Linear Discriminant Analysis(LDA)による IBM社の離職分析とその改善策

LDAの次元削減を利用した探索的データ分析(EDA)による相関の調査と、
離職分析をもとにした新規事業の提案

有明工業高等専門学校　3年 齋藤 健吾

目次 - Index -

概要 - Abstract -

目的

離職が会社に及ぼす影響について分析し、会社に利益をもたらすための事業を提案する。

現状の分析・解析

今回の事業提案では、IBM社のデータセット (IBM HR Analytics Employee Attrition) [1] をもとに、
様々な特徴量からLDA(Linear Discriminant Analysis)の次元削減の作用を用いて分類問題を解いたのち、
EDA(探索的データ解析)の結果から、どのカラムがどのくらい離職に影響しているのかを示す。

事業の提案

離職分析の結果から改善策(事業内容)を提案し、提案によりどのくらいIBM社に利益が発生するのかを
回帰分析により予測する。

情報業の離職率の動向

市場(他職業)との比較

厚生労働省の「雇用動向調査結果の概要」[2] によると、2021年の情報業の離職率は9.1%という結果であった。この離職率は他の職種に比べ、決して高くはなく、年々離職率は減少傾向にあるように見えるが、全体平均との相対的な割合では年々増加傾向にある。

年	情報業	全体平均
2021年（令和3年）	9.1%	13.9%
2020年（令和2年）	9.2%	14.2%
2019年（令和元年）	9.6%	15.6%
2018年（平成30年）	11.8%	14.6%
2017年（平成29年）	10.5%	14.9%

情報業/全体平均
65.5%
64.8%
61.5%
80.8%
70.5%

離職率が高いことは、一見良くないように思われるが、会社の人件費や運用コストなど、様々な要因の上、離職の良し悪しを測る必要がある。

顧客データの概要

データセット

分析・予測に使うデータセット(IBM HR Analytics Employee Attrition)[1] の内容は以下のとおりである。

	Age	Attrition	BusinessTravel	DailyAchievement	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyAchievement
0	25	0	Travel_Rarely	1280	Research & Development	7	1	Medical	1	143	4	Male	64

JobInvolvement	JobLevel	JobRole	JobSatisfaction	MaritalStatus	MonthlyIncome	MonthlyAchievement	NumCompaniesWorked	Over18	OverTime	PercentSalaryHike	PerformanceRating	RelationshipSatisfaction
2	1	Research Scientist	4	Married	2889	26897	1	Y	No	5	1	3

StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany	YearsInCurrentRole	YearsSinceLastPromotion	YearsWithCurrManager	HowToEmploy	Incentive	RemoteWork
80	2	2	2	3	2	2	2	1	intern	0	4
80	3	5	2	3	5	3	0	3	intern	0	1
80	0	6	1	3	6	4	0	3	agent_A	0	2
80	1	5	2	1	5	2	0	2	New_graduate_recruitment	0	5
80	0	5	5	1	0	0	0	0	New_graduate_recruitment	0	1
80	0	5	3	3	5	4	0	4	New_graduate_recruitment	0	2

カラムの内容

年齢, 性別, 離職, 家から職場までの距離, 成果など様々な特徴量がある。
今回行うのは, 離職 (Attrition) 以外のカラムの値から離職するかを予測することである。

PyCaretによる分類モデルの選定

最適モデルの選定

`classfication.compare_models()` で様々なモデルの中から最も分類の精度がよいモデルを選定する。

右の図では、LDAがAccuracy, AUC, その他評価も全体的に高く、最も今回の分類モデルに適しているモデルだと言える。

今回はLinear Discriminant Analysis (LDAモデル)を用いて、このモデルの特性である次元削減(ある種の正則化)からデータの特徴を考察し、また特徴量と結果の相関を解析する。

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lda	Linear Discriminant Analysis	0.8784	0.8598	0.4044	0.7361	0.5154	0.4532	0.4830	0.0440
ridge	Ridge Classifier	0.8716	0.0000	0.2412	0.9000	0.3657	0.3227	0.4137	0.0320
lr	Logistic Regression	0.8473	0.7796	0.1868	0.6476	0.2746	0.2183	0.2777	0.8280
et	Extra Trees Classifier	0.8444	0.8269	0.0849	0.5333	0.1431	0.1127	0.1724	0.0890
dummy	Dummy Classifier	0.8385	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0450
knn	K Neighbors Classifier	0.8172	0.5900	0.0908	0.2844	0.1339	0.0644	0.0777	0.3150
qda	Quadratic Discriminant Analysis	0.7977	0.5898	0.2463	0.4114	0.2752	0.1716	0.1933	0.0490
rf	Random Forest Classifier	0.7976	0.7655	0.1051	0.3528	0.1018	0.0469	0.0871	0.1040
xgboost	Extreme Gradient Boosting	0.7839	0.7315	0.1426	0.2464	0.1287	0.0567	0.0753	0.0950
catboost	CatBoost Classifier	0.7829	0.7072	0.1243	0.2159	0.1029	0.0342	0.0537	1.4770
lightgbm	Light Gradient Boosting Machine	0.7781	0.5889	0.1110	0.0932	0.0873	0.0157	0.0173	0.1860
ada	Ada Boost Classifier	0.7751	0.5686	0.1040	0.1578	0.0972	0.0182	0.0233	0.0770
gbc	Gradient Boosting Classifier	0.7585	0.6132	0.1423	0.0957	0.0906	0.0161	0.0209	0.0980
svm	SVM - Linear Kernel	0.7579	0.0000	0.1577	0.0386	0.0619	0.0141	0.0245	0.0350
dt	Decision Tree Classifier	0.7565	0.5241	0.1790	0.3028	0.1438	0.0526	0.0803	0.0410
nb	Naive Bayes	0.7549	0.7860	0.6625	0.3607	0.4660	0.3250	0.3510	0.0420

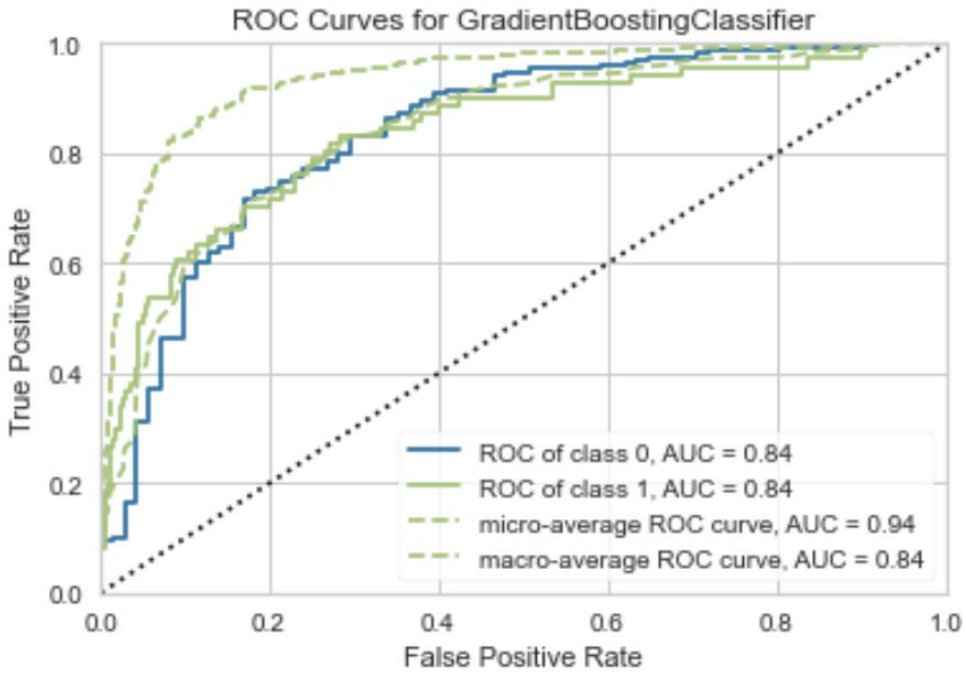
PyCaretによる分類モデルの評価

チューニングと評価

選定した最良のモデルのハイパーパラメータチューニングを行う。

`classfication.tune_model(model)` でチューニングを行った後、テストデータに対して予測を行い、10回の正確性(`Accuracy`)の平均を算出した。

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8350	0.5636	0.0000	0.0000	0.0000	0.0000	0.0000
⋮							
9	0.8529	0.8590	0.1875	0.6000	0.2857	0.2281	0.2766
Mean	0.8678	0.8101	0.2746	0.7534	0.3914	0.3371	0.3952
Std	0.0191	0.0410	0.1089	0.1816	0.1281	0.1265	0.1280



`classfication.plot_model(tuned_model)`

最適化した結果

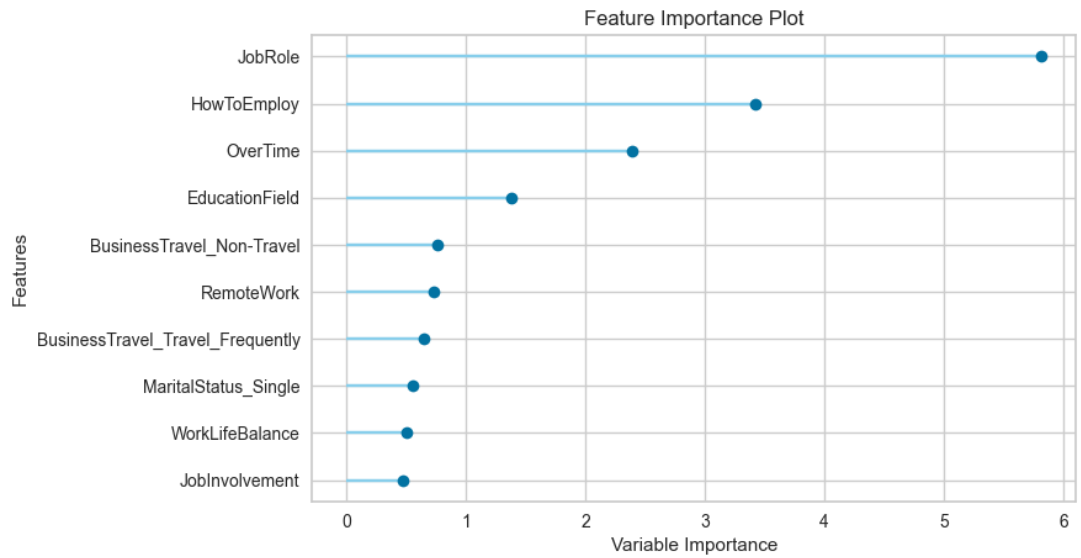
`Accuracy : 0.867` まで精度を上げることができた。

PyCaretによる特徴量の分析

特徴量の分析

チューニングしたモデルが、どの特徴量を主に使っているかを分析する。(通常と手順は逆)

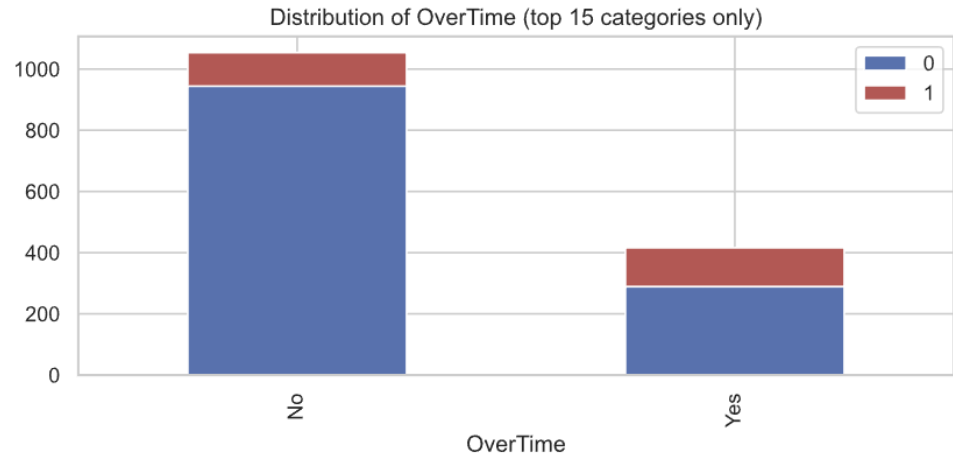
`plot_model(tuned_model, plot='feature')` でどの特徴量が重要かを可視化する。



分析の結果

`Variable Importance`として突出しているのは `JobRole`, `HowToEmploy`, `Overtime` のおおよそ3つが挙げられる。

これらに対して `Attrition` との関係をさらにグラフで見てみる。



PyCaretによる特徴量の分析

分析の結果

`JobRole`, `HowToEmploy`, `Overtime` の３つのグラフを見た。

`JobRole`

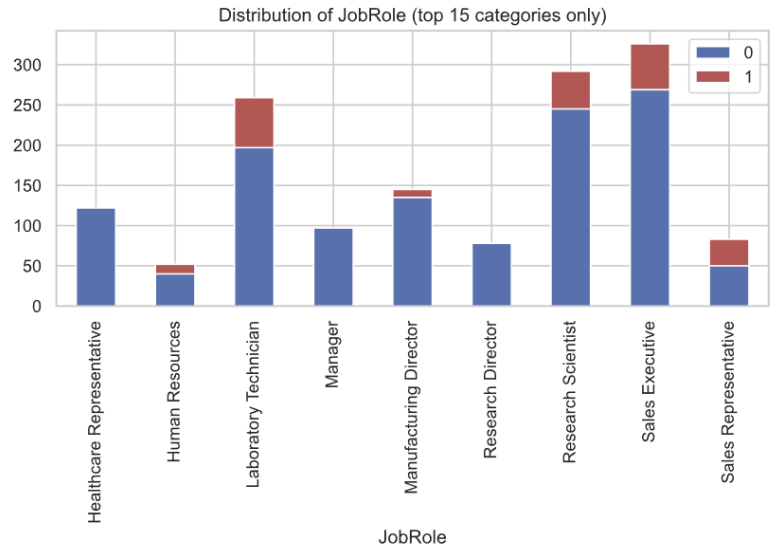
`LaboratoryTechnician`, `SalesRepresentative` が突出して `Attrition` の割合が高いことが分かる。

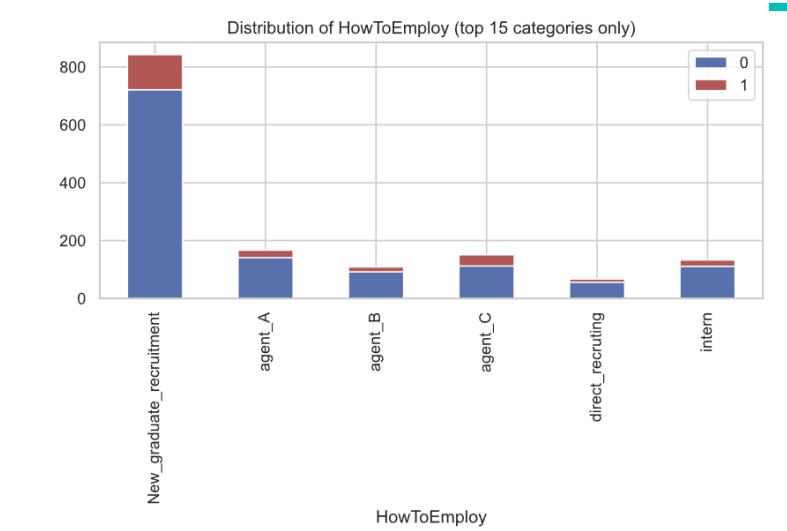
`HowToEmploy`

`New_graduate_recruitment` は総数が多いが、特別に他の項目と比べて `Attrition` の割合が高いということはない。
おそらく母数が大きいので、`0(=Attrition:False)` とするだけで評価が高くなってしまったのだと推測できる。

`Overtime`

これは他の特徴量よりも直感的で分かりやすい結果が出ている。
`Overtime:Yes` のとき `1(=Attrition:True)` の割合が高いことが読み取れる。また、`Overtime` は「時間外労働」であると推測できる。





離職分析をもとにした新規事業の提案

離職分析から予測・懸念されること

`Overtime:No` の場合、離職(`Attrition`)は減るか?

`Overtime:No` の場合、成果(`Achievement`)も減るか?

離職率を下げることは会社の利益にとって本当に良いか?

検証すること

`Overtime:No` の場合の離職(`Attrition`)の予測(回帰分析)

`Overtime:No` の場合の成果(`Achievement`)の予測(回帰分析)

離職率を下げることによる人件費などを考慮した総合的評価 … 人件費の項目がない為、不可能

=> 以上の操作により離職率を下げたときの Achievement / Number of Employees で評価する

新規事業の利得の予測

改善策の検証と結果

`Overtime:No` の場合、離職(`Attrition`)は減るか?

検証

先ほど最適化したモデル(LDA) に `data[“OverTime”]=0` を行ったものを予測させ、`value_counts()` で `Attrition` がとる結果の個数を確認した。

結果

```
predict_model(tuned_model, data=Data_No_Overtime).value_counts()
Data={0:1233, 1:237}(19.2%), Data_No_OverTime={0:1395, 1:75}(5.4%)
```

=> 離職率は 19.2% から 5.4% に減少し、

時間外労働を無くすことで離職率の低下が見込まれることが分かった。

新規事業の利得の予測

改善策の検証と結果

`Overtime:No` の場合、成果(`Achievement`)も減るか?

検証

`DailyAchievement` を予測する回帰モデルを同様の手順で作成し、最適化したモデル (Model:Dummy, R2:-0.0089)

に `data[“OverTime”]=0` の処理を行ったものを予測させ、`DailyAchievement` の総和を、

より厳密には `0=Attrition`$_i$ を満たす i で $\sum_i$ `DailyAchievement`$_i$ を計算した。

結果

`Data=1001818, Data_No_OverTime=1250669`

=> 予想とは反対に、会社全体で 約 1.25倍 の成果が見込めるという結果となった。

これは時間外労働をしないほうが成果が出ることを意味するほか、

他カラムの値と相関があったことが考えられる。

新規事業の利得の予測

改善策の検証と結果

検証

`Overtime:No` とし、離職(`Attrition`)率を下げたときの DailyAchievement / Number of Employees ,

より厳密には `0=Attrition`$_i$ を満たす i で $\sum_i$ `DailyAchievement`$_i$ / i の個数 を評価する。

先ほど最適化した`DailyAchievement` を予測する回帰モデルに `data[“OverTime”]=0` を行ったものを予測させ、

`0=Attrition`$_i$ を満たす i で $\sum_i$ `DailyAchievement`$_i$ / i の個数 を計算した。

結果

```
Data=812.504, Data_No_OverTime=896.536
```

=> 従業員１人当たり１.10倍 の成果が見込めるという結果となった。

新規事業の利得の予測

改善策の提案(まとめ)

以上の検証より時間外労働(`Overtime`) を削減することにより、19.2% であった離職率は 5.4% まで減少し

以前より一人当たり 1.10倍 の成果が見込めるという結果が予測されたため、

会社の利益を上げるために時間外労働の削減を行うことを提案する。

参考文献

[1] kaggle : IBM HR Analytics Employee Attrition & Performance (online)
〈https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset〉(2024.1.16)

[2] 厚生労働省 : 雇用動向調査結果の概要 (online)
〈https://www.mhlw.go.jp/toukei/list/9-23-1c.html〉(2024.1.16)