**고급 통계적 머신러닝**

**기말 레포트**

**HR 데이터**

**2019021059 조윤영**

1. **데이터 소개 및 EDA**

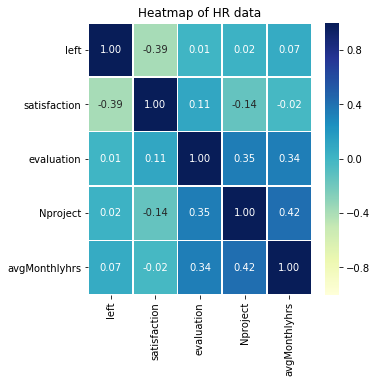
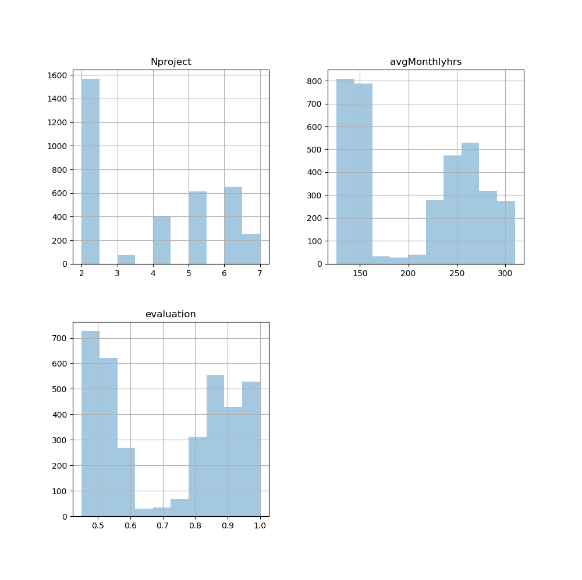
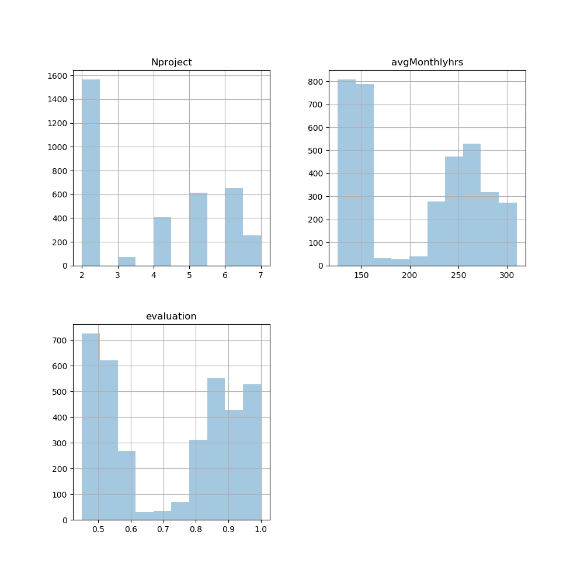
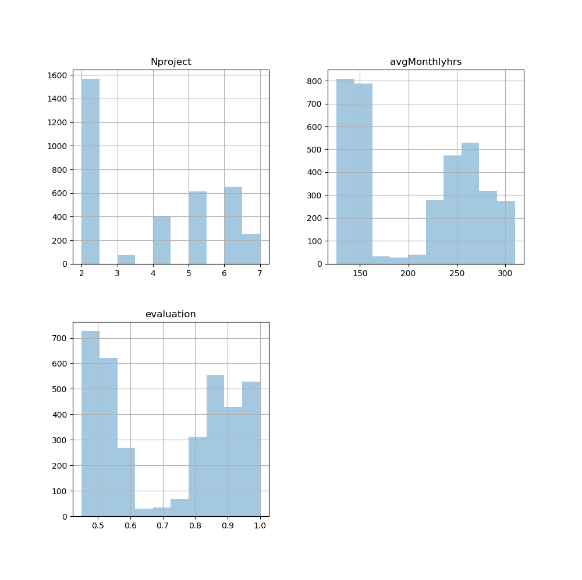
분석에 이용할 데이터는 한 회사의 HR 데이터로, 회사의 직원들의 이직 여부를 예측하기로 하였다. 여러 분석 방법을 모두 시도해본 후, 데이터에 가장 적합한 모델을 찾고 회사의 이직률을 줄이는 방법을 제안하고자 한다. HR 데이터는 전체 14999개의 obs를 가지고, 변수는 8개이다. 변수는 다음과 같다.

|  |  |  |
| --- | --- | --- |
|  | 변수명 | 변수 설명 |
| 반응변수 | Left | 0,1로 이루어진 binary 변수 |
| 수치형 변수 | satisfaction  evaluation  Nproject  avgMonthlyHrs  yrsAtCompany | 업무 만족도 0<X<1  업무 평가 0<X<1  연간 프로젝트 참여 횟수  월별 평균 근로시간(시간)  근로년수 |
| 범주형 변수 | promotion  salary | 최근 5년간 승진 여부 0,1  연봉 (low, medium, high) |

먼저 반응변수를 살펴보았을 때, 이직한 직원이 3571명, 이직하지 않은 직원이 11428명으로 아주 불균형한 데이터임을 확인할 수 있었다. 변수간 상관계수를 확인했을 때, left변수와 업무만족도가 -0.39의 음의 상관관계를 보였고, 업무평가, 연간 프로젝트 참여횟수, 월평균 근로시간이 각각 0.34이상의 양의 상관이 있음을 확인할 수 있었다(그림1).

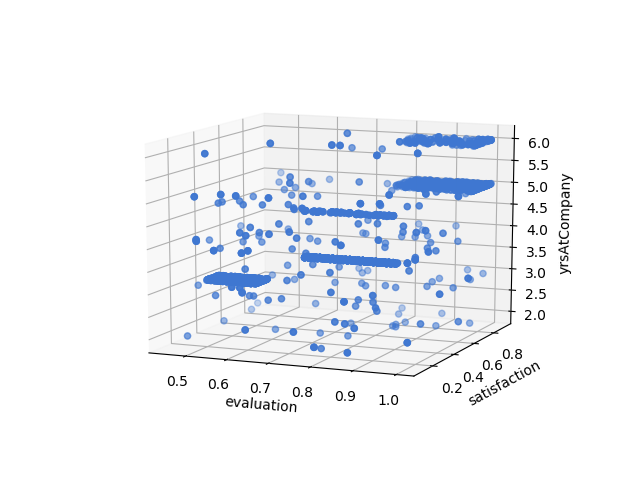
이때 이직한 직원들의 업무평가, 연간 프로젝트 참여횟수, 월평균 근로시간 세 변수의 히스토그램을 살펴보니 세 변수 모두 bimodal하며(그림2), 이를 둘로 나누어 모든 변수에서 Q1보다 작은 값을 가진 직원들과 모든 변수에서 Q3보다 큰 값을 가진 직원들로 나누어 보았다. 또한, 이직한 직원들의 속성을 더 살펴보고자 산점도를 그려보았는데, 이때 evaluation, Nproject, avgMonthlyhrs변수들은 상관계수가 높으므로 그 중 하나인 업무평가만을 이용하여 업무만족도(satisfaction)와 근속년수로 산점도를 그려보았다. 그러자 업무평가가 Q3보다 큰 그룹에서 만족도를 기준으로 두 개의 그룹으로 나뉘는 모습을 보여, 결과적으로 총 세 개의 클러스터가 확인되었다(그림3). 만족도는 낮으나 일한 기간은 중간이고 평가가 좋은 군집을 능력자 군집이라고 명명하고, 만족도도 중간, 일한기간도 적고 평가도 낮은 군집을 월급루팡이라고 명명하고, 만족도도 높고 일한기간도 길고 평가도 높은 군집을 워커홀릭이라고 명명하였다. 워커홀릭과 능력자 군집을 자세히 살펴보자, 이들의 80% 이상이 퇴사한 것으로 드러났다.

<그림2>



<그림1>

1. **SMOTE & ADASYN**



<그림3>

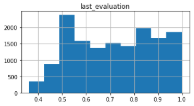
반응변수인 left변수의 분포는 이직한 직원이 3571명, 이직하지 않은 직원이 11428명으로 불균형한 데이터임을 확인할 수 있었다. 불균형한 데이터는 단순히 우세한 클래스를 택하는 모형의 정확도가 높아지는 문제점을 가지게 된다. 14999명중 전부가 이직하지 않을 것이라고 예측한다면 무려 11428/14999=0.76로 76%를 맞추게 되는 셈이다. 이를 방지하기 위해서 under sampling이나 over sampling을 한다. Under sampling의 경우 중요한 정보를 누락시킬 수 있는 위험이 있고, oversampling의 경우 소수 클래스를 복사하여 과적합의 문제가 생길 수 있다. 이 글에서는 over sampling을 진행하며, 그에 따른 과적합의 문제를 줄여주는 SMOTE와 ADASYN으로 줄여보고자 한다.

* SMOTE : SMOTE의 경우 데이터의 부분 집합은 소수 클래스로부터 예제로써 추출되고, 새로운 합성 유사 사례가 생성된다[[1]](#footnote-1).
* ADASYN : SMOTE의 경우 새로운 합성 유사 사례들은 부모 샘플들에 linearly correlated 되어 있는데, 이에 variance를 약간 더 해주면서 좀 더 현실적인 값들로 sampling 해주게 된다.

|  |  |  |  |
| --- | --- | --- | --- |
| SMOTE 결과 | | ADASYN 결과 | |
| Left=0 | Left=1 | Left=0 | Left=1 |
| 9129 | 9129 | 9129 | 9072 |

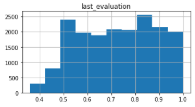
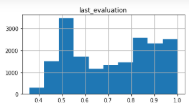
설명 변수 중 하나인 evaluation의 변화를 눈으로도 확인할 수 있었다(그림4).

원본 데이터



SMOTE

SMOTE(k\_neighbors=2



ADASYN

SMOTE(k\_neighbors=2

<그림4>

1. **KNN**

KNN을 이용해서 이직한 직원을 예측하는 모델링을 하고자 하였다. KNN은 k개의 가장 가까운 point들을 확인하고, 그 가까운 포인트들이 가장 많이 속하는 label로 지정한다. 예를 들어, 관측치1의 가장 가까운 관측치들이 이직이 3명 stayed가 2명이라면 이직으로 분류한다. 이 거리를 측정하는 데에 유클리디안, 맨하튼, minkowski 거리 등을 이용할 수 있다. 거리측정과 K를 지정하기 위해서 GridSearchCV를 이용하였다. 그 결과로 거리는 전부 유클리디안 거리를 이용하였으며 K는 oversampling방법과 표준화 여부에 따라 달라지는 것을 확인할 수 있었다. 거리를 구하기 위해 범주형 변수인 promotion과 salary변수는 이용하지 않았다.

Over sampling을 한 모델들과 그렇지 않은 모델들 중 오히려 over sampling을 하지 않은 모델이 오히려 그 정확도가 높음을 확인할 수 있었다. 또한 ADASYN 방법보다는 SMOTE방법이 조금 더 나은 모델을 보였다. Oversampling을 하지 않고 표준화를 한 데이터를 이용한 모델이 가장 높은 accuracy를 보임을 확인할 수 있었다. 이 데이터에서는 이직을 하는 직원, 즉 left가 1이 되는 것을 예측하는 것이 중요하므로, left의 precision도 함께 확인하였고, 이 역시 같은 모델이 가장 예측력이 좋음을 확인할 수 있었다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| KNN without oversampling | | | KNN with SMOTE | | KNN with ADASYN | | |
|  | **Std[[2]](#footnote-2)** | **original** | **Std** | **original** | | **Std** | **original** |
| K | 6 | 6 | 3 | 4 | | 3 | 3 |
| Precision | 0.96 | 0.91 | 0.90 | 0.89 | | 0.84 | 0.79 |
| Accuracy | 0.97 | 0.96 | 0.97 | 0.96 | | 0.95 | 0.94 |
| Confusion  Matrix |  |  |  |  | |  |  |
| Recall | 0.91 | 0.92 | 0.97 | 0.96 | | 0.98 | 0.98 |
| F1-score | 0.93 | 0.91 | 0.94 | 0.92 | | 0.90 | 0.88 |

1. **Logistic** **Regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Logistic | | Logistic with SMOTE | Logistic with ADASYN |
| Precision | | 0.59 | 0.49 | 0.45 |
| Accuracy | | 0.79 | 0.76 | 0.72 |
| Confusion  Matrix | |  |  |  |
| Recall | | 0.36 | 0.81 | 0.76 |
| F1-score | | 0.45 | 0.61 | 0.56 |

Logistic 모델에서는 범주형 변수(salary, promotion)들을 추가하여 모델을 적합 시켰다. L1규제화, L2 규제화를 주기 전에도 정확도는 0.79로 좋은 모델은 아니었다. Precision 또한 어떤 모델도 0.6을 넘기지 못하였다. KNN과 마찬가지로 오히려 Over sampling을 했을 때 정확도가 떨어졌다.

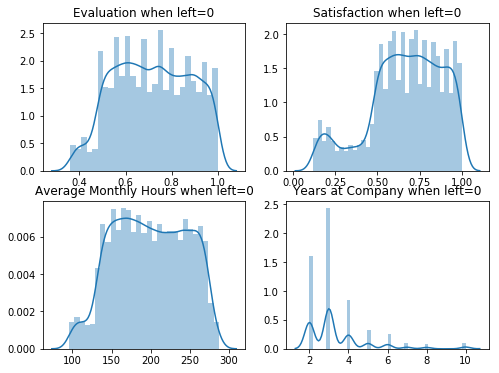
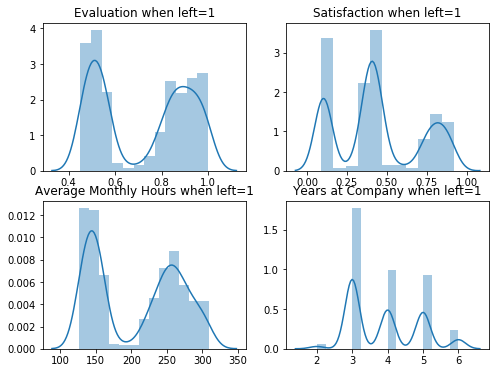
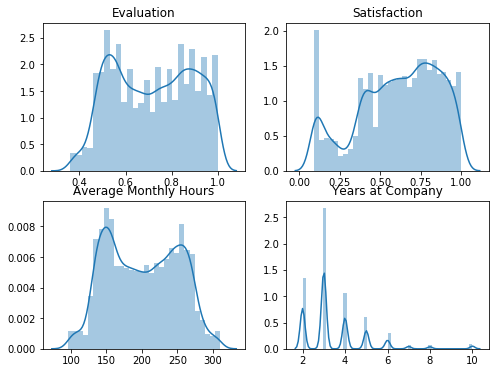
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Logistic | | Logistic with SMOTE | Logistic with ADASYN |
| Penalty | | L2 | L2 | L2 |
| C(=) | | 10 | 0.1 | 10 |
| Precision | | 0.59 | 0.49 | 0.45 |
| Accuracy | | 0.79 | 0.76 | 0.73 |
| Confusion  Matrix | |  |  |  |
| Recall | | 0.36 | 0.82 | 0.77 |
| F1-score | | 0.45 | 0.61 | 0.57 |

L1규제화와 L2규제화를 모두 해보았을 때, 모든 데이터에서 L2규제화가 가장 나은 것으로 나왔다. 이때 coefficient가 0으로 간 변수는 avgMonthlyhrs로 앞선 EDA에서 상관계수가 아주 없지는 않는 변수들이 있었기에 납득이 가는 결과라고 할 수 있다. 다만, 규제화를 하더라도 모델의 정확도는 거의 나아지지 않았으며, 어느 모델이 낫다고 말하기 어렵다.

선형 분류가 어려운 것으로 생각되어 앞에서 EDA한 결과를 바탕으로 워커홀릭, 능력자, 베테랑 군집으로 나누어 모델을 적합하자 거의 완벽에 가까운 예측력을 보여주었다.

1. **LDA & QDA**

전체 데이터



Left=0

Left=1

<그림 5>

정규분포 가정도 성립하지 않고 등분산 가정도 성립하지 않는다. 따라서 LDA 뿐만 아니라 QDA도 적합한 모델이 아니다. X가 서로 독립도 아니기에 Gaussian Naïve Bayes 모형도 적합하지 않다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | LDA | | LDA with SMOTE | LDA with ADASYN |
| Precision | | 0.47 | 0.46 | 0.48 |
| Accuracy | | 0.76 | 0.74 | 0.75 |
| Confusion  Matrix | |  |  |  |
| Recall | | 0.25 | 0.78 | 0.89 |
| F1-score | | 0.33 | 0.58 | 0.62 |

LDA는 독립변수가 정규분포를 따르고 n이 작을 때 로지스틱보다 좋은 결과를 보여주는데, 이 데이터는 n이 1만개가 넘으므로 n이 충분하다. 실제로 Logistic 모델보다 좋은 결과를 주지는 못했다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | QDA | | QDA with SMOTE | QDA with ADASYN |
| Precision | | 0.76 | 0.68 | 0.46 |
| Accuracy | | 0.91 | 0.88 | 0.72 |
| Confusion  Matrix | |  |  |  |
| Recall | | 0.89 | 0.89 | 0.94 |
| F1-score | | 0.82 | 0.77 | 0.61 |

예상대로 그렇게 좋은 결과를 보여주지는 못했다. 그래도 LDA보다는 좋은 결과를 보여줬는데, 정규분포가정은 성립하지 못하지만 등분산 가정이 없기 때문이라고 추측할 수 있다.

1. **NB**

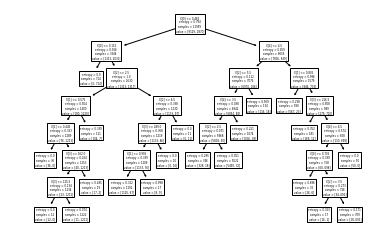
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Gaussian NB | | NB with SMOTE | NB with ADASYN |
| Precision | | 0.55 | 0.33 | 0.34 |
| Accuracy | | 0.80 | 0.55 | 0.56 |
| Confusion  Matrix | |  |  |  |
| Recall | | 0.73 | 0.93 | 0.95 |
| F1-score | | 0.63 | 0.49 | 0.50 |

좋은 결과를 얻지 못하였다.

1. **CART**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CART without oversampling | | | CART with SMOTE | | CART with ADASYN | | |
|  | **Std** | **original** | **Std** | **original** | | **Std** | **original** |
| Criterion | entropy | Entropy | Entropy | Entropy | | Entropy | Entropy |
| Max\_depth | 7 | 7 | 20 | 17 | | 20 | 20 |
| Max\_leaf\_nodes | 20 | 20 | 300 | 300 | | 300 | 300 |
| Min\_samples\_leaf | 10 | 10 | 10 | 10 | | 10 | 10 |
| Precision | 0.99 | 0.99 | 0.92 | 0.95 | | 0.89 | 0.88 |
| Accuracy | 0.98 | 0.98 | 0.97 | 0.97 | | 0.96 | 0.96 |
| Confusion  Matrix |  |  |  |  | |  |  |
| Recall | 0.92 | 0.92 | 0.95 | 0.94 | | 0.97 | 0.96 |
| F1-score | 0.95 | 0.95 | 0.93 | 0.94 | | 0.92 | 0.92 |

의사결정나무의 경우 over sampling과 표준화를 기준으로 총 6개의 모델을 비교해보았다. 각각은 전부 gridSearchCv로 최적의 초모수로 설정한 상태이다. Oversampling을 한 데이터들은 max\_depth가 거의 최대치로, 나무의 깊이가 깊고, 노드들도 많아 복잡한 모델이 적합되었다. 가장 좋은 모델인 over sampling과 표준화를 하지 않은 모델의 그래프를 그려보았다. 그래프를 살펴보면, 가장 먼저 satisfaction변수를 이용하여 둘로 나뉘었다. 왼쪽 가지에서는 계속해서 satisfaction변수와 evaluation변수로 나눠가는 반면, 오른쪽에서는 avgMonthlyhrs 변수로 나누기 시작했다. 왼쪽 가지를 살펴보면, evaluation, avgMonthlyhrs, Nproject의 가 각각 Q1에 가까운 값들보다 낮은 것으로 나누기 시작하는데, 이는 EDA에서 살펴보았던 월급루팡 군집에 해당하는 값이다. 따라서, 앞서 보았던 월급루팡, 워커홀릭, 능력자 군집들이 실제로 분류에 영향을 미치고 있음을 확인할 수 있었다.



<그림 6>

1. **SVM**

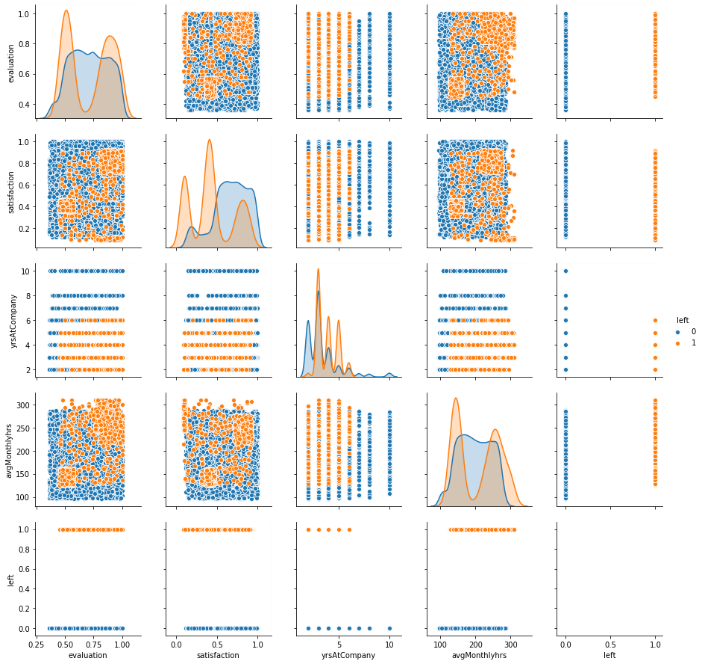
SVM은 분류할 변수를 나누는 경계선 중 관측치와의 폭이 가장 넓은 것을 찾아낸다. 이 때 초모수는 C인데, C가 크면 목적함수가 커져 폭이 좁아지고, C가 작으면 폭이 넓어진다. 이 때문에 C가 지나치게 클 경우 과적합의 위험이 있고, 너무 작을 경우 편이가 발생한다. 여러가지 C값과, SMOTE, ADASYN을 진행한 데이터를 이용하여 모델을 적합시켜 가장 예측력이 좋은 모델을 찾고자 했다. MOTE와 ADASYN의 경우 컴퓨팅 시간이 너무 오래 걸리는 문제로 데이터를 5천개만을 가지고 진행하였다. 또한, salary 변수를 one-hot encoding을 진행해서 모델에 추가하였다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SVM | | | | | |
| C | 1 | 100 | 500 | 1000 |
| Precision | 0.54 | 0.57 | 0.57 | 0.58 |
| Accuracy | 0.78 | 0.79 | 0.79 | 0.80 |
| Confusion  Matrix |  |  |  |  |
| Recall | 0.25 | 0.49 | 0.48 | 0.53 |
| F1-score | 0.34 | 0.52 | 0.52 | 0.55 |

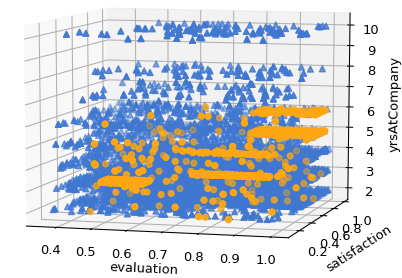
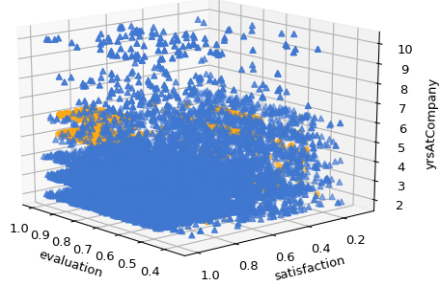
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SVM with SMOTE | | | | | |
| C | 1 | 100 | 500 | 1000 |
| Precision | 0.55 | 0.53 | 0.50 | 0.49 |
| Accuracy | 0.80 | 0.76 | 0.75 | 0.75 |
| Confusion  Matrix |  |  |  |  |
| Recall | 0.89 | 0.85 | 0.81 | 0.80 |
| F1-score | 0.68 | 0.65 | 0.62 | 0.61 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SVM with ADASYN | | | | | |
| C | 1 | 100 | 500 | 1000 |
| Precision | 0.49 | 0.49 | 0.48 | 0.48 |
| Accuracy | 0.75 | 0.76 | 0.75 | 0.75 |
| Confusion  Matrix |  |  |  |  |
| Recall | 0.89 | 0.88 | 0.86 | 0.88 |
| F1-score | 0.63 | 0.63 | 0.62 | 0.62 |

Parameter를 여러 개로 시도해 보았지만, 원 데이터, SMOTE, ADASYN 그 어느 것도 SVM에서 좋은 성능을 보여주지 못했다. 그래도 그 중에서 가장 좋은 모델을 골라보자면 over sampling을 하지 않은 모델에서 보라색으로 칠해진 모델이다. 이렇게 예측력이 안 좋은 그 이유를 알아보기 위해 다시 한번 변수들의 scatter plot을 그려 보았다. 확실히 left 변수를 구분하는 선형적인 구분선을 찾기에는 어려워 보이는 분포를 가지고 있었다(그림7).



left가 주황색, evaluation, satisfaction, yrsAtCompany, avgMonthlyhrs



<그림 7> left가 주황색

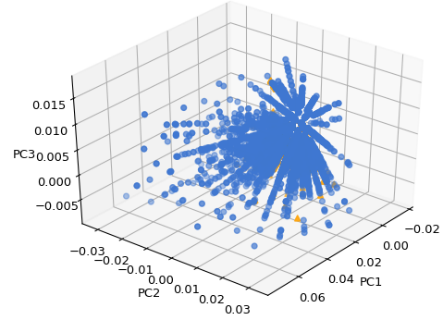
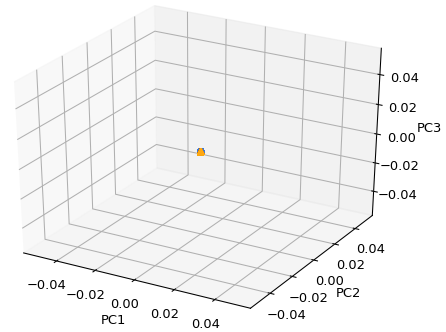
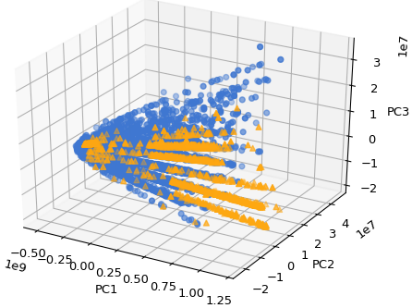
선형적인 방법으로는 분류가 불가능해 보이기에, 이를 해결하기 위해 비선형 분류모형인 Kernel SVM을 이용해보았다. Kernel SVM은 가상의 변수를 추가해 선형적으로는 분류가 불가능한 데이터를 완전하게 나누는 선형평면을 만들어 낸다. 비선형 SVM에는 gamma라는 초모수가 추가되는데 이는 kernel coefficient로 하나의 데이터 샘플이 영향력을 행사하는 거리를 결정한다. Gamma가 클수록 한 관측치의 영향력을 행사하는 거리가 짧아지고, 작을수록 길어진다. 따라서 gamma가 커지면 과적합의 위험이 있다. 각 데이터별 가장 알맞은 C값과 gamma값을 정하기 위해 역시 gridSearchCv를 이용하였다. 결과는 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Kernel SVM | | Kernel SVM  with SMOTE | Kernel SVM  with ADASYN |
| C | | 5000 | 5000 | 5000 |
| gamma | | 0.01 | 0.01 | 0.01 |
| Precision | | 0.89 | 0.89 | 0.83 |
| Accuracy | | 0.96 | 0.96 | 0.94 |
| Confusion  Matrix | |  |  |  |
| Recall | | 0.96 | 0.95 | 0.97 |
| F1-score | | 0.92 | 0.92 | 0.89 |

Linear SVM보다는 더 나은 모델을 보여주었다. 그 중에서도 over sampling을 하지 않은 데이터로 적합시킨 kernel SVM이 가장 나은 모델임을 알 수 있었다.

1. **차원축소**

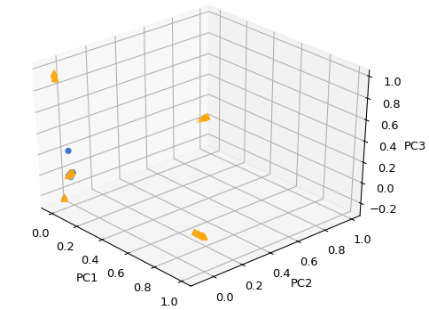
앞선 EDA에서 살펴보았을 때, avgMonthlyhrs, evaluation, Nproject가 높은 상관관계는 아니었지만 상관 관계가 있었기 때문에 PCA를 통해 변수를 줄인 후 모델에 적합해보고자 하였다. Kernel 차원축소에서는 gamma라는 초모수가 있는데, 이는 SVM에서의 gamma와 마찬가지로 gamma가 커질수록 개체의 영향이 커진다고 이해할 수 있다. PCA를 한 후 Logistic 모델과 LDA모델을 적합시켜 보았으나, 역시 결과는 0.77로 비슷하게 나왔으며 정확도 면에서 나아지지 않았다. Kernel PCA를 한 뒤, Kernel SVM을 적합시켜 보았다. Kernel 함수는, plot을 그려본 결과 rbf가 그나마 분류하기에 눈에 보이는 결과일 것이라고 예상하여 rbf로 모델링을 하였다(그림8).



**sigmoid**

**polynomial**

**cosine**



**rbf**

<그림8>

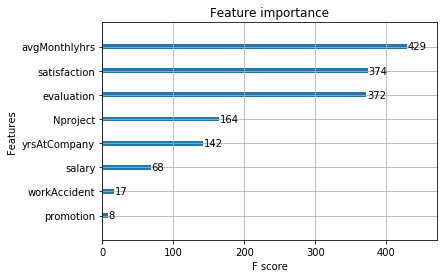
하지만 이 역시 0.77의 정확도를 보이며 의 confusion matrix 형태를 보였다. 이 때, PCA kernel 함수를 polynomial로 수정해서 다시 Kernel SVM에 적합 시켜본 결과, 정확도 0.93과 의 confusion matrix를 보이며 radial basis function로 PCA를 한 것 보다는 훨씬 좋은 결과를 얻었다. 하지만 PCA를 한 모델과 안 한 모델 중에서는 PCA를 안 한 모델이 더 좋은 결과를 주었다.

1. **Ensemble**

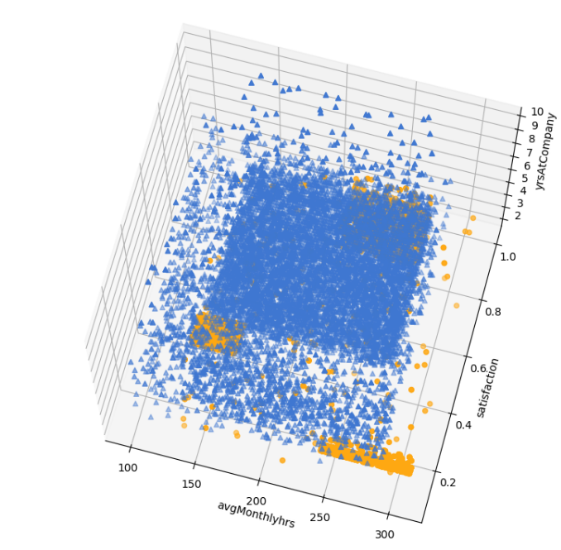
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Voting | Bagging | ADABOOST | GBM | XGboost |
| Hyper- parameter | Logistic random forest, svm, | max\_depth 5  max\_samples 0.5  n\_estimators 500  max\_features 0.5 | n\_estimator 1000  max\_depth 5  learning rate 0.5 | n\_estimator 500  max\_depth 5  learning rate 0.1 | colsample\_bytree 0.75  learning\_rate 0.5  max\_depth 5  n\_estimators 100 |
| Precision | 1 | 0.99 | 0.98 | 0.98 | 0.98 |
| Accuracy | 0.85 | 0.98 | 0.99 | 0.99 | 0.98 |
| Confusion  Matrix |  |  |  |  |  |
| Recall | 0.35 | 0.91 | 0.97 | 0.97 | 0.97 |
| F1-score | 0.52 | 0.94 | 0.97 | 0.97 | 0.98 |

앙상블 모델 중에서는 Voting이 가장 안 좋은 결과를 주었고, ADAboost, GBM, XGboost 모두 비슷한 결과를 주었다. 이는 voting에 성능이 좋지 않은 로지스틱 모델이 포함되어 그런 것이라고 추측할 수 있다. 모두 gridSearchCv를 이용하여 초모수를 조정하였다. Bagging의 경우 oob자료로 score를 구했을 때도 거의 비슷한 정확도를 보여주었다. 이 중에 가장 빠른 XGboost가 가장 낫다고 말할 수 있다.

변수의 중요도를 보여주는 importance plot을 그려보았을 때(그림9) avgMonthlyhrs가 큰 중요도를 가졌다. avgMonthlyhrs는 left와의 상관계수가 매우 낮았던 것을 생각하면, 의외의 결과일 수도 있다. 이 의문점을 해결하고자 evaluation이 아닌 avgMonthlyhrs를 이용한 3차원 산점도를 그려보았다(그림10). 그러자 앞서 evaluaiton에서 보았던 것보다 더 명확하게 left와 stay가 나뉘어지는 모습을 확인할 수 있었다. 그 뒤에 다른 변수들을 예상할 수 있던 순위임을 확인할 수 있었다.



<그림9>



<그림10>

1. 결론

이직 의도를 선행하는 연구들을 조사해본 결과, 이직의도에 영향을 주는 요소에는 경력학습, 고용 가능성, 조직몰입, 감정노동, 성별, 직무스트레스, 승진가능성, 직무만족 등이었다. 이 요소들 중에 가지고 있는 데이터는 직무만족과 승진여부 뿐이었다. 또한, 이직은 산업별로 원인에 큰 차이가 있는데 반면 이 데이터로는 어떤 산업인지 알 수 없었다는 한계점이 있었다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | KNN | CART | Kernel SVM | Bagging | ADABOOST | GBM | XGboost |
| Precision | 0.96 | 0.99 | 0.89 | 0.99 | 0.98 | 0.98 | 0.98 |
| Accuracy | 0.97 | 0.98 | 0.96 | 0.98 | 0.99 | 0.99 | 0.98 |
| Con- fusion  Matrix |  |  |  |  |  |  |  |
| Recall | 0.91 | 0.92 | 0.96 | 0.91 | 0.97 | 0.97 | 0.97 |
| F1-score | 0.93 | 0.95 | 0.92 | 0.94 | 0.97 | 0.97 | 0.98 |

데이터 불균형을 over sampling을 통해 해결하려고 시도하였으나, 대부분의 분석기법에서 오히려oversampling 이후에 예측력이 더 낮아졌다. 이직을 예측하는 것을 분석 목표로 삼았을 때에는 XGboost를 이용하는 것이 빠르고 정확도를 높이는 방법이다.

회사의 입장에서 능력자 집단과 워커홀릭 집단의 퇴사원인규명이 중요하기 때문에 이를 위해 추가적인 EDA를 해 본 결과, 이 집단들의 salary, promotion이 일반 회사원들과 큰 차이가 없는 수준이었다. 이는 XGboost의 important plot에서 두 변수가 낮은 중요도를 차지한 것과 연결된다. 또한, 능력자 집단은 연간 프로젝트 개수가 많을수록 퇴사율이 높았는데, 프로젝트가 능력자 집단과 워커홀릭 집단에 편항 되었을 가능성도 있다. 능력자집단과 워커홀릭 집단이 월급루팡 집단을 이끌 수 있게 승진이나 임금에 차이를 두어 동기부여를 해주는 것이 좋겠다.

1. <https://sherry-data.tistory.com/22> [↑](#footnote-ref-1)
2. stardardization [↑](#footnote-ref-2)