# Amazon Employee Resource Access Prediction

Auhtor: Igor Ljucović

Date: 8.9.2025.

## 1. Introduction

The Amazon dataset contains **32,769 observations (rows)** and **10 columns**, of which **one is the target column ("ACTION")** — the value to be predicted, indicating whether an employee should be granted access to a resource or not. One column represents the resource ID, one the manager ID, and the remaining seven columns contain information related to the employee.

*Note: All variables are represented as IDs without semantic meaning. Amazon did this to ensure data privacy — for example, every occurrence of "ROLE_TITLE" equal to "PROJECT_MANAGER" was replaced with the corresponding ID `176508`.*

The goal is to predict, based on historical data of employee requests for resource access, whether a new request (for example, access to a specific website) should be approved (1) or denied (0).

Employees often request access to various applications, and ideally, they should only be granted access to those that increase their productivity at work, because:

1. **Incorrectly granting access** may lead to **security breaches** or **delays** instead of task completion.
2. **Incorrectly denying access** may **reduce employee productivity**.

Due to the large number of access requests, manual evaluation is not scalable. If access is wrongly granted or denied, valuable time is lost while the manager or responsible employee manually corrects access rights during working hours, which is inefficient. Therefore, machine learning algorithms are used to learn patterns from previous examples and assist in automated decision-making. Before using these models, it is necessary to properly prepare the data, and after training, to evaluate the models' performance in order to draw meaningful conclusions.

## 2. Data preparation

The column "ROLE_CODE" contains the same information as the column "ROLE_TITLE", so we can reduce the dimensionality of the problem by one column by **removing "ROLE_CODE."**

Creating a smaller subset of the dataset was not necessary because the models had no issues with time efficiency when working with Amazon's dataset of about 32,000 rows.

**Data normalization** (e.g., scaling values to a 0–1 range) **was unnecessary** because all columns are categorical, i.e., non-numerical. However, it was necessary to perform **one-hot encoding** before using the **MLP (neural network)** model.

**Removing rows** that deviate by certain characteristics (**outliers**) **was necessary** because the columns **"ROLE_TITLE"** and **"ROLE_FAMILY_DESC"** contained certain values that appeared only 1, 2, or 3 times. These values represented **noise**, meaning there were not enough observations (rows) for the models to learn useful patterns from them for prediction. From Amazon's dataset, 0.23% of rows were removed due to outlier values in the "ROLE_TITLE" column – the number of unique values was reduced from 343 to 290, i.e., 53 of the rarest values were removed. Then, from the remaining rows, as much as 12.03% of the data was removed because of outlier values in the "ROLE_FAMILY_DESC" column – the number of unique values was reduced from 2358 to 700, i.e., 1658 of the rarest values were removed. Although 12.03% is a large portion of removed rows, this change positively affected the performance of all models – which makes sense, since high-cardinality columns (categorical variables with a large number of possible values) can cause a model to learn from noise. If there are only 1–3 rows with a certain value, the model won't have enough data to draw reliable conclusions from those rows, and they only confuse the model – hence their removal. Another column with very high cardinality – in fact, the highest among all columns – is "MGR_ID"**,** with as many as 2540 unique values across the remaining 28,000 rows. It was not worth removing outliers from this column, as doing so reduced model performance. This is because "MGR_ID" turned out to be one of the most useful columns for prediction in many models. Moreover, there was no manager ID that appeared significantly more or less frequently than others, so there were far fewer extremely rare values for which the model couldn't learn anything meaningful.

## 3. Models

Seven machine learning models were evaluated, of which three performed poorly, one performed moderately, and three performed excellently – depending on which parameters were prioritized in prediction.
Since our dataset consists exclusively of categorical variables, models capable of handling qualitative (categorical) variables performed better (such as those based on decision trees and/or boosting) than models that work best with quantitative (continuous, numerical) variables (such as linear regression).

The model parameters were tuned to optimize metric values, with the main focus on maximizing the ROC-AUC score, as it was stated to be the most important metric on Amazon's Kaggle page for this dataset.

| Predictive power | Model | Model | Model | Comment |
|---|---|---|---|---|
| Poor | Logistic Regression | | | Too simple |
| Moderate | Cat Boost | Neural Network | Logistic Regression with Random Forest stacking | Other models were better in at some aspects |
| Excellent | Random Forest | Decision Tree | Light GBM | Potentially the best model, depending on your criteria |

## 4. Evaluation

Metrics:

1. **ROC-AUC** – how much better the model predicts compared to the simplest possible model that always predicts the most common case (that access should be granted)
2. **PRECISION** – out of all employees to whom the model granted access, how many actually should have been granted access
3. **SPECIFICITY** – out of all employees who should not have been granted access, how many were correctly denied by the model
4. **RECALL** – out of all employees who should have been granted access, how many were correctly identified by the model.
5. **F1** – a combination of Precision and Recall; if either of the two metrics is low, this metric will also be low
6. **TIME** – the time (in seconds) needed for the model to be trained and to make a decision about granting access to an employee
7. **OVERALL RANK** – the model's rank when all six metrics are considered equally important; a model gets 1 point for a metric if it performs the worst among all five models, and 5 points if it performs the best (the best overall model is ranked 1st)
8. **SUBJECTIVE RANK** – the model I personally consider the best. I valued **Specificity** the most – the model's ability to accurately predict cases where access should not be granted, because unlike the cases where access should be given, here there is not only a potential drop in productivity but also a security risk

Performance of the 5 best models:

| Model | ROC-AUC | PRECISION | SPECIFICITY | RECALL | F1 | TIME | OVERALL RANK | SUBJECTIVE RANK |
|---|---|---|---|---|---|---|---|---|
| Random Forest | 0.880 | 0.965 | 0.439 | 0.985 | 0.975 | 5.64 | 1 (25p) | 3 |
| Decision Tree | 0.809 | 0.755 | 0.732 | 0.757 | 0.853 | 0.20 | 5 (14p) | 1 |
| Light GBM | 0.875 | 0.937 | 0.646 | 0.954 | 0.965 | 2.68 | 4 (17p) | 2 |
| Cat Boost | 0.874 | 0.940 | 0.608 | 0.960 | 0.967 | 13.99 | 2 (18p) | 4 |
| Neural Network | 0.879 | 0.956 | 0.281 | 0.960 | 0.967 | 64.90 | 2 (18p) | 5 |

## 5. Conclusion

There is no objectively best model for this dataset:

1. If it is most important to minimize granting access to someone who should not have received it (due to security or reduced productivity) **– Decision Tree**
2. If it is most important to minimize denying access to someone who should have received it (to avoid productivity loss) – **Random Forest**
3. If neither criterion is significantly more important than the other **– LightGBM**

For prediction, the models benefited most from the columns "RESOURCE", "MGR_ID", and "ROLE_DEPTNAME" while the least useful columns were "ROLE_ROLLUP_1" and "ROLE_FAMILY."
The table below shows the five most important columns for each of the best performing models for the dataset:

| Model | 1st most important column | 2nd most important column | 3rd most important column | 4th most important column | 5th most important column |
|---|---|---|---|---|---|
| Random Forest | RESOURCE | MGR_ID | ROLE_DEPTNAME | ROLE_FAMILY_DESC | ROLE_ROLLUP_2 |
| Decision Tree | MGR_ID | ROLE_DEPTNAME | ROLE_FAMILY_DESC | RESOURCE | ROLE_ROLLUP_2 |
| Light GBM | RESOURCE | MGR_ID | ROLE_DEPTNAME | ROLE_FAMILY_DESC | ROLE_TITLE |