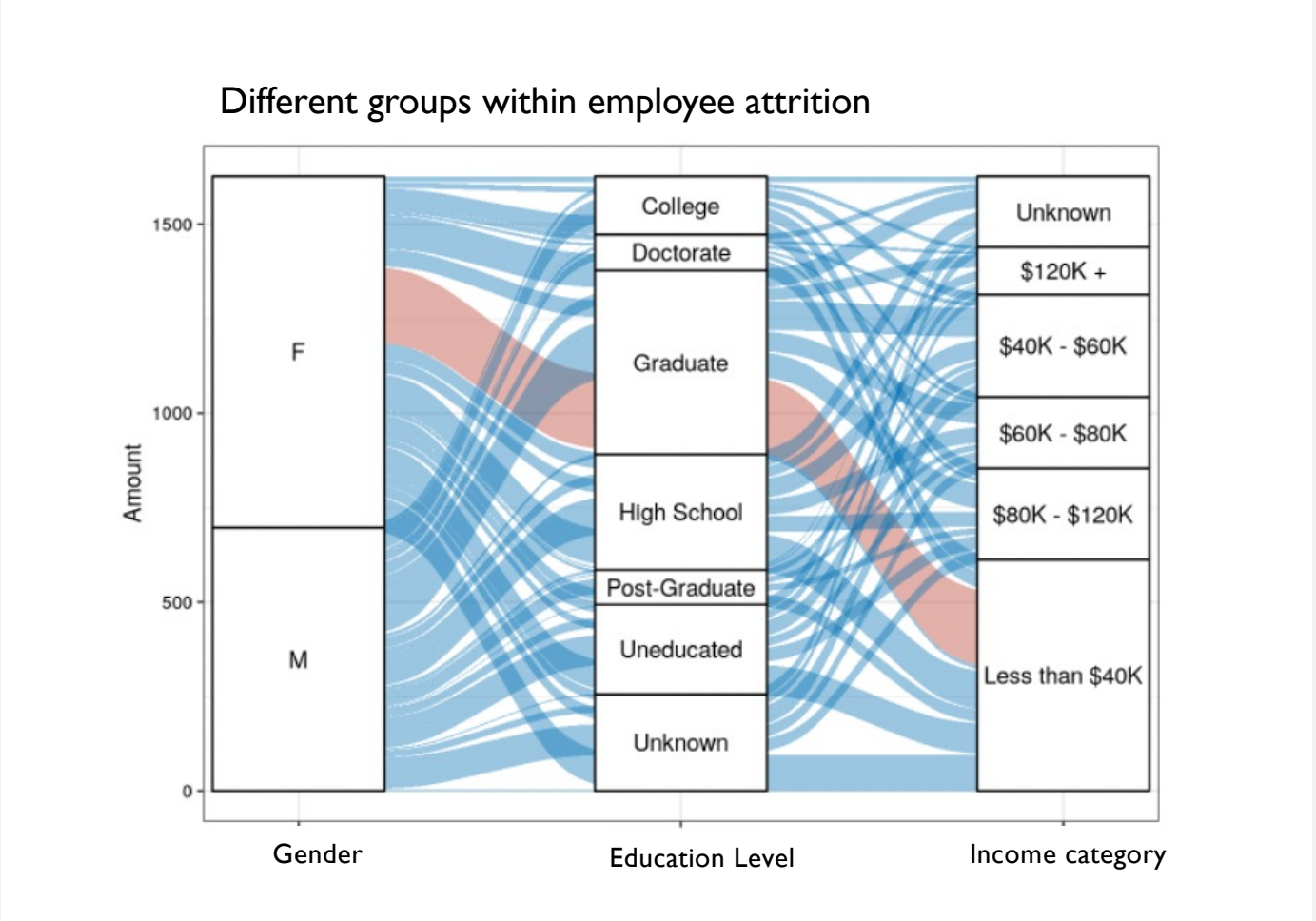






IBM HR Employee Attrition

CLUSTER ANALYSIS



- Shows the clusters groups from each categorical variable from the attrited employees.
- Female graduates with low-income account for the highest amount of attrition

R LIBRARIES USED FOR PROJECT

- Tidymodels
- mlbench
- Keras
- DataExplorer
- Caret
- gmodels

KEY POINTS

- **Total data points after cleaning/pre-processing:** 5699 observations x 21 features
- **% for Training & Testing:**
    - 75% - Training
    - 25% - Testing
- **Technique used for sampling:** Bootstrap sampling
- **Algorithm Type :** Classification
- **Method used for Model evaluation:** Confusion matrix, ROC_AUC, Accuracy

PHASE III – STEP I FEATURE SELECTION

- ❑ Normalize/scale data
- ❑ Removes highly correlated & near zero variance features
- ❑ Assigns dummy variables using one-hot encoding
- ❑ Handles class imbalance

```
Recipe

Inputs:


Training data contained 4274 data points and no missing data.

Operations:

Dummy variables from business_travel, department, education, education_field, gender, job_role,
marita... [trained]
Centering and scaling for distance_from_home, environment_satisfaction, job_involvement,
monthly_... [trained]
Sparse, unbalanced variable filter removed department_HR, education_Ph.D,
job_role_Human.Resources, employ... [trained]
Correlation filter on business_travel_Travel_Frequently [trained]
```

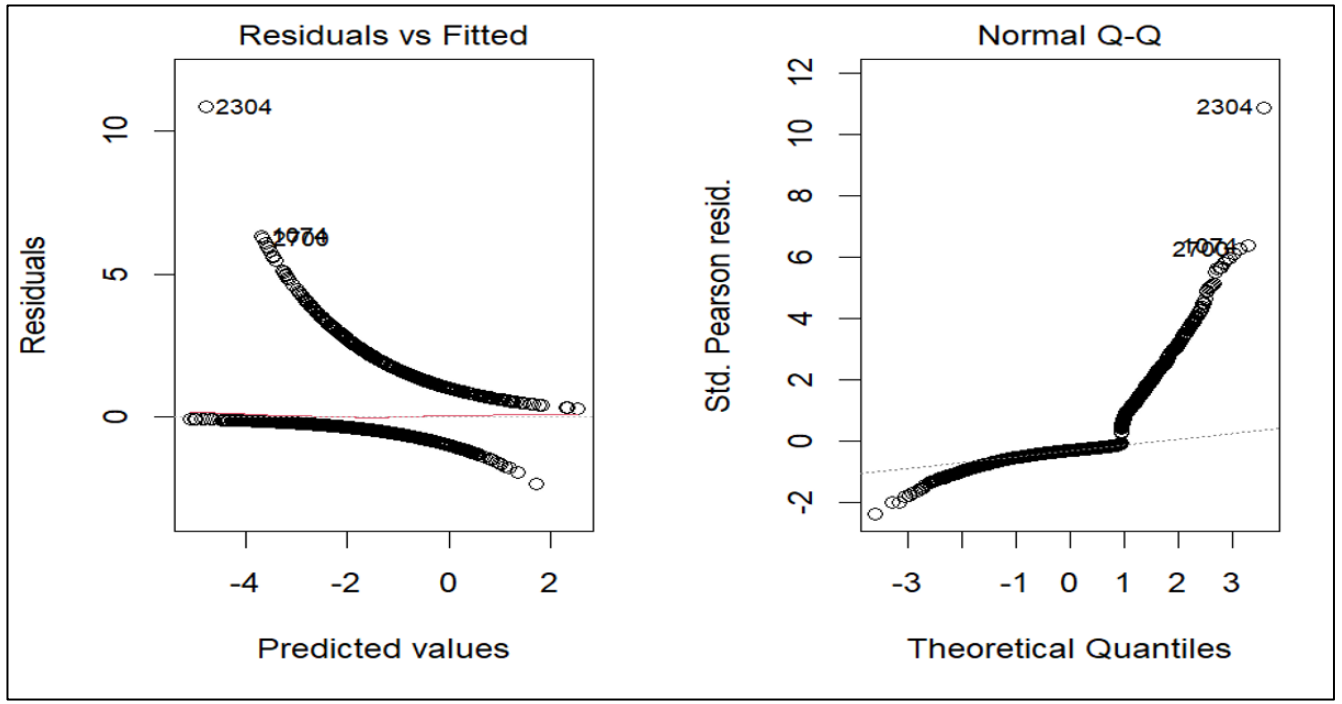
Description: df [2 × 2]

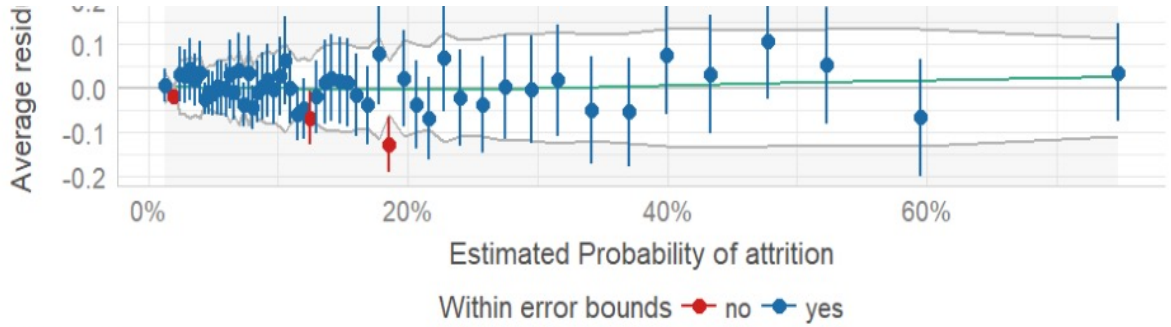
role <fctr>	#variables <int>
outcome	1
predictor	20

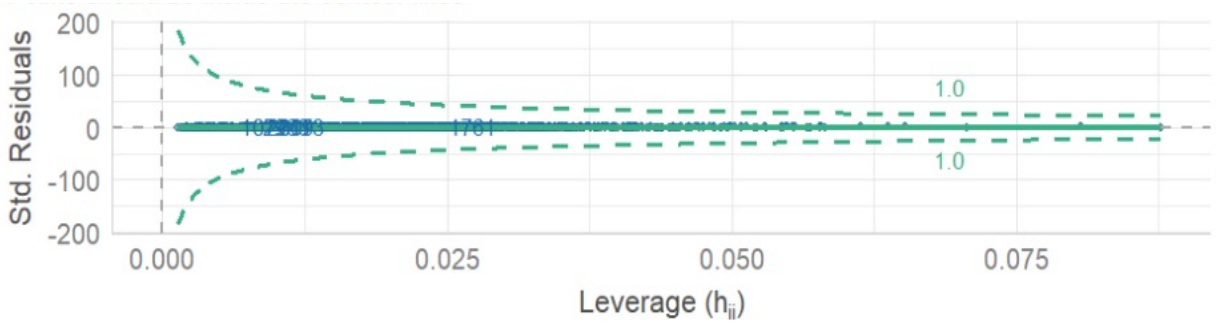
2 rows

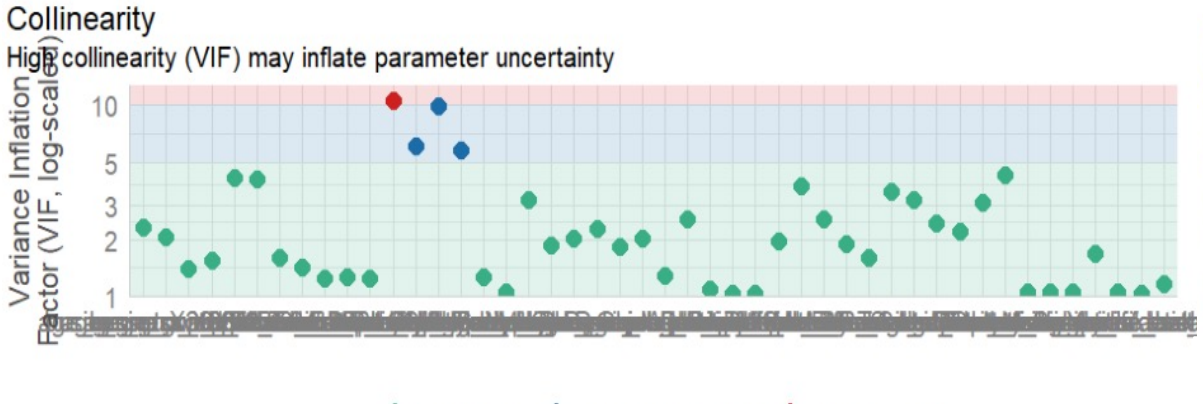
PHASE III – STEP II MODEL BUILDING

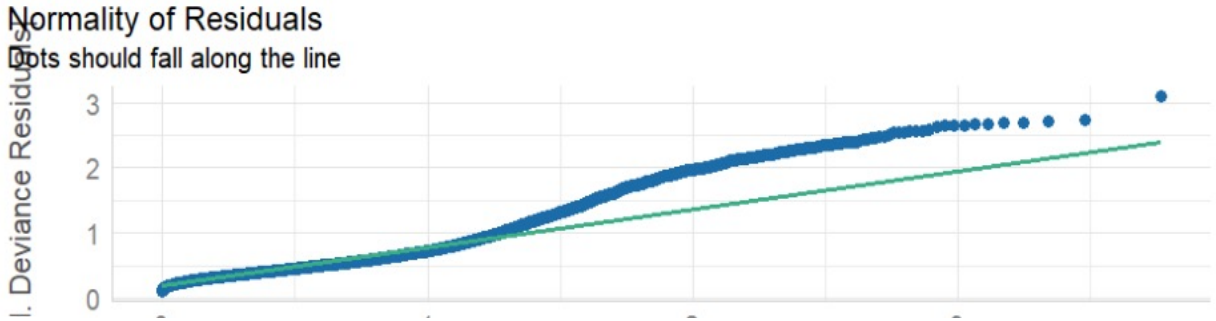
PERFORMANCE CHECK FOR LOGISTIC REGRESSION(GLMNET)











PERFORMANCE CHECK FOR LOGISTIC REGRESSION(GLMNET)

XGBOOST CLASSIFIER

- Extreme gradient Boosting algorithm with 3 parameters
- xgBoost further speeds up the training process by using parallel processing
- Better performance is achieved with 800 trees & 4 response variable at each leaf node

```
xg_model <- boost_tree(mode = "classification", # binary response
                       trees = tune(),
                       mtry = tune(),
                       loss_reduction = tune()) # parameters to be tuned
xg_wf <-
  workflow() %>% add_model(xg_model) %>% add_recipe(logis_recipe)

xg_grid <- xg_wf %>%
  tune_grid(boot_strap,
            metrics = metric_set(accuracy), control = control,
            grid = crossing(trees = 800, mtry = c(3, 4),
                            loss_reduction = c(0.001, 0.01)))

xg_grid %>% show_best(metric = "accuracy") %>% mutate(.metric = "accuracy")
```

A tibble: 1 × 4

mtry	trees	loss_reduction	.config
<dbl>	<dbl>	<dbl>	<chr>
4	800	0.01	Preprocessor1_Model4

XGBOOST MODEL PREDICTIONS

A tibble: 4 × 9

mtry	trees	loss_reduction	.metric	.estimator	mean	n	std_err
<dbl>	<dbl>	<dbl>	<chr>	<chr>	<dbl>	<int>	<dbl>
4	800	0.010	accuracy	binary	0.8944924	51	0.002399145
4	800	0.001	accuracy	binary	0.8935230	51	0.002381731
3	800	0.001	accuracy	binary	0.8934423	51	0.002404739
3	800	0.010	accuracy	binary	0.8931017	51	0.002434252

A tibble: 1,425 × 7

id	.pred_No	.pred_Yes	.row	.pred_class	attrition	.config
<chr>	<dbl>	<dbl>	<int>	<fctr>	<fctr>	<chr>
train/test split	0.3193222284	6.806778e-01	1	Yes	Yes	Preprocessor1_Model1
train/test split	0.1134401783	8.865598e-01	2	Yes	Yes	Preprocessor1_Model1
train/test split	0.9992475510	7.524490e-04	5	No	Yes	Preprocessor1_Model1
train/test split	0.9998456240	1.543760e-04	7	No	No	Preprocessor1_Model1
train/test split	0.7819626927	2.180373e-01	17	No	No	Preprocessor1_Model1
train/test split	0.9957861304	4.213870e-03	26	No	No	Preprocessor1_Model1
train/test split	0.9964895844	3.510416e-03	29	No	No	Preprocessor1_Model1
train/test split	0.9782382250	2.176178e-02	31	No	No	Preprocessor1_Model1
train/test split	0.9833704233	1.662958e-02	36	No	No	Preprocessor1_Model1
train/test split	0.0569363683	9.430636e-01	38	Yes	No	Preprocessor1_Model1

1–10 of 1,425 rows Previous 1 2 3 4 5 6 … 100

Predicted attrition

Actual attrition

MODEL EVALUATION

CONFUSION MATRIX LEVELS

	Implication	Criticality
True Positive	Correctly predicted employee will leave.	**High** - Allows company to retain the employees who will leave via early intervention OR avoid hiring such employees in the selection stage to decrease attrition rates.
True Negative	Correctly predicted employee will not leave.	**Low** - Generally, not a critical issue for the company.
False Positive	Predicted employee will leave but did not leave.	**Medium** - Loss of potential talent
False Negative	Predicted employee will not leave but left.	**High** - The company would have hired and exhausted resources on investing in employees who would leave. This increases attrition rates and hurts the productivity of the company.

EVALUATION METHODS

Legend:-
True positives(TP)
False positives(FP)
True negatives(TN)
False negatives(FN)

- **Precision:** What proportion of positive identification was correct

  Precision = TP/(TP+FP)

- **Recall:** What proportion of actual positive were identified correctly

  Recall = TP/(TP +FN)

- **F1 Score:** It is the harmonic mean of precision and recall.

  F1 = 2 * Precision * Recall /(Precision + Recall)

- **Accuracy:** how many observations, both positive and negative, were correctly classified.

  ACC = TP/(TP +FN)

Penalized Logistic regression

Precision	Recall	F1-score
0.54	0.69	0.69

Random Forest

Precision	Recall	F1-score
0.96	0.67	0.80

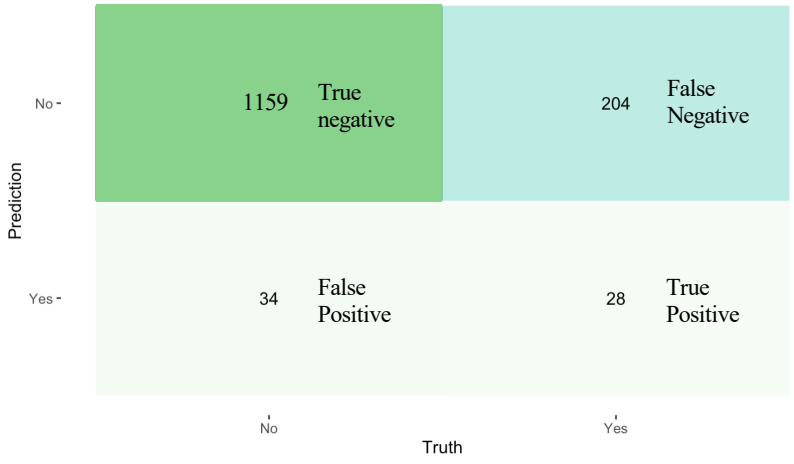
xgBoost

Precision	Recall	F1-score
0.82	0.77	0.87

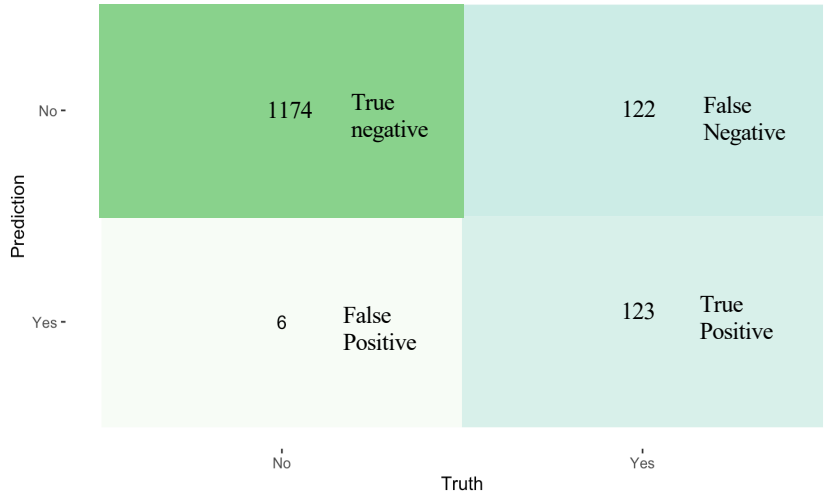
COMPARISON OF F1-SCORE FOR PENALIZED LOGISTIC REGRESSION, RANDOM FOREST & xgBOOST

BEST MODEL

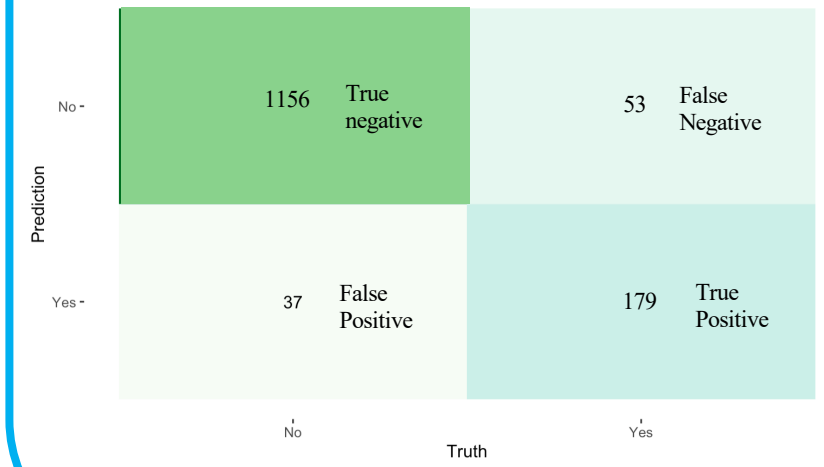
Penalized Logistic Regression



Random Forest Classifier



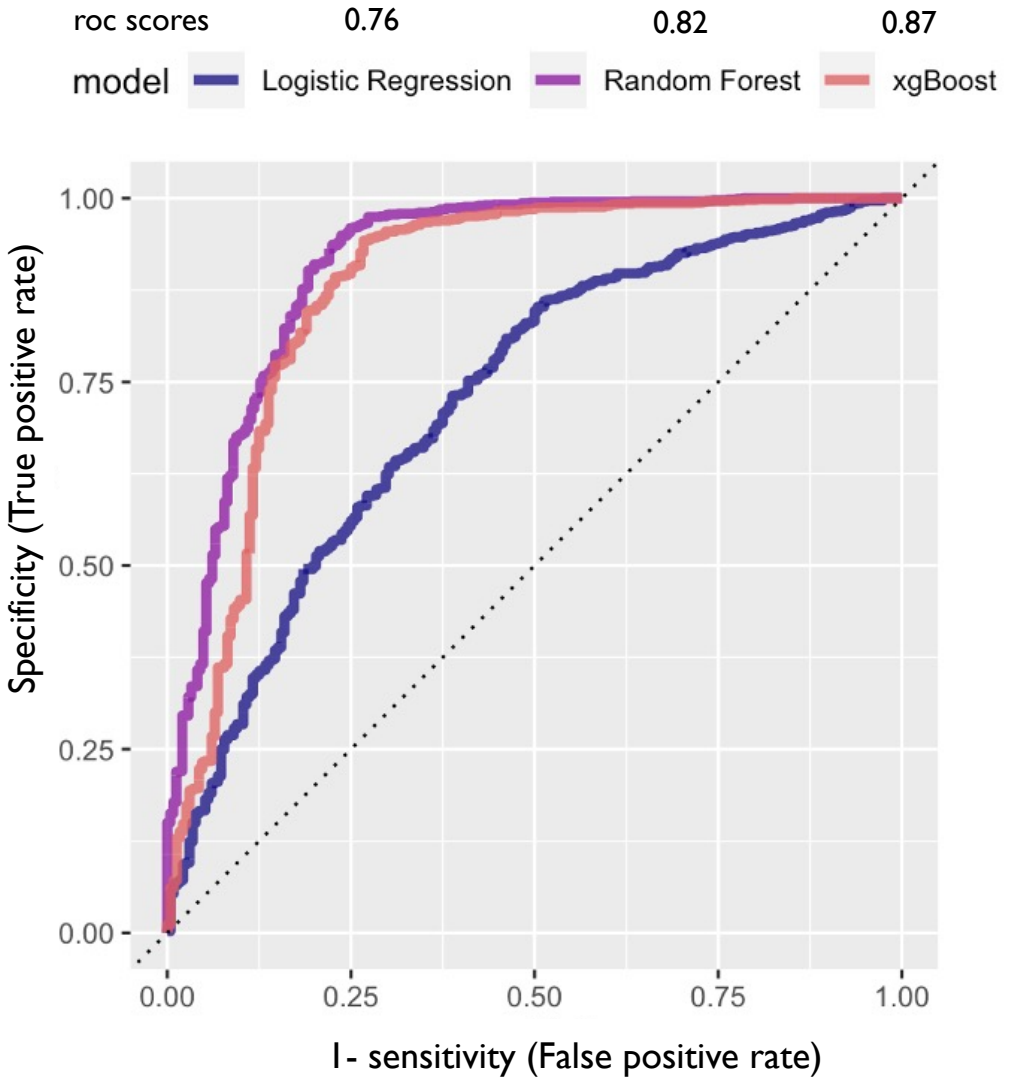
xgBoost



CONFUSION MATRIX FOR 3 MODELS

ROC_AUC CURVE

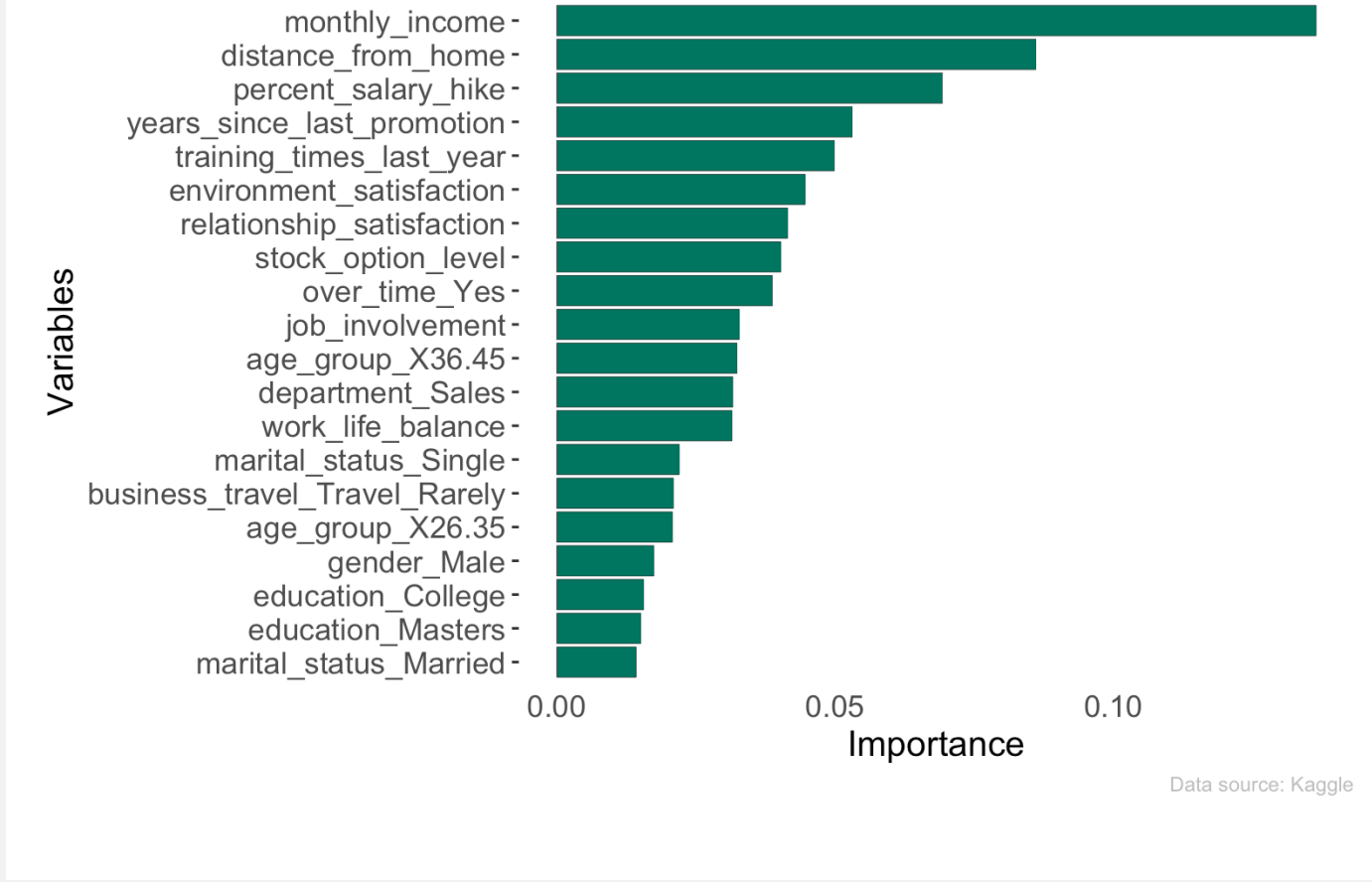
- Used for measuring model performance
- Higher the better(Close to 1)
- The ROC curve is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR)
- xgBOOST classifier with a higher ROC AUC value indicates better performance.



WHAT WE FOUND INFLUENCE EMPLOYEES TO LEAVE

Top 5 influencing factors are,

- Monthly income (Satisfactory pay)
- Distance from home
- Percent Salary Hike
- Work environment satisfaction (Working conditions)
- Years since last promotion



CONCLUSION

- We predicted the feature variable "attrition" using 3 supervised models.
- Out of the 3 main models, xgBoost Classifier is the best with an accuracy of 89%, followed by Random Forest model with an accuracy of 88% and Logistic Regression model with the least accuracy of 83%.
- Therefore, boosted ensemble trees proves more accurate than other models.

RECOMMENDATIONS

- Develop learning programs/training
- Providing transportation allowances
- Promotions opportunities for income below $40K
- Offer higher stock options

FUTURE WORK

- Overcome the weakness b/w more model parameters and computational time
- Reverse engineer the problem to find what made employees stay
- With more updated data, the algorithms can be re-trained to find high-risk employees based on probabilistic labels assigned to each feature variable

THANK YOU