Proactive HR Decision Making using ML

Julian Mirijello, Richard Tan, Danyang Wang, Omar Abouelmagd

Table of contents

01

Problem Scoping

Business problem

Background

- ABC 🔤, a SAAS company based in Toronto, recently conducted an HR pulse survey and employee census
- According to the YoY results,
  - ⬇️Employee satisfaction,
  - ⬆️ Employee churn
  - ⬇️ Productivity and performance
- The Chief People Officer (CPO) does not want to present these results to the board without offering solutions – year end results are only in a few weeks

How can we help them build a suite of HR tools?

Business Problem Statement:

How can ABC use predictive modeling to keep a pulse on their company culture and effectively improve key HR metrics in 2025?

Business value

People and talent are a company's most valuable asset

- According to researchers, **replacing mid-level and high-level employees** can cost between **125% - 400%** of their annual salary [1], when you consider both tangible and intangible costs (loss in productivity)
- **Employees dissatisfaction** cost companies $1.9 trillion in lost productivity last year [2]
- A HubSpot report [3] found that **lost productivity** costs U.S. businesses a shocking $1.8 trillion every year

[1] MGR Workforce
[2] Gallup
[3] Hubspot

Another real world example

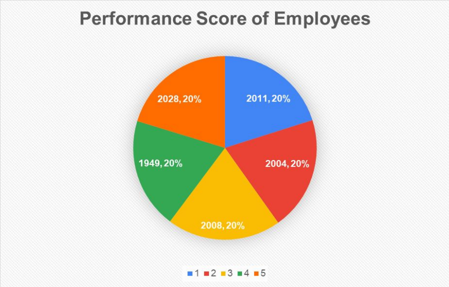
In 2022, **Amazon** was has a turnover rate of **100%** annually, more than double the industry average. This potentially cost the company **8 billion** dollars in value.[1]

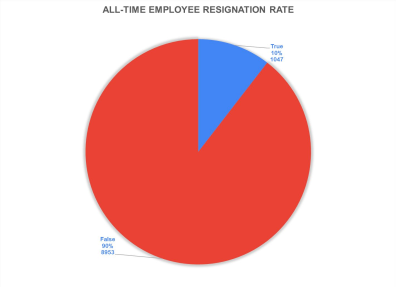
[1] Forbes

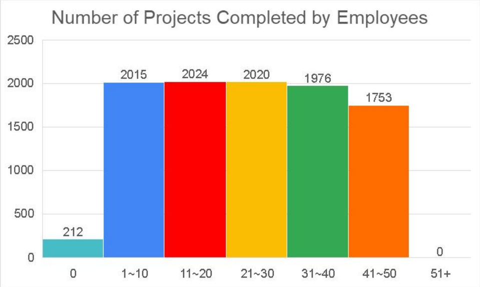
About the data

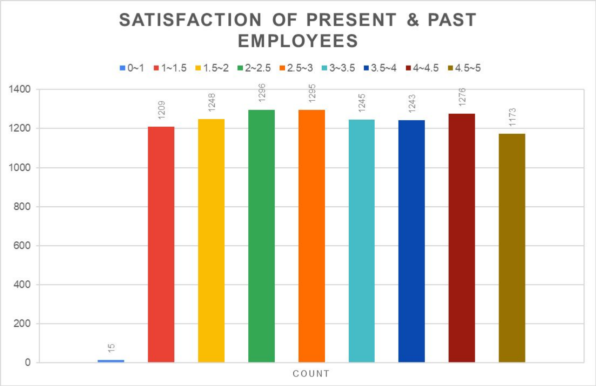
- Data collected as part of the HR pulse survey and employee census
- 100,000 rows and 20 features
- No *NULL* values, some variables will require label/one-hot encoding
- **Key Target Variables:**
  - Churn
  - Performance Score
  - Employee Satisfaction
  - Projects Handled (proxy for productivity)
- **Other Key Features**
  - Monthly Salary
  - Overtime Hours
  - Remote Work Frequency
  - Promotions
  - Education Level
  - Gender
  - Age

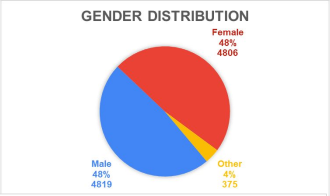
Understanding the data

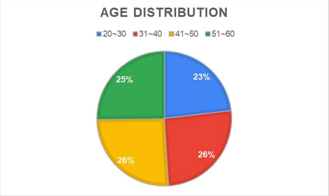


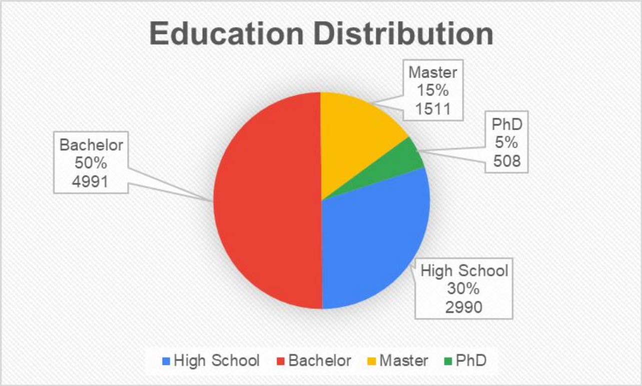












Key stakeholders

Chief People Officer (CPO)

Employees

Customers

02

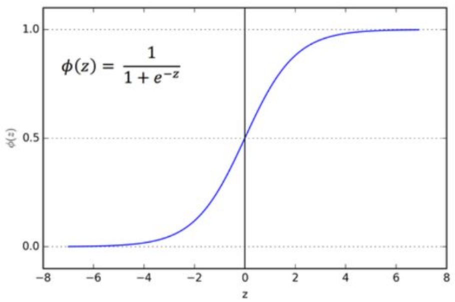
Churn Model

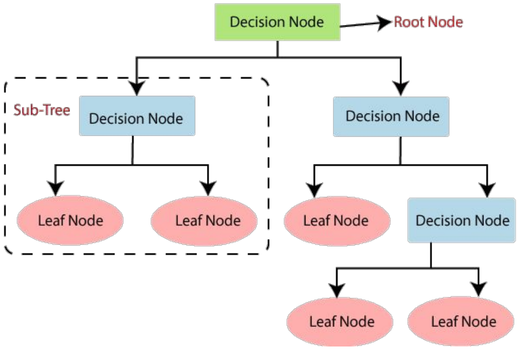
Machine learning problem

ML Problem Statement:

Build a binary **classification model** to predict the likelihood of employees resigning (churn prediction). By identifying potential resignation early, company ABC can initiate retention strategies and reduce turnover rates.

Models: Logistic Regression, Decision Trees





The data

Problem

- The original dataset is quite **imbalance** (only 10% of cases are marked Resigned), which caused initial versions of the model to predict the majority class most of the time (high TN and FN)

Solutions Implemented:

- Oversampling minority class (**resignations**) and undersampling majority classes (non-**resignations**)
    - 10/90 ❌vs 50/50 (reduced generalization) ❌ vs 30/70 ✅(happy medium)
- Updating Logistic regression class weighting = 'balanced'

Other Actions

- Label encoding data points like **Job_Title**, **Education Level,**
- Data cleaning:
    - Filtered out 'Consultants' as they are not relevant
    - Filtered employees with < 1 year tenure
- Feature engineering
- Split the data set into a training and testing set (80/20)
- Feature scaling

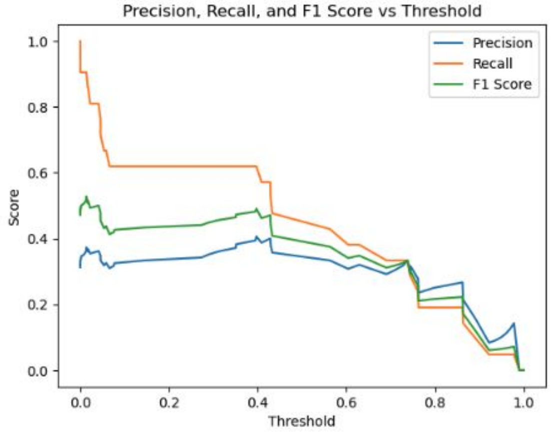
Measuring performance

Direct Metrics

- Since in the business case **false negatives are costly** (employees predicted not to resign, but actually did) , we want to optimize for **recall**
- With this said, we don't want to neglect precision because we want it to make the model as useful for real-world scenarios as possible – try to achieve a balanced **F1 score**
- We will use **Precision-Recall curve** to determine the optimal threshold, since this metric focuses on positive predictions, making it beneficial for imbalance data sets

In-Direct Metrics

- No too concern with speed as this task is not time sensitive



Experimentation

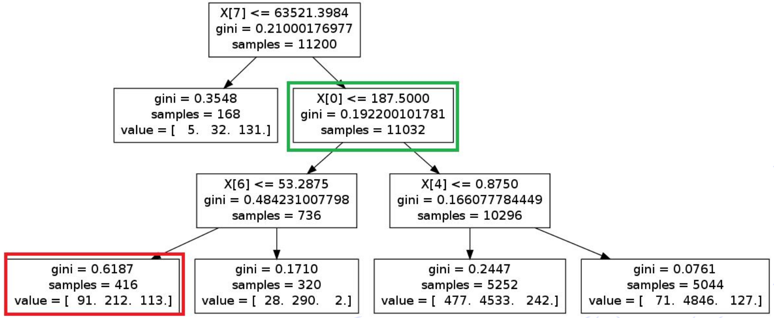
- Ran a baseline **Logistic Regression** model with degree of polynomial of **2** – very low F1 score
- Focused on improving the model by **(1)** adding complexity **(2)** finding the optimal threshold **(3)** collecting more / resampling data
  - **Adding complexity**: hyper-parameter tuning with GridSearch and feature engineering

```
#Grid of hyperparameter options
param_grid = {
'poly_features__degree': [1, 2, 3, 4], #Testing degrees
'logistic_regression__C': [0.01, 0.1, 1, 10, 100], #Regularization strength
'logistic_regression__penalty': ['l2'], #L2 penalty
}
```

  - **Feature engineering**
    - Tried creating interaction features (e.g. Overtime Hours x Number of Hours per Week, Age x Years at Company)
    - Squared, cubed features etc.

Experimentation Cont.

- After hours of experimentation, we started to see a flatline performance from our Logistic Regression model so we wanted to try other models to improve the performance further
- We used **Decisions Tree** models
  - First, DecisionTreeClassifier since it is optimized for classification tasks, and did parameter tuning
  - Tried different algorithms like RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
- We saw no meaningful improvement using these models, so we continued with the Logistic Regression model since it is simpler



Final Model Performance and Recos

Performance on Training Set

- Accuracy: **0.58**
- Precision: **0.39**
- Recall: **0.57**
- F1 Score: **0.46**

Hyper Parameters

- Regularization strength: **0.01**
- **L2 penalty**
- Polynomial Degree: **2**

Performance on Test Set

- Accuracy: **0.39**
- Precision: **0.32**
- Recall: **0.81**
- F1 Score: **0.45**

Takeaways/Recommendations

- The model has a high recall for resignations, but precision is low, meaning the model may trigger false alarms (aka False Positives)
- Therefore, this tool should only be used as a warning system for resignation (keeping a pulse)
- The need for intervention should be evaluated on a case by case basis, using the discretion of the HR team, the individual's manager, and how difficult/costly it may be to replace the employee
- For example, if the model raises alarm for the VP of Product, it might be worth looking into / intervening over an analyst

Areas for Improvement

This model is a good start but it can be improved with more time and resources

Ways of Improving

- Collect more data on the minority class (resigned employees) to minimize overfitting / impact of imbalance dataset
  - E.g., Conduct exit surveys / interviews with resigned employees
- Capture new features that could help predict churn
  - Examples:
    - Usage of vacation days
    - Time since last promotion
    - Distance of commute
    - Usage of benefits
    - Amount of equity
- Try new models model types that could add more nonlinearity to capture new relationships in the data

03

Performance Model

Analyzing Performance Score

ABC ranks employee performance on a scale from 1-5. We want to know how we can best predict an employee's performance score through **classification**, specifically with **Softmax regression**.

Feature Definition

	VIF Factor	features
0	18826484.25	const
1	1.00	Employee_ID
2	1.00	Department
3	1.00	Gender
4	1.00	Age
5	1.03	Job_Title
6	38.80	Hire_Date
7	38.79	Years_At_Company

Variance Inflation Factor (VIF)

	Employee_ID	Department	Gender	Age	Job_Title	Years_At_Company	Education_Level	Performance_Score	Monthly_Salary	Work_Hours_Per_Week
Employee_ID	1.00	0.01	0.00	0.00	0.00	-0.01	-0.00	-0.00	-0.00	-0.00
Department	0.01	1.00	-0.01	-0.00	0.00	-0.00	-0.00	-0.00	-0.00	0.00
Gender	0.00	-0.01	1.00	-0.00	-0.00	-0.00	0.00	0.00	-0.00	0.00
Age	0.00	-0.00	-0.00	1.00	0.00	0.00	0.00	0.00	0.00	-0.00
Job_Title	0.00	0.00	-0.00	0.00	1.00	0.01	-0.00	0.00	-0.17	-0.00
Years_At_Company	-0.01	-0.00	-0.00	0.00	0.01	1.00	-0.00	0.00	-0.00	0.00
Education_Level	-0.00	-0.00	0.00	0.00	-0.00	-0.00	1.00	0.01	0.01	-0.00
Performance_Score	-0.00	-0.00	0.00	0.00	0.00	0.00	0.01	1.00	0.51	-0.01
Monthly_Salary	-0.00	-0.00	-0.00	0.00	-0.17	-0.00	0.01	0.51	1.00	-0.00
Work_Hours_Per_Week	-0.00	0.00	0.00	-0.00	-0.00	0.00	-0.00	-0.01	-0.00	1.00
Projects_Handled	-0.00	0.00	0.00	-0.00	0.00	0.00	-0.00	0.00	-0.00	-0.00
Overtime_Hours	-0.01	0.00	-0.00	0.00	-0.00	0.00	0.00	-0.00	-0.00	0.01
Sick_Days	0.00	-0.01	0.00	0.01	-0.01	-0.00	-0.00	0.00	0.00	-0.00
Remote_Work_Frequency	-0.00	-0.00	0.00	-0.00	0.00	-0.00	-0.00	0.00	-0.00	-0.00
Team_Size	0.00	-0.00	-0.00	-0.00	0.00	0.00	-0.00	-0.01	0.00	0.00
Training_Hours	-0.00	-0.01	-0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	0.00
Promotions	-0.00	0.00	0.00	-0.00	0.00	-0.00	0.00	-0.00	-0.00	0.00
Employee_Satisfaction_Score	-0.00	0.01	-0.00	-0.00	-0.00	-0.00	-0.00	0.00	0.00	0.00
Resigned	-0.00	0.00	-0.00	0.00	-0.00	0.00	0.00	-0.00	-0.00	0.00

Correlation Matrix

Data Processing

	Employee_ID	Department	Gender	Age	Job_Title
0	1	IT	Male	55	Specialist
1	2	Finance	Male	29	Developer
2	3	Finance	Male	55	Specialist
3	4	Customer Support	Female	48	Analyst

Label encoding was used to transform strings into integer values for the Softmax model.

Key Takeaways: Hire date (year) is redundant, monthly salary is important. Drop Hire_Date from features and ensure that Monthly_Salary is kept.

Model Performance Improvement

The initial model yielded an accuracy of 0.35. This indicates that our model would be 15% more accurate than random guessing, which isn't great. How could this be improved?

Smaller Set of Features?
What if we used a subset of the original features?

- Age
- Job Title
- Monthly Salary
- Hours Worked per Week
- Projects Handled
- Employee Satisfaction

Result: **0.33**

Polynomial Features?
Result: **0.54**

Hyperparameter Tuning?
How about when we determined the optimal model parameters?
Result: **0.62**

What does this mean?
The fourth iteration of this model, with hyperparameter tuning, has an accuracy of 0.62, more than tripling the accuracy of a random guess.

Insights from Performance Score

Performance score can have important implications for a company, especially when deciding on promotions. Employees who work hard to improve their performance score can be rewarded accordingly, which helps reduce attrition and workplace dissatisfaction.

Data-Driven Decisions
It is important to have an accurate model so ABC can properly classify employees. Based on the prediction and actual performance score, ABC can clearly see which employees are performing above expected level and which employees are performing below expected level. This can support management decisions for promotions and other career outcomes.

04

Conclusion

Final Insights

CHURN Model

- High employee turnover is a **costly issue** for ABC (and companies in general)
- The CHURN model would essentially be an **early warning system** to flag any employees at risk of resignation
- This will give time for **proactive interventions**, which will can reduce turnover-related costs and loss of productivity

Performance Model

- The performance model can be used in the decision making process of **promoting employees**, and organizational restructuring to ensure optimal output.
- This is assisted by the model's ability to **predict performance levels** of employees according to their environment and compensation

How Both Models Align

A Holistic System

- **Churn Model:** Predicts **potential departures**, enabling better forecasting of staffing needs.
- **Performance Model:** Helps identify internal candidates for **promotions** or skill gaps to be filled.
- **Together:** Plan succession strategies to ensure **seamless transitions** and **minimize disruption.**

Unified HR Predictive Tool

By integrating both models into a **single dashboard**, ABC can continuously monitor HR metrics:

- Are they likely to **leave**? (CHURN Model)
- How are they **performing**? (Performance Model)
- What **proactive actions** can we take?

Due to the nature of the tools, it is important to keep them **up-to-date** with **current data** HR data.

Thank You!

Time for Q&A