CS5691 PRML

DATA CONTEST - 2020

EMPLOYEE CHURN PREDICTION

ME17B112 | ME17B120, Team 22

Mohammed Irfan T | Sujay Bokil

Introduction

The problem statement was to predict whether or not an employee would leave a company. Binary Classification.

Data Cleaning

Negative values for 'emp'

In the 'remarks' there were rows with negative values for emp. As suggested on the competition site, these rows were dropped.

Null Values in 'remarks', 'txt' column.

Nan in the 'txt' field was replaced with an empty string. This was to use the field later to calculate the length of remark and use it as a feature.

Redundant Column, support, and oppose.

Since the two columns support and oppose conveyed the same meaning (one was the complement of the other), one was dropped.

Feature Engineering

"remarks" and "remarks_supp_opp"

- ‘txt’ field was replaced with the length of the string.
- All entries in ‘remarks_supp_opp’ for which the ‘remarkId’ occurred in the ‘remark’ file also was retained. (logic was to use support-oppose information for only the remarks which we are dealing with).
- The support column was label encoded. (1 for support -1 for oppose)
- Support factor: All remarks were grouped by ‘remarkId’ and **support_factor** was calculated. (sum and take mean of ‘support’ column for each group)
- Extract Time: Information from the ‘remarkDate’ field was extracted. This includes the **number of days since the last remark**, the number of days since the first remark, the remark day, remark month, etc. The number of days since the last remark later proved to be a useful feature later.

“Unique_id”

All files were provided with a unique_id, which was a tuple (emp, comp), employee, and company id combination, which was later used for all grouping and further feature engineering. This was to identify a unique employee.

“train”

With **“unique_id”** as the key for grouping, the following features were created:

Employee-Specific features: The average rating, number of ratings, and median rating given by an employee.

Company-Specific features: The data frame was grouped by the ‘comp’ field and features like the number of ratings, average rating, and median rating a company received were generated. The ‘comp’ (company) column was label encoded.

Time: A lot of features were created from the ‘lastratingdate’ field. This includes the **number of days since the last rating**, the rating day, month, year, quarter, day of the week, etc.

Including information from the “remarks” file to the “train” file.

Grouping by 'unique_id' for all matching unique_id's in the two files, features like **average remark length**, **average support factor**, the **average number of days since the last rating**, **number of remarks**, **number of days since the last and first rating**, mean absolute difference, etc. were added to the training file. For all rows in which the unique_id's didn't match, the field was filled with 0.

Polynomial Features

Selected Numerical columns were used to create new columns of degree 2.

Feature Selection

LGBM feature importance was used to rank features and eliminate the least important features. In the final step, there were **86 features**, including the polynomial features (generated from **19 base features**).

Modeling

XGboost Classification Algorithm was used for modeling.

Hyperparameters '**scale_pos_weight**' was chosen to be **5**, since that was the weight used for **weighted accuracy** score evaluation, for the competition. This hyperparameter was to deal with the **class imbalance**, (**80% of data was class 0**).

Other hyperparameters were tuned using **Randomized Search Cross-Validation**, and later using **Grid Search Cross Validation** on the limited search space.

```
(scale_pos_weight = 5,colsample_bytree = 0.7, max_depth= 2,n_estimators=
200,reg_lambda=1.2,subsample=0.8)  // Hyperparameters used.
```

Public Leaderboard Score.

0.89026, Position 6th