# Amazon.com - Employee Access Challenge

Machine Learning | Final Project

Xiaotian Huang, Cheng Zeng, Wenyu Zeng

# Introduction

- Amazon Employee Access - computer access
- Protects user information, and prevent unauthorized access, disclosure, disruption, modification, inspection, recording or destruction.
- Employees access encounter roadblocks
  - Not able to log into a reporting portal
- Access discovery/recovery cycle wastes a nontrivial amount of time and money

## Objective

- Creating a best model to determine an employee's access needs

# About the dataset...

- The data consists of real historical data collected from 2010 & 2011
- Original file
  - Train
    - ACTION (ground truth), RESOURCE, and information about the employee's role at the time of approval
  - Test
    - Each row asks whether an employee having the listed characteristics should have access to the listed resource.
- Column Descriptions
  - Action, Resource, Manager ID, Role grouping Category, Role department description, Role title, Role family description and role code.
- Target: Action

| | ACTION | RESOURCE | MGR_ID | ROLE_ROLLUP_1 | ROLE_ROLLUP_2 | ROLE_DEPTNAME |
|---|---|---|---|---|---|---|
| 0 | 1 | 39353 | 85475 | 117961 | 118300 | 123472 |
| 1 | 1 | 17183 | 1540 | 117961 | 118343 | 123125 |
| 2 | 1 | 36724 | 14457 | 118219 | 118220 | 117884 |
| 3 | 1 | 36135 | 5396 | 117961 | 118343 | 119993 |

# Data Preprocessing

- Original training dataset

| | # rows | # columns |
|---|---|---|
| **0** | 32769 | 10 |

- Splitted training dataset into training (80%) and validation (20%).
- After cleaning the missing values, uncommon values, and identifiers
- Training

| | # rows | # columns |
|---|---|---|
| **0** | 26215 | 10 |

- Validation

| | # rows | # columns |
|---|---|---|
| **0** | 6554 | 10 |

# Models

```
models = {'lr': LogisticRegression(class_weight='balanced', random_state=42),
          'dtc': DecisionTreeClassifier(class_weight='balanced', random_state=42),
          'rfc': RandomForestClassifier(class_weight='balanced', random_state=42),
          'hgbc': HistGradientBoostingClassifier(random_state=42),
          'xgbc': XGBClassifier(seed=42),
          'mlpc': MLPClassifier(early_stopping=True, random_state=42)}
```

- We use sklearn StandardScaler to do hyperparameter tuning and model selection
- We creating the dictionary of the parameter grids
  - Acronym of the model
  - The value is the parameter grid of the model.

- We use sklearn GridSearchCV to fine-tune the hyperparameters the models.

# Model Comparison

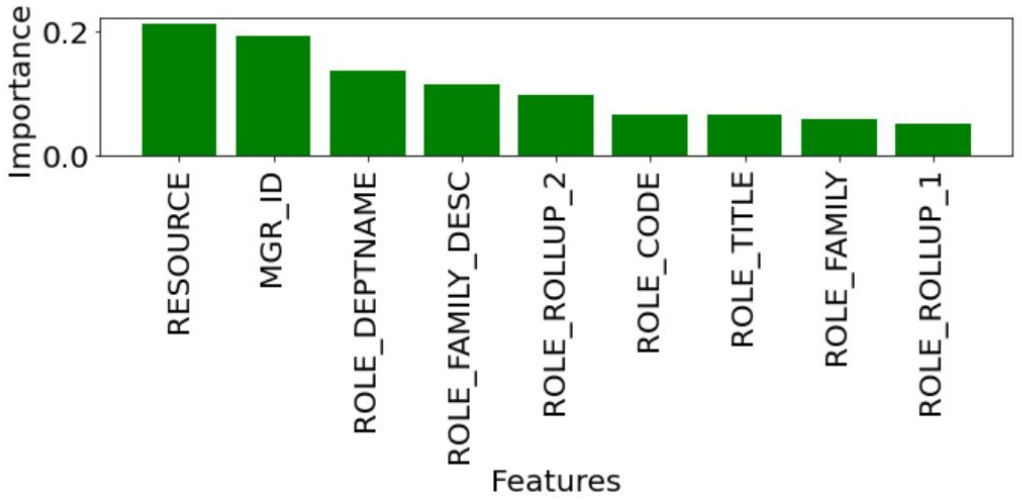
- For the best scores, we considered some important values of each model, including 'rank_test_score', 'mean_test_score', 'std_test_score', 'mean_train_score', 'std_train_score', 'mean_fit_time', 'std_fit_time', 'mean_score_time', 'std_score_time'.

| | best_score | best_param | best_estimator |
|---|---|---|---|
| **0** | 0.948886 | {'model__min_samples_leaf': 1, 'model__min_sam... | ((DecisionTreeClassifier(ccp_alpha=0.0, class_... |
| **1** | 0.945224 | {'model__learning_rate': 0.1, 'model__min_samp... | (HistGradientBoostingClassifier(l2_regularizat... |
| **2** | 0.942478 | {'model__alpha': 1e-05, 'model__learning_rate_... | (MLPClassifier(activation='relu', alpha=1e-05,... |
| **3** | 0.942325 | {'model__eta': 0.0001, 'model__gamma': 0, 'mod... | (XGBClassifier(base_score=0.5, booster='gbtree... |
| **4** | 0.754959 | {'model__max_depth': 1, 'model__min_samples_le... | (DecisionTreeClassifier(ccp_alpha=0.0, class_w... |
| **5** | 0.433323 | {'model__C': 0.01, 'model__tol': 1e-06} | (LogisticRegression(C=0.01, class_weight='bala... |

# Conclusion

- Random Forest has the highest score, which means that the variables selected have high accuracy and are important for predicting the action.
- Since logistic regression model has a best score value of 0.43, thus logistic regression might not be a good model for prediction the “Action”.

| | Features | Importance |
|---|---|---|
| 0 | RESOURCE | 0.212523 |
| 1 | MGR_ID | 0.193336 |
| 2 | ROLE_DEPTNAME | 0.136798 |
| 3 | ROLE_FAMILY_DESC | 0.116233 |
| 4 | ROLE_ROLLUP_2 | 0.0975683 |



# Reference


1. https://www.kaggle.com/c/amazon-employee-access-challenge/data
2. https://github.com/yuxiaohuang/teaching/tree/master/gwu/machine_learning_I/spring_2020


thank
you