Term Project Report

이종근

Department of Industrial Management Engineering, Korea University

IMEN321

Professor Jun Geol Baek

June 21, 2023

〈Overview〉

I. Motivation

1. 데이터셋 소개

Diabetes Dataset (https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset)

데이터마이닝 수업에서 현재까지 주로 Classification 기법을 다뤘기 때문에 이를 효과적으로 이용할 수 있는 데이터를 선정하기 위해서는 종속변수를 뜻하는 Target이 Categorical 변수이며, 독립변수를 뜻하는 Factor들이 여러 개로 구성된 데이터셋을 선정해야 한다. 위의 데이터는 Target인 Diabetes 유무를 8개의 Numeric의 Factor로 예측할 수 있게 되어있어 적합하다고 판단하였다. 또한, 당뇨병은 제 1형 당뇨병 (인슐린 의존성 당뇨병)과 제 2형 당뇨형(인슐린 비의존성 당뇨병), 임신성 당뇨병으로 나눌 수 있다. 먼저 제 1형 당뇨병은 췌장의 베타세포가 파괴되어 인슐린이 나오지 않는 병을 뜻한다. 이는 선천적인 질병으로 인한 것으로, 인슐린 분비 정도를 측정하면 진단이 가능하다. 제 2형 당뇨병은 잘못된 생활 습관에서 비롯한 비만 또는 과체중 증상으로 인해 인슐린 저항성이 커지며 발생하는 질병이다. 마지막으로 임신성 당뇨병은 태아에서 분비되는 호르몬에 의해 인슐린의 기능이 떨어져 발생하는 당뇨병이다. 이러한 세 가지 당뇨병을 아래 데이터셋의 독립변수들로 예측할 수 있을 것 같아 데이터를 선정하였다.

또한 데이터가 가지는 의미에 대해 생각해봤을 때, 당뇨병은 혈액 속의 포도당이 세포 속으로 들어가 에너지원으로 이용되지 못해 혈당이 비정상적으로 올라가는 질환이다. 당뇨병에 걸리게 된다면 망막병증, 신경병증, 신장병증, 동맥경화로 인한 뇌졸중, 협심증, 심근경색증과 같은 합병증이 발생해 개인에게 큰 피해를 초래하기 때문에 조기에 진단하여 예방하는 것이 중요한 질병이다. 이러한 특성을 가졌기 때문에 높은 "예측 정확도"를 갖는 모델을 만들어 의사의 판단을 보조하는 역할로 사용될 수 있다면 사회적으로 비용을 줄일 수 있다.

2. 데이터셋 설명

분석에 이용한 당뇨병 데이터는 존스홉킨스 대학의 Vincent G. Sigillito 교수 연구실에서 수집한 데이터로, Fima Indian 혈통을 가진 21 세 이상의 여성 768 명으로 구성되었으며, 각 개인의 8 가지 특성과 당뇨병 결과로 구성되어 있다. 데이터 특성 설명은 아래와 같다.

독립변수

1. Pregnancies: Number of times pregnant
2. Glucose: Plasma glucose concentration a 2hours in an oral glucose tolerance test
3. BloodPressure: Diastolic blood pressure (mm Hg)

4. SkinThickness: Triceps skin fold thickness (mm)
5. Insulin: 2-Hour serum insulin (mu U/ml)
6. BMI: Body mass index (weight in kg/(height in m)^2)
7. DiabetesPedigreeFunction: Diabetes pedigree function
8. Age: Age (years)

종속변수

9. Outcome: Class variable (not diabetes: 0 or diabetes: 1)

II. Exploratory Data Analysis

1. Data Type 및 Null Data 확인

#	Column	Null Count	Dtype
0	Preganacies	768 non-null	Int64
1	Glucose	768 non-null	Int64
2	BloodPressure	768 non-null	Int64
3	SkinThickness	768 non-null	Int64
4	Insulin	768 non-null	Int64
5	BMI	768 non-null	Float64
6	DiabetesPedigreeFunction	768 non-null	Float64
7	Age	768 non-null	Int64
8	Outcome	768 non-null	Int64

본 데이터 셋을 여러 모델에 학습하기 위해서는 Null Type의 Data가 있는지, 데이터 타입은 무엇인지 확인하여야 한다. 혹시 Categorical Data가 있다면 1-of-C Coding 같은 방법을 이용하여 전처리를 해주어야 하기 때문이다. 하지만 확인 결과, Null Data가 없고 모두 수치형 변수인 것을 확인할 수 있었다.

2. 단변량통계

	Preg	Glucose	BlodPres	SknThicknes	Insul	BMI	DiaPediFunc	Age	Outcome
Count	768.00	768.00	768.00	768.00	768.00	768.00	768.00	768.00	768.00
Mean	3.85	120.89	69.11	20.54	79.80	31.99	0.47	33.24	0.35
Std	3.37	31.97	19.36	15.95	115.24	7.88	0.33	11.76	0.48
min	0.00	0.00	0.00	0.00	0.00	0.00	0.08	21.00	0.00
25%	1.00	99.00	62.00	0.00	0.00	27.30	0.24	24.00	0.00
50%	3.00	117.00	72.00	23.00	30.50	32.00	0.37	29.00	0.00
75%	6.00	140.25	80.00	32.00	127.25	36.60	0.63	41.00	1.00
max	17.00	199.00	122.00	99.00	846.00	67.10	2.42	81.00	1.00

본 데이터 셋에 대한 기본 통계량을 조사해보았다.

- Pregnancies : 임신 횟수를 나타내는 변수로, 평균이 3.85인데 표준편차가 3.37를 보이고 있고 최댓값은 17회로 확인되는 것을 보아 편차가 크고, 이상치가 있는 것을 확인할 수 있었다.
- Glucose, BloodPressure, BMI, Age : 평균과 표준편차가 적당한 비율을 유지하고 있어 노말한 분포를 갖는 데이터로 예상된다.
- SkinThickness : 피부 두께를 나타내는 변수로, 평균이 20.54인데 표준편차가 15.95를 보이고 있고 최댓값은 99로 이상치가 존재하는 것을 확인할 수 있다.
- Insulin : 체내 인슐린 수치를 나타내는 변수로, 평균이 79.80인데 표준편차가 자그마치 115.24인 것을 보아 엄청난 편차를 가진 데이터 분포를 가졌다는 것을 예상할 수 있다. 또한 Q3이 127.25인데 최대치가 846인 것을 보아 이상치가 존재하는 것을 확인할 수 있다.
- Outcome : 평균이 0.35인 것을 보아, 당뇨병 클래스가 적고 정상 클래스가 많은 불균형 클래스 분포를 가진 데이터셋임을 확인할 수 있다.

추가적으로 표로 확인하는 것에 더해 시각화하여 확인한다면 더욱 데이터셋의 분포를 확실하게 이해할 수 있을 것 같아 그림1에 시각화하여 나타내었다. Distplot을 확인해보았을 때 데이터셋이 전반적으로 Skewness와 Kurtosis가 높은 것으로 보여서 추가적으로 관찰하기로 정했다. 또한, Boxplot을 살펴보았을 때 Outlier들이 많이 존재하는 것을 확인할 수 있었다. 마지막으로, Probability Plot을 통해 보았을 때 데이터들이 대체로 정규성을 만족하지 않음을 확인할 수 있다.

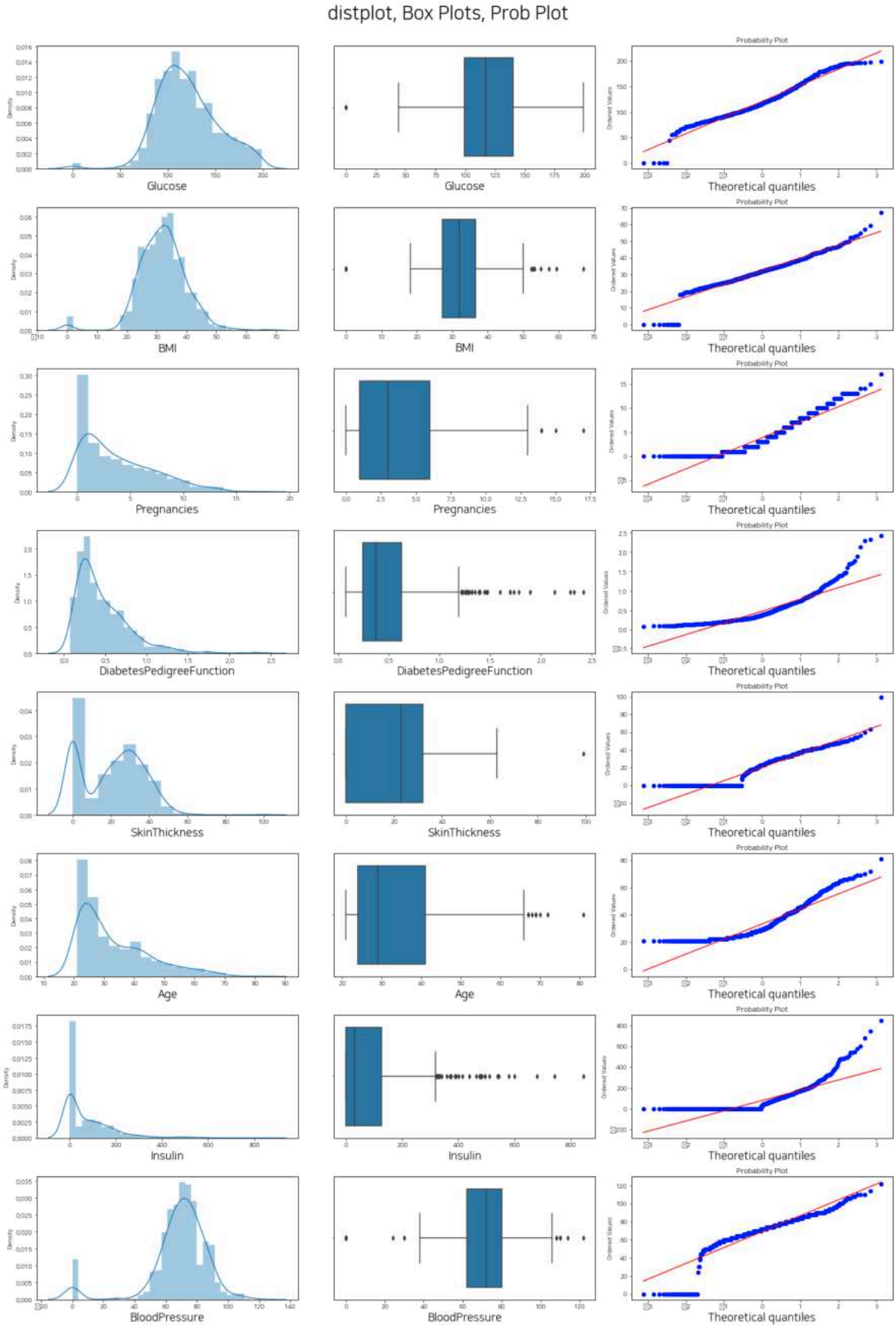
distplot, Box Plots, Prob Plot
Glucose
Glucose
Probability Plot
Theoretical quantiles
BMI
BMI
Probability Plot
Theoretical quantiles
Pregnancies
Pregnancies
Probability Plot
Theoretical quantiles
DiabetesPedigreeFunction
DiabetesPedigreeFunction
Probability Plot
Theoretical quantiles
SkinThickness
SkinThickness
Probability Plot
Theoretical quantiles
Age
Age
Probability Plot
Theoretical quantiles
Insulin
Insulin
Probability Plot
Theoretical quantiles
BloodPressure
BloodPressure
Probability Plot
Theoretical quantiles

그림 1: 단변량통계

3. Skewness & Kurtosis

	skewness	kurtosis
Age	1.13	0.64
BMI	-0.43	3.29
BloodPressure	-1.84	5.18
DiabetesPedigreeFunction	1.92	5.59
Glucose	0.17	0.64
Insulin	2.27	7.21
Outcome	0.64	-1.60
Pregnancies	0.90	0.16
SkinThickness	0.11	-0.52

Skewness 는 왜도를 뜻하는 단어로써, Normal Distribution 에서 왜곡 정도를 뜻한다. 이는 데이터 분포에서 대칭성의 정도를 나타내는 척도다. Skewness 가 -0.5 ~ +0.5 일 때 데이터는 대칭적이라 할 수 있고, -1 ~ -0.5, +0.5 ~ +1.0 는 용인 가능한 수준으로 치우쳐져 있다고 할 수 있고, -1 보다 작거나 +1 보다 크면 상당히 치우쳐져 있다고 할 수 있다. 위의 데이터에서는 BloodPressure 가 -1.84, Age 가 1.13, DiabetesPedigreeFunction 이 1.92, Insulin 이 2.27 로 상당히 치우쳐져 있는 변수라는 것을 확인할 수 있다.

Kurtosis 는 첨도를 뜻하는 단어로써 Distribution Graph 에서 꼬리 부분에 관한 것이다. 이는 간단하게, 분포에 존재하는 Outlier 를 뜻한다고 볼 수 있다. Kurtosis 가 높을 때 (Kurtosis > 3) Leptokurtic 이라고 하며 데이터가 두꺼운 꼬리나 Outlier 를 가지고 있다고 볼 수 있다. 반대로 Kurtosis 가 낮으면 (Kurtosis < 3) 분포는 데이터가 얇은 꼬리나 Outlier 를 가지고 있지 않다고 할 수 있다. 위의 데이터에서는 BMI, BloodPressure, DiabetesPedigreeFunction, Insulin 의 변수가 Leptokurtic 상태인 것을 확인할 수 있다.

3. Correlation

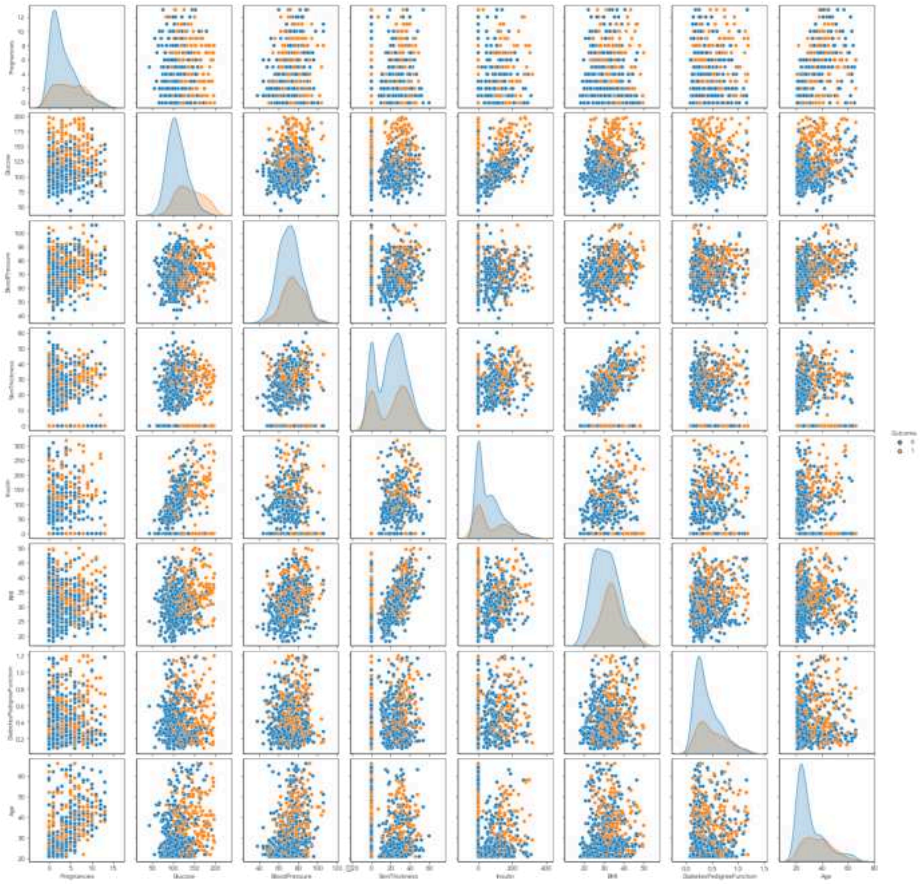

그림 2 : 변수들 사이의 PairPlot

변수들 사이의 상관관계를 확인해보기 위해 PairPlot 을 시각화하여 그림 2 로 나타내었다. 하지만, 데이터 인스턴스들이 많아 한 눈에 확인하기 어려운 점이 있어 정량적으로 확인할 수 있는 Heatmap 을 시각화하여 그림 3 으로 나타내었다.

상관계수는 -1 에서 +1 의 값을 가지는데, 1 에 가까울수록 높은 양의 상관관계를 가진다고 할 수 있다. 통상적으로 +0.3 보다 크거나 -0.3 보다 작을 때 상관관계를 갖는다고 할 수 있다. 본 데이터셋에서 상관관계를 갖는 조합을 나열한 결과는 아래와 같다.

- Age & Pregnancies (상관계수 : +0.55): 임신은 평균 10 달의 기간을 통해서 한 주기가 지나가기 때문에 당연한 상관관계를 갖는 것을 확인할 수 있다.
- Age & BloodPressure (상관계수: +0.35): 노화로 인한 대동맥 경화가 발생할 수 있다. 이는 ‘노인성 고혈압’이라고 칭하는데, 이러한 요인 때문에 양의 상관관계를 갖는 것으로 추측할 수 있다.

- BloodPressure & BMI (상관계수: +0.30): 지방이 많으면 그 지방조직에 공급되어야 할 여분의 혈액이 필요하고, 또 비만에 따른 신경 호르몬의 변화가 고혈압을 일으키도록 유도하기 때문에 양의 상관관계를 가지는 것으로 추측할 수 있다.

- BMI & SkinThickness (상관계수: +0.39): SkinThickness 는 상완삼두근 부위의 피부 주름 두께를 나타내는 지표로, 체지방량을 추정하는 데 사용되는 지표기 때문에 BMI 와 양의 상관관계를 가지는 것으로 추측할 수 있다.

- SkinThickness & Insulin (상관계수: +0.49): 전혀 예상하지 못했던 지표로써, SkinThickness 가 높으면 체지방량이 높아 인슐린 분비를 촉진한다고 추측할 수 있다.

그림 3: 변수들 사이의 Heatmap

III. Data Preprocessing

1. 이상치 제거

II-2, 3에서 각 특성의 평균에 비해 큰 표준편차 혹은 큰 최댓값을 갖는 것을 확인하여 이상치 제거 과정이 필요함을 확인할 수 있었다. 일반적으로 이상치 제거는 Q1 - 1.5 * IQR 보다 작은 값 또는 Q3 + 1.5 * IQR보다 큰 값 대상으로 전처리를 진행한다. 어느 한 특성이라도 Outlier 값이 존재한다면 값을 제거하는 기준으로 해당 변수의 수를 확인해보니 총 129개가 존재하였다. 데이터셋의 크기가 매우 적은데, 이러한 많은 데이터를 제거하게 된다면 모델 성능 저하로 귀결될 수 있다고 생각하여 Q1 - 2 * IQR 보다 작거나 Q3 + 2 * IQR 보다 큰 값을 대상으로 전처리 과정을 진행하였다. 하지만 이때, Insulin 변수는 당뇨병과 직접적으로 관련되는 변수라고 생각하여 Q1 - 3 * IQR, Q1 + 3*IQR 조건을 이용하여 전처리를 진행했다. 총 76개의 데이터가 제거되었다.

2. 데이터 스케일링

II-2의 단변량통계에서 특성간 스케일 차이가 심하게 존재하는 것을 확인할 수 있었다. 원래라면 데이터 특성, 모델의 특성에 따라 전처리를 진행하여야 하지만 해당 데이터는 여러 모델에 골고루 쓰일 뿐만 아니라 이후 과정에서 모델간 비교도 진행하기 때문에 변인을 통제하여 같은 데이터셋을 이용하는 게 타당하다는 판단을 하였다. Sclaer 중에서는 조교님께서 실습 시간에 효과적이라고 알려주신 MinMaxScaler를 이용해 Scale 통합을 진행하였다.

3. 학습 데이터셋 / 테스트 데이터셋 분리

이상치 제거 후의 데이터 수는 692개로 많지 않기 때문에 적절한 수의 학습 데이터를 확보하지 않고, 검증 데이터와 테스트 데이터에 많은 비율을 할애하게 된다면 모델의 성능이 예측에 충분하지 않을 것으로 예상되어 전체의 70%의 데이터를 학습 데이터로 선정하였다. 이후 검증 데이터셋의 분류 여부에 대해 고민한 결과, 사람의 생명과 관련된 분류 문제이기 때문에 모델 성능을 증가시키기 위한 검증데이터를 확보하기보다는 Cross-Validation 기법을 통해 그 역할을 대신하고, 테스트 셋을 최대한 확보하여 충분한 테스트 과정을 거쳐 신뢰성 있는 결과를 내는 것이 추가적인 위험을 감소시킬 수 있다고 생각하여 전체의 30%를 선정하였다. 즉, 이러한 이유로 학습:검증:테스트 데이터셋 비율을 70:0:30으로 선정하였다.

IV. Predictive Modeling

1. Logistic Regression

Variable	P-value
Constant	0.0000
Pregnancies	0.1194
Glucose	0.0000
BloodPressure	0.4481
SkinThickness	0.5856
Insulin	0.8554
BMI	0.0019
DiabetesPedigreeFunction	0.0338
Age	0.0871

학습 결과 유의수준 0.05에서 유효한 변수는 Glucose, BMI, DiabetesPedigreeFunction로 총 3가지를 확인할 수 있었다.

- Glucose: 당뇨병은 혈액 내 고혈당을 보이는 증상이기 때문에 포도당 주입 2시간 후 혈당 농도를 나타내는 Glucose는 당연히 유효하게 작용하는 변수라고 생각할 수 있다.
- BMI: 당뇨병의 원인은 잘못된 생활 습관으로 비롯한 비만 또는 과체중 증상으로 인해 인슐린 저항성이 커지며 발생하는 질병이기 때문에 당연히 유효하게 작용하는 변수라고 판단할 수 있다.
- DiabetesPedigreeFunction: 당뇨병 유전병 지수를 나타내는 지표기 때문에 당연하게 제 1형 당뇨병과 관련이 깊어 예측에 유의하게 작용하는 변수임을 확인할 수 있다.

하지만, Insulin 변수의 P-value의 값이 제일 높은 것이 분석 결과 가장 놀라운 점이었다. 일반적으로 생각하였을 때 당뇨병은 인슐린 저항성이 커지게 되어 발생하는 질병이기 때문에 당연히 유의한 결과가 도출될 것이라고 예상하였지만 예상밖의 결과가 나오게 되었다. 이러한 결과가 나오게 된 이유에 대해 생각해보자면 본 데이터가 3가지의 당뇨병 타입에 대한 구분이 없어 세부 당뇨병의 분포가 다를 수 있는 것이다. 즉, 제 2형 당뇨병의 데이터가 당뇨병 안에서도 불균형한 분포를 가지고 있다면 이러한 결과가 도출될 수 있는 것이다.

테스트 결과

Confusion Matrix	학습 데이터 이용		테스트 데이터	
Actual	Predicted		Predicted	
	82	82	38	33
	27	293	14	123

Full Model	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
학습 데이터	0.50	0.75	0.92	0.77	0.68	0.60
테스트 데이터	0.54	0.73	0.90	0.77	0.69	0.62

학습 데이터에서 Simple Accuracy는 0.77, BCR은 0.68, F1-Measure은 0.60을 기록하고 있는 것을 확인할 수 있다. 테스트 데이터에서 Simple Accuracy는 0.77, BCR은 0.69, F1-Measure은 0.62를 기록하고 있는 것을 확인할 수 있다.

Simple Accuracy는 클래스 간 분포가 균형적인 경우에는 유용한 지표이다. 하지만, Confusion Matrix를 살펴본 결과, 학습 데이터와 테스트 데이터 모두 클래스 간 분포가 불균형 문제를 보이는 것을 확인할 수 있었다. 이에 따라, Simple Accuracy보다 BCR을 사용하는 것이 더 올바른 평가 척도라고 할 수 있다. BCR과 F1-Measure도 조금 다른 성격을 가지고 있는데, F1-Measure은 클래스 간 불균형이 있는 데이터셋에서도 잘 작동하는 평가 척도로써, Precision과 Recall을 조화평균 형태로 나타내어 실제 Positive 객체 중에서 모델에 의해 Positive Class로 정확히 예측된 비율과 모델이 Positive Class로 예측한 객체들 중에서 실제 Positive Class인 비율을 종합적으로 나타내는 지표라고 할 수 있다. 이 당뇨병 데이터셋은 건강과 관련된 데이터기 때문에 실제 Positive 객체 중에서 모델에 의해 Positive Class로 정확히 예측된 비율인 Recall 이 높아야 한다. 이에 따라 F1-Measure를 제일 중요하게 봐야한다는 것을 파악할 수 있다. 또한, 학습 데이터보다 테스트 데이터에서 좋은 Performance Metrics들을 보이는 것을 확인할 수 있는데, 이는 학습 데이터에 모델이 Overfitting이 되지 않고 Unseen Data에서 더 좋은 성능을 보이고 있음을 확인할 수 있다.

위에서 확인한 Confusion Matrix와 Performance Metrics들은 모두 분류 기준값 (Cut-off)에 영향을 받는 지표들이다. 이에 따라, Cut-off에 독립적인 측정 지표인 AUROC가 고안되었는데 가능한 모든 Cut-off에 대해 TPR과 FPR을 계산하여 나타낸 그래프의 넓이를 뜻한다. 테스트

데이터의 AUROC 값이 학습 데이터의 AUROC 값보다 커 학습데이터에 Overfitting 되지 않고 Unseen data에 보다 좋은 성능을 내고 있음을 재확인할 수 있다.

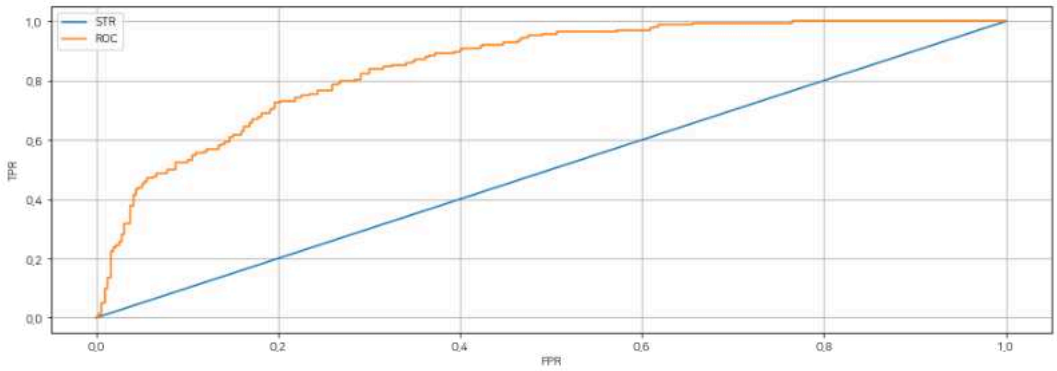

그림 4: Logistic Regression의 학습 데이터에 대한 ROC Curve

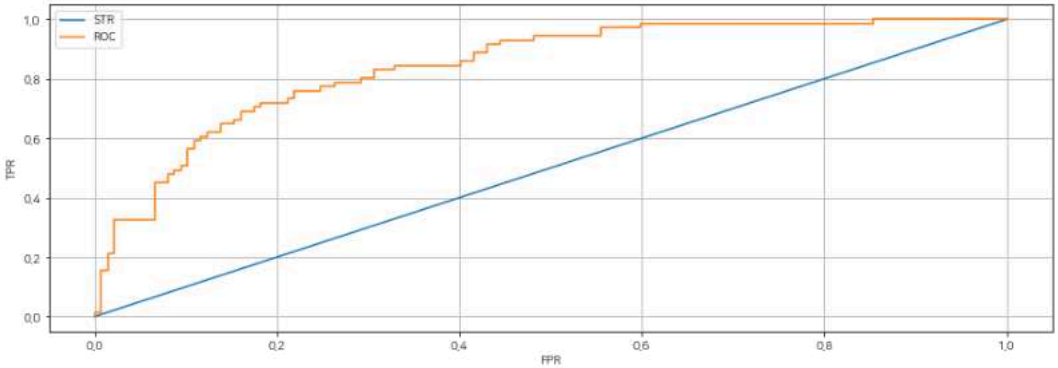

그림 5: Logistic Regression의 테스트 데이터에 대한 ROC Curve

Full Model	학습 데이터	테스트 데이터
AUROC	0.709	0.717

과연 유의하다고 나옴 변수만을 이용하면 모델 성능이 개선될까 궁금하여 위에서 유의하다고 도출된 Glucose, BMI, DiabetesPedigeeFunction 변수만 이용하여 모델을 학습 후 테스트까지 해보았다.

Confusion Matrix	학습 데이터 이용		테스트 데이터	
Actual	Predicted		Predicted	
	79	85	32	39
	23	297	8	129

Reduced LR	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
학습 데이터	0.48	0.77	0.93	0.78	0.67	0.59
테스트 데이터	0.45	0.80	0.94	0.77	0.65	0.58

Precision과 TNR은 증가하였고 나머지는 감소하는 모습을 확인할 수 있는데, 이는 모델이 정상임을 판단하는 개체가 늘어서기 때문이다. 본 데이터셋은 불균형한 분포를 가지고 있기 때문에 정상이라고 판단할 경우 눈에 보이는 Precision, TNR 같은 지표는 증가하지만 실질적으로 모델 목적에 부합하는 성능을 나타내는 지표인 TPR, BCR, F1-Measure는 대체로 성능이 훨씬 감소한 것을 확인할 수 있었다.

이러한 결과도 마찬가지로 Cut-off Value에 따라 달라지기 때문에 AUROC 값을 조사해보았다.

Reduced Model	학습 데이터	테스트 데이터
AUROC	0.705	0.696

하지만, 적은 변수로도 비슷한 성능을 낼 수 있다면 그만큼 데이터 수집에 이점이 있다고 볼 수 있기 때문에 적절한 Trade-off 수준을 고려하여 방법을 선정하는 게 좋은 방법이라고 느낄 수 있었다.

2. K-Nearest Neighbor Classifier

K-Nearest Neighbor Classifier (이하 K-NN)은 Lazy Learner의 한 종류이다. 학습 데이터에 대해 학습을 진행하지 않고 저장만 해놓은 후 테스트 데이터가 주어질 때 계산하는 방식이다. 이는 학습에 걸리는 시간이 없어 매우 간편하게 실행할 수 있다. 하지만 지역적으로 최적값을 찾아나가는 방식이기 때문에 Outlier에 민감하고 Overfitting이 될 수 있는 가능성이 높다는 특징을 가지고 있다. 이러한 K-NN에서는 가장 중요한 것이 어떤 K를 정할지인데, 해당 하이퍼파라미터 후보군을 구성한 후 Cross-Validation에서 가장 좋은 성능을 내는 하이퍼파라미터를 선정하여 분석에 이용하였다.

K는 인스턴스를 판단할 때 주위 몇 개의 인스터스를 통해 클래스를 판단할 것인지 나타내는 하이퍼파라미터이고, Weights는 가중치를 어느 방법을 통해 줄 것인가를 정하는 하이퍼파라미터이다. 즉, Uniform 방법은 모든 거리가 동일하게 반영되는 것이고 Distance는 가까울수록 더 많은 가중치를 주는 개념이라고 생각하면 된다.

KNN	하이퍼파라미터의 탐색 범위			
K값	3	5	7	10
Weights	Uniform		Distance	

최적 하이퍼파라미터 조합은 K값이 10일 때, 그리고 Distance 방법을 선택할 때가 검증데이터에 대해 가장 좋은 결과를 보인다고 도출되었다.

해당 조합을 사용해서 테스트 데이터에 대해 예측해본 결과가 아래 나열되어 있다.

Confusion Matrix	K-Nearest Neighbor Classifier	
	Predicted	
Actual	43	28
	25	112

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
KNN	0.61	0.63	0.82	0.75	0.70	0.62

수업 때 너무 단순한 구조를 가지고 있어 성능이 그렇게 좋지 않다고 배웠는데 생각보다 좋아서 놀랐다. 특히 TPR, BCR, F1-Measure에서 Logistic Regression 모델에 비해 모두 높은 값을 가지고 있어 헬스케어에 관련된 데이터인 당뇨병 예측에서는 더욱 알맞은 모델인 것을 확인할 수 있었다.

이러한 결과도 마찬가지로 Cut-off Value에 따라 달라지기 때문에 AUROC 값을 조사해보았다.

	K Nearest Neighbor
AUROC	0.711

확실하게 Logistic Regression보다 좋은 성능을 보이고 있는 것을 확인할 수 있다.

3. SVM

Support Vector Machine (이하 SVM)은 '데이터를 분리하려면 최전방 선만 그으면 되지 않을까'라는 생각에서 시작된 알고리즘으로 Kernel Function을 이용해 Data Transformation 후 경계면을 찾은 뒤 Inverse Function을 이용해 원상태를 복구하는 방식을 가지고 있다.

하이퍼파라미터 선정

SVM	하이퍼파라미터의 탐색 범위		
Penalty	L1		L2
C	0.25	0.5	1

SVM에서 하이퍼파라미터로 사용되는 것이 크게 Penalty와 C가 존재하는데, Penalty는 모델을 규제하는 파라미터로, 규제를 통해 변수들의 계수값을 작게 만들어 다중공선성을 줄이거나 복잡도를 조절할 수 있는 파라미터이다. L1이 Rasso , L2가 Ridge 를 의미한다. C는 오류를 얼마나 허용할지에 해당한다. 클수록 하드마진, 작을수록 소프트마진이라고 볼 수 있다.

최적 조합은 Penalty가 L2, C가 1일 때의 조합이 선정되었고 아래 해당 조합을 사용하여 테스트 데이터를 예측한 결과가 나열되어있다.

Confusion Matrix	Support Vector Machine	
	Predicted	
Actual	42	29
	15	122

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
SVM	0.59	0.74	0.89	0.79	0.73	0.66

TPR이 KNN보다 살짝 떨어지기는 하지만 정확도 및 BCR, F1-Measure에서 Logistic Regression 및 KNN에 비해 압도적인 성능을 보여주는 것을 확인할 수 있다. 이는 별도의 Kernel Function을 선언하지 않았는데 (즉, 선형 분류경계면 사용) 좋은 결과를 보여주는 것이 조금 특이했다.

이러한 결과도 마찬가지로 Cut-off Value에 따라 달라지기 때문에 AUROC 값을 조사해보았다.

	SVM
AUROC	0.741

확실하게 Logistic Regression, KNN보다 좋은 성능을 보이고 있는 것을 확인할 수 있다.
위의 과정에서 별도의 Kernel 함수를 선언하지 않았는데 좋은 성능을 보이고 있어 다양한 Kernel Function을 통해 결과를 비교해보면 좋겠다라는 생각이 들어 추가적으로 분석을 진행하였다.

SVM(Kernel)	하이퍼파라미터의 탐색 범위		
Kernel	rbf	poly	sigmoid
C	0.25	0.5	1

Kernel Function의 rbf는 Gaussian basis Function을 이용하는 것으로 표현할 수 있다. Poly는 다항식 커널, sigmoid는 Neural Network에서 Activation Function으로 자주 사용되는 함수이다. 최적 조합은 Kernel Function이 rbf, C가 0.5일 때의 조합이 선정되었고 아래 해당 조합을 사용하여 테스트 데이터를 예측한 결과가 나열되어있다.

Confusion Matrix	SVM (Kernel)	
Actual	Predicted	
	39	32
	17	120

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
SVM(Kernel)	0.55	0.70	0.88	0.76	0.69	0.61

Performance Metric을 확인해보면 선형 분류경계면을 사용했던 위의 모델보다 성능이 떨어지는 것을 확인할 수 있다. 해당 결과를 통해 무조건적으로 비선형 분류경계면을 사용하는 것이 효과적인 게 아니라는 것을 알 수 있었다.

이러한 결과도 마찬가지로 Cut-off Value에 따라 달라지기 때문에 AUROC 값을 조사해보았다.

	SVM (Kernel)
AUROC	0.712

확실하게 단순한 SVM보다 좋지 않은 성능을 보이고 있는 것을 확인할 수 있다.

4. Decision Tree

Full Tree의 Figure은 그림 6과 같다. 독립변수의 수가 8개임에도 불구하고 Decision Tree의 Depth가 깊고, Leaf Node의 수가 많은 것을 한 눈에 파악할 수 있다. 즉, Overfitting의 가능성이 높아 Pre-Pruning 혹은 Post-Pruning이 필요한 것을 확인할 수 있다.

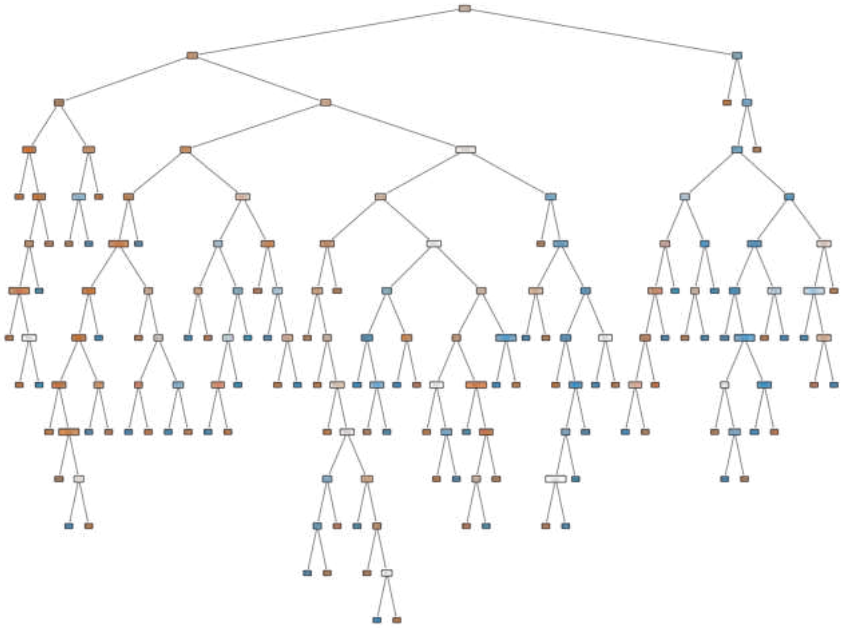

그림 6: Full Tree의 Figure

Post-Pruning

다음으로는 Post-Pruning Tree가 만들어지는 과정을 관찰하겠다. Post-Pruning은 비용 복잡도를 사용하여 최적의 의사결정나무 구조를 선택하는 과정이다. 이 비용 복잡도는 Cost Complexity(T) = Err(T) + alpha * L(T)이다. (Err(T) : 검증데이터에 대한 오분류율, L(T): 말단 노드의 수, alpha : Err(T)와 L(T)를 결합하는 가중치)

Alpha 값이 변함에 따라 Depth와 Leaf Node의 수가 감소하고, 이로 인한 Over fitting이 감소하여 학습데이터의 Impurity가 증가하는 것을 그림 7에서 관찰할 수 있다.

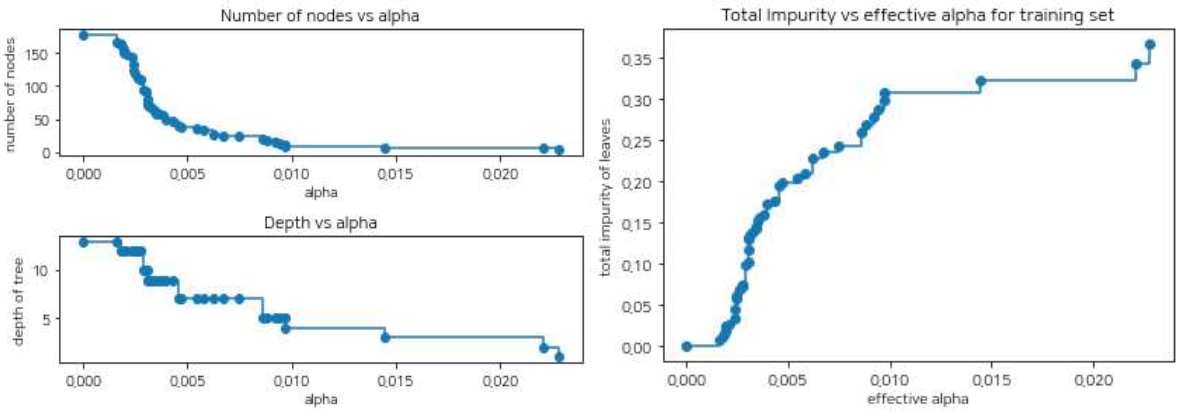

그림 7: 학습데이터에서 Alpha에 따른 변화 양상

이 Alpha들 중에서 최적의 값은 검증 데이터에서 가장 큰 Accuracy를 갖는 Alpha 값이 선정된다. 그림 7에서 해당 Alpha 값이 0.003과 0.004 사이에 존재함을 확인할 수 있다.

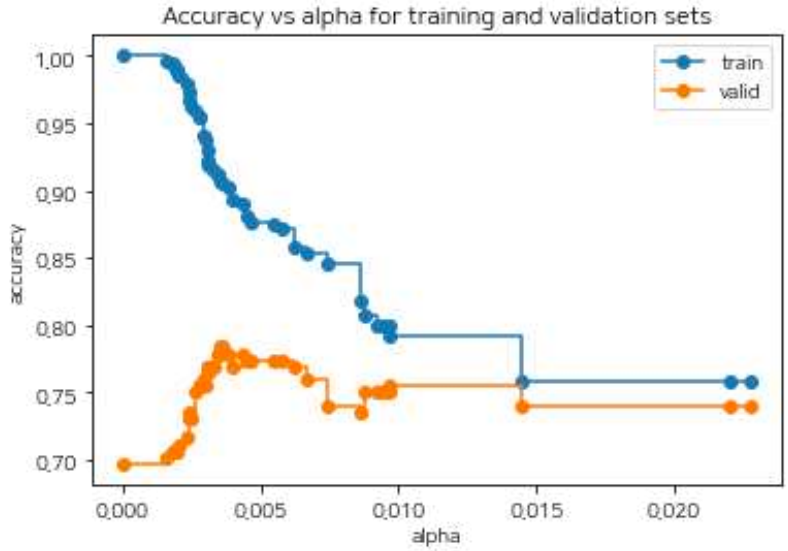

그림 8: Alpha 값에 따른 Accuracy 변화

이 과정들을 통해 생성된 Full-Tree와 Post-Pruning Tree의 Threshold가 0.5 (Default) 일 때의 Confusion Matrix는 아래와 같다.

Confusion Matrix	Full-Tree		Post-Pruning	
	Predicted		Predicted	
Actual	41	30	47	24
	26	111	21	116

Full-Tree와 Post-Pruning Tree을 비교해보았을 때 모델이 양성으로 예측한 수는 1 증가했으나 TPR과 TNR이 모두 증가한 것을 확인할 수 있다. 모델의 성능이 크게 개선된 것이다. 이러한 Confusion Matrix를 바탕으로 생성한 Performance에 대해 자세하게 살펴보겠다.

트리 종류	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Full-Tree	0.58	0.61	0.81	0.73	0.68	0.59
Post-Pruning	0.66	0.69	0.85	0.78	0.75	0.68

Post-Pruning 과정 후에 모든 Performance 지표에서 향상이 일어난 것을 확인할 수 있다.

TPR은 0.58에서 0.66으로 크게 증가하여 현재까지의 모델 중 가장 좋은 성능을 보이고 있음을 확인할 수 있다. TNR은 0.81에서 0.85로 증가하였으나 Feature Selection 후의 Logistic Regression 모델의 0.94보다는 떨어지는 것을 확인할 수 있다. Accuracy는 0.73에서 0.78으로 소폭 상승하였다.

Accuracy는 클래스 간 분포가 균형적인 경우에는 유용한 지표이다. 하지만 클래스 간 분포가 불균형 문제를 보이는 데이터에서는 BCR을 사용하는 것이 더 올바른 평가 척도라고 할 수 있다. 즉 BCR로 평가하는 것이 올바른 상황에서 BCR의 증가량이 Accuracy의 증가량보다 크게 증가한 것에서 더 큰 성능 향상이 일어났다고 생각할 수 있다.

F1-Measure은 클래스 간 불균형이 있는 데이터셋에서도 잘 작동하는 평가 척도로써, Precision과 Recall을 조화평균 형태로 나타내어 실제 Positive 객체 중에서 모델에 의해 Positive Class로 정확히 예측된 비율과 모델이 Positive Class로 예측한 객체들 중에서 실제 Positive Class인 비율을 종합적으로 나타내는 지표라고 할 수 있다.

이 당뇨병 데이터셋은 건강과 관련된 데이터기 때문에 실제 Positive 객체 중에서 모델에 의해 Positive Class로 정확히 예측된 비율인 Recall 이 높아야 하기 때문에 F1-Measure를 중요하게 봐야한다. 이 F1-Measure 도 Post-Pruning 과정 이후 0.59에서 0.68로 크게 증가한 것을 통해 성능이 향상되었다고 볼 수 있다.

본 분석은 Threshold가 Default인 0.5에 대해서만 분석을 했기 때문에 모든 Threshold 값을 조사하는 과정이 필요하다. 이 과정을 대표하는 AUROC에 대해서도 조사한 결과 성능 감소가 일어난 것을 확인할 수 있었다.

본 분석은 Threshold가 Default인 0.5에 대해서만 분석을 했기 때문에 모든 Threshold 값을 조사하는 과정이 필요하다. 이 과정을 대표하는 AUROC에 대해서도 조사한 결과 성능 향상이 일어난 것을 확인할 수 있었다.

	Full-Tree	Post-Pruning
AUROC	0.694	0.754

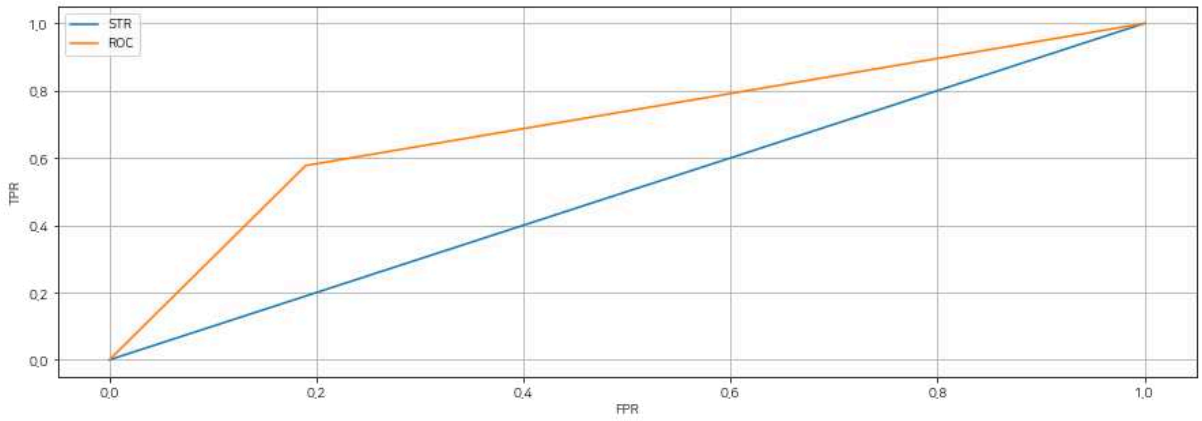

그림 9: Full Tree의 ROC Curve

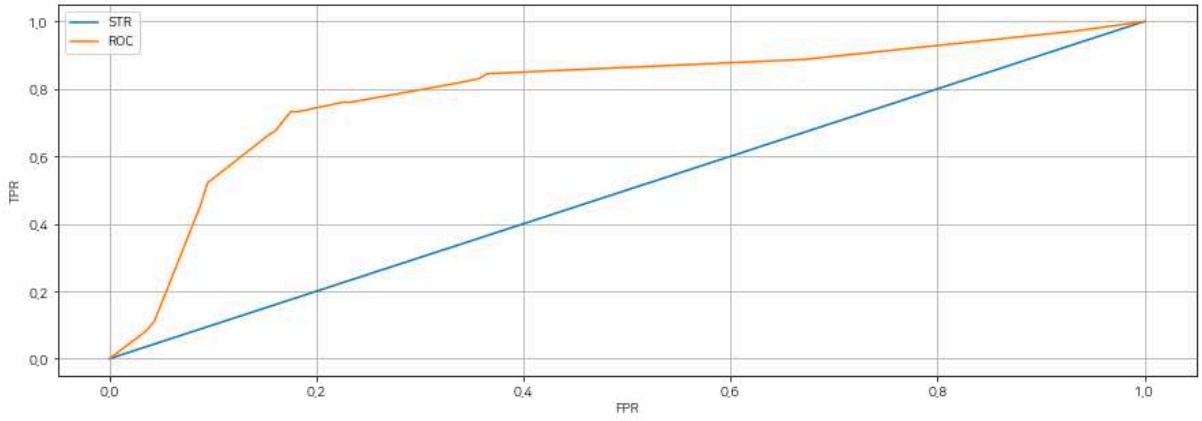

그림 10: Post Pruning 후의 ROC Curve

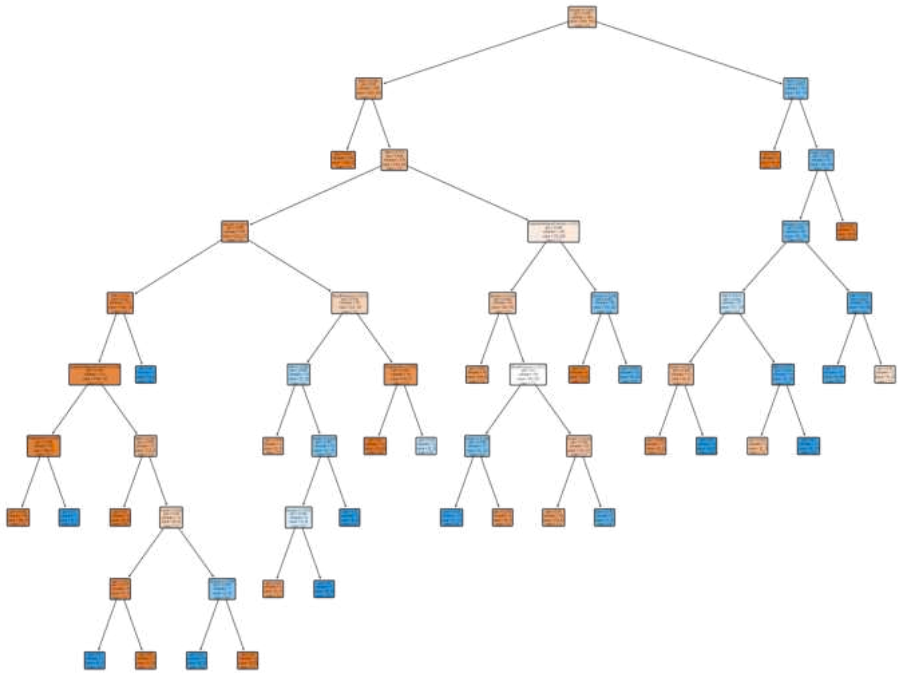

그림 11: Post Pruning이 완료된 Tree

Pre-Pruning

Pre-Pruning 과정에 사용할 하이퍼파라미터를 Criterion, Min_samples_split, Max_depth, min_samples_leaf 총 4가지를 선정하였다.

- Criterion: 분기의 성능을 평가하는 방법
- Min_samples_split: 분기 대상이 되는 노드에 속하는 최소 객체 수
- Max_depth: 의사결정나무가 가질 수 있는 최대 깊이
- Min_samples_leaf: 말단 노드에 속하는 최소 객체 수

	하이퍼파라미터의 탐색 범위			
Criterion	Gini	Entropy	Log_loss	
Min_split	10	20	30	50
Max_depth	5	10	15	
Min_leaf	3	5	10	

- Criterion은 대표적인 분기 성능 평가 지표인 Gini, Entropy, Log_loss를 선정하였다.
- Min_samples_split은 본 학습 데이터가 총 528개밖에 되지 않기 때문에 분기에 속할 수 있는 객체 수가 적을 수밖에 없다고 판단하여 스케일이 낮게 값을 설정하되, 너무 작으면 과적합 위험이 있기 때문에 이를 적절하게 방지할 수 있는 10, 20, 30, 50을 선정하였다.
- Max_depth 같은 경우 Full Tree의 Depth가 16개였기 때문에 이보다 적은 범위에서 균등하게 나눌 수 있는 5, 10, 15를 선정하였다.
- Min_samples_leaf: 말단 노드에 속하는 객체수가 너무 적다면 말단 노드가 무수히 많아지는 과적합 위험이 존재하기 한다. 이를 방지하기 위해서는 Min_samples_leaf의 값을 크게하면 되지만 본 데이터의 경우 당뇨병 데이터가 적은 불균형한 클래스 분포를 가진다. 이 때문에 당뇨병 데이터의 경우 말단 노드에 속하는 개체 수가 적을 수밖에 없기 때문에 이를 적절하게 3, 5, 10을 선정하였다.

이러한 하이퍼파라미터 조합 중 Criterion은 Gini, Min_split이 50, Max_depth가 5, Min_leaf가 3인 조합이 검증데이터에 대한 AUROC 값이 가장 높아 선정되었다.

이러한 조합이 선정된 이유에 대해 생각해보자면, 먼저 Min_split의 경우 너무 작은 값보다 50의 값이 적절한 클래스 분포를 유지하게 도와줘 과적합을 막을 수 있었고, 이로 인해 검증 데이터에서 좋은 성능을 내게 했을 것이라는 추측을 할 수 있다. 또한, Max_depth의 경우도

마찬가지다. 너무 깊은 Depth 값을 가진다면 수많은 조건들을 타고 간 다음에서야 분류가 진행되기 때문에 과적합의 위험이 높다. 이를 Full Tree와 비교하여 적절한 수준을 유지했기 때문에 최적의 값으로 선정되었다고 추측할 수 있다. Min_leaf의 경우 불균형 데이터셋이기 때문에 말단 노드 위의 Min_split 조건 위에서 당뇨병 데이터를 Purity가 높게 분류해내기 위해서는 적은 Leaf Node의 수가 필요하다는 것을 추측할 수 있다.

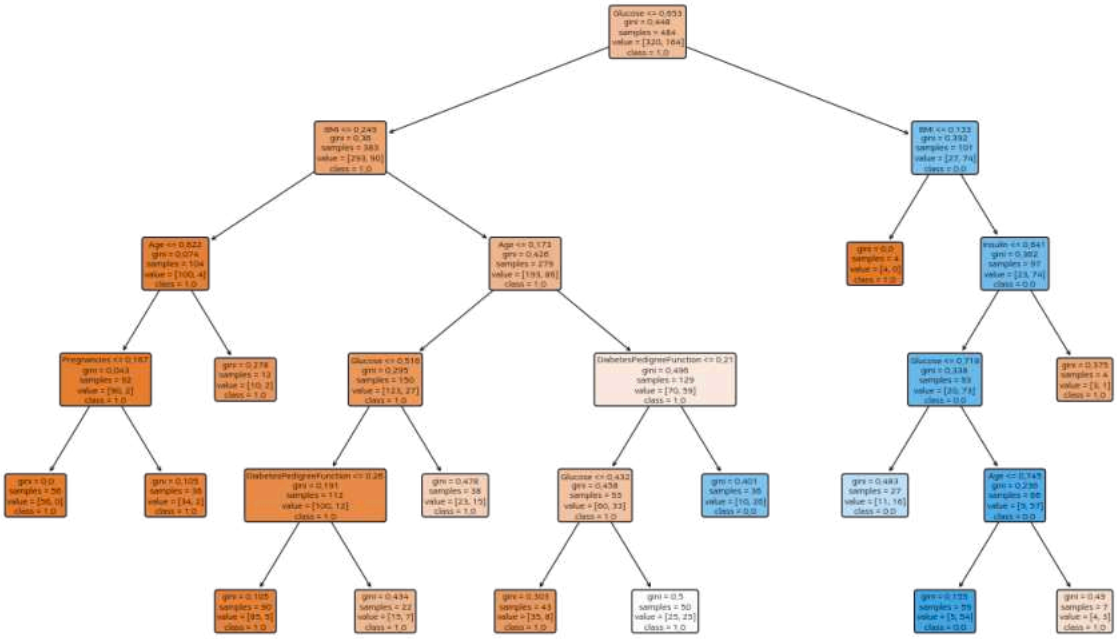

그림 12: Pre-Pruning의 Tree

정리

Decision Tree의 과정을 한눈에 파악하고자 Full Tree, Post-Pruning Tree, Pre-Pruning Tree의 테스트 데이터에 대한 Performance Measure를 아래에 나타내었다.

Threshold = 0.5일 때

Confusion Matrix	Full-Tree		Post-Pruning		Pre-Pruning	
Actual	Predicted		Predicted		Predicted	
	41	30	47	24	40	31
	26	111	21	116	22	115

트리 종류	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Full-Tree	0.58	0.61	0.81	0.73	0.68	0.59
Post-Pruning	0.66	0.69	0.85	0.78	0.75	0.68
Pre-Pruning	0.56	0.65	0.84	0.75	0.69	0.60

세 Tree 에 대해 Performance 지표를 비교했을 때 Post-Pruning Tree에서 모든 지표에서 가장 좋은 성능을 내는 것을 확인할 수 있다. Pre-Pruning이 Full-Tree보다 TPR을 제외하고는 모두 높아, Post-Pruning의 뒤를 잇는다고 할 수 있다. 즉, Post-Pruning > Pre-Pruning > Full-Tree 관계가 나타나는 것을 확인할 수 있었다.

단, 이 모든 결론은 Threshold가 0.5일 때 기준이기 때문에 AUROC 값을 확인해야 한다. 아래 표는 AUROC 값에서도 Post-Pruning이 가장 높은 값을 가지는 것을 확인할 수 있다.

	Full-Tree	Post-Pruning	Pre-Pruning
AUROC	0.694	0.754	0.701

5. Neural Network

Neural Net 학습에 사용될 하이퍼파라미터를 Solver, Max_iter, Alpha, Hidden Layer Sizes총 4가지를 선정하였다.

- Solver: 어떠한 Optimizer를 사용할지에 해당
- Max_iter: 최대 Iteration 수
- Alpha: L2-Regularization의 강도
- Hidden Layer Sizes: Hidden Layer에 속하는 노드의 수

	하이퍼파라미터의 탐색 범위		
Solver	L-BFGS	Adam	SGD
Max_iter	500	1000	1500
Alpha	1e-3	1e-4	1e-5
Hidden Layer Size	5	10	15

이러한 하이퍼파라미터 조합 중 Solver는 Adam, Max_iter이 1000, Alpha가 1e-3, Hidden Layer size가 5인 조합이 검증데이터에 대한 AUROC 값이 가장 높아 최적으로 선정되었다.

이러한 조합이 선정된 이유에 대해 생각해보자면, 먼저 Adam는 RMSProp + Momentum 기법이 합쳐진 방법이다. 즉, Objective Function에서 관성을 고려하고, 상황을 고려하며 Learning Rate를 조절하는 방식이다. 방향과 보폭 모두 결정하여 최적 값에 나아가는 방법이기 때문에 단순히 고정된 값을 이용하는 다른 방법보다 효과적임을 추측할 수 있다.

Max_iter는 학습을 통해 모델이 충분한 성능에 수렴할 수 있게 만든 지표이다. 학습 데이터의 크기가 작기 때문에 1000번 이하의 Iteration으로도 Loss Function이 충분한 값에 수렴한 것을 추측할 수 있다.

Alpha의 값이 클수록 규제의 강도가 강해진다. 이는 모델의 복잡성을 감소하게 만든다고 할 수 있다. 반대로, Alpha의 값이 작을수록 규제의 강도가 약해져 모델의 복잡성을 증가시킬 수 있다. 이와 비슷하게 Hidden Layer Size도 모델의 복잡성과 연결될 수 있다. Hidden Layer는 새로운 Latent Space에 데이터를 mapping하는 과정을 뜻한다. 이러한 Hidden Layer의 Size에 따라 Latent Space의 차원이 정해진다. 이는 Hidden layer의 Size가 클수록 복잡한 분류 경계면을 가진다고 할 수 있다. 이 두 가지 하이퍼파라미터의 최적값을 관찰하였을 때 복잡한 분류경계면을 가진 것보다 단순한 경계면을 가진 모델이 더욱 검증데이터에 대한 성능이 좋았음을 알 수 있다.

이러한 하이퍼파라미터를 이용한 모델의 테스트 데이터에 대한 예측 결과는 아래와 같다.

Confusion Matrix	Neural Network	
	Predicted	
Actual	43	28
	15	122

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
ANN	0.61	0.74	0.89	0.79	0.73	0.67

일반적으로 Neural Network의 성능이 가장 좋다고 알려져 있어서 그러한 양상을 보여줄 것이라고 기대를 하였는데, Post-Pruning Tree의 성능이 더 좋은 것을 확인할 수 있었다. 이러한 결과가 도출된 이유를 생각해보았을 때 '적절한 하이퍼파라미터를 선정하지 못하지 않았는가'라는 의문이 들었다. 현재 모델은 Layer가 1개이고 그 Layer 속의 노드들이 여러 개 존재하는 형태이다. 하지만 여러 층을 쌓아 DNN에 가까운 구조를 만든다면 보다 성능이 향상될 것 같아 추가 연구를 진행하였다.

	하이퍼파라미터의 탐색 범위		
Solver	L-BFGS	Adam	SGD
Max_iter	500	1000	1500
Alpha	1e-3	1e-4	1e-5
Hidden Layer Size	(5,)	(5,5,)	(5,5,5,)

하지만 최적 파라미터 선정 결과 이전과 동일한 결과가 도출되었다. 즉, 항상 Neural Network가 다른 모델보다 뛰어날 수는 없고 데이터셋의 특성에 따라 달라진다고 생각을 할 수 있다. Decision Tree는 축과 평행 or 수직으로의 분류경계면이 생긴다는 특징을 가지고 있다. 당뇨병 데이터에서는 이러한 분류경계면이 더욱 효과적일 수 있다.

이러한 결과도 마찬가지로 Cut-off Value에 따라 달라지기 때문에 AUROC 값을 조사해보았다.

	ANN
AUROC	0.748

확실하게 Post-Pruning Tree보다 좋지 않은 성능을 보이고 있는 것을 확인할 수 있다.

6. Ensemble

Bagging

현재까지 가장 좋은 성능을 가진 모델인 Post-Pruning된 모델을 바탕으로 Bagging을 진행하였다.

Bootstrap	TPR(Recall)	Precision	TNR	ACC	BCR	F1
단일 모델	0.66	0.69	0.85	0.78	0.75	0.68
30	0.61	0.66	0.84	0.76	0.71	0.63
60	0.65	0.64	0.81	0.75	0.72	0.64
90	0.65	0.67	0.83	0.77	0.73	0.66
120	0.59	0.63	0.82	0.74	0.70	0.61
150	0.61	0.67	0.85	0.76	0.72	0.64
180	0.62	0.67	0.84	0.76	0.72	0.64

210	0.59	0.66	0.84	0.75	0.70	0.62
240	0.66	0.65	0.82	0.76	0.74	0.66
270	0.61	0.65	0.83	0.75	0.71	0.63
300	0.65	0.65	0.82	0.76	0.73	0.65

Bootstrap 수를 30부터 300까지 30씩 차이나게 하면서 변하는 값을 관찰하여 보았는데, 전체적으로 Bootstrap 수에 따라 성능 지표에 큰 차이가 없는 것을 확인할 수 있었다. 심지어, 단일 모델보다 좋은 성능을 가진 모델조차 없었다. 이러한 이유는 앙상블은 서로 다른 모델은 서로 다른 분류 경계면/예측 곡선을 생성하는 것을 이용하여, 한 모델이 잘 분류하지 못하는 문제를 합해 잘 분류할 수 있게끔 만들어주는 장치이다. 하지만 본 Bagging 방식으로는 단순히 한 모델만 여러 개 생성했기 때문에 분류 경계면이 크게 달라지지 않는다. 그 결과 Bootstrap 수에 따라 큰 차이가 발생하지 않음을 확인할 수 있다.

Random Forest

Bootstrap	TPR(Recall)	Precision	TNR	ACC	BCR	F1
단일 모델	0.66	0.69	0.85	0.78	0.75	0.68
30	0.51	0.63	0.85	0.73	0.66	0.56
60	0.61	0.67	0.85	0.76	0.72	0.64
90	0.63	0.70	0.86	0.78	0.74	0.67
120	0.58	0.67	0.85	0.76	0.70	0.62
150	0.59	0.65	0.83	0.75	0.70	0.62
180	0.61	0.68	0.85	0.77	0.72	0.64
210	0.63	0.69	0.85	0.78	0.74	0.66
240	0.58	0.65	0.84	0.75	0.70	0.61
270	0.61	0.67	0.85	0.76	0.72	0.64
300	0.61	0.65	0.83	0.75	0.71	0.63

CART Bagging과 Random Forest 중 Random Forest가 Accuracy, BCR 지표 모두에서 CART Bagging보다 높은 분류 정확도를 가진다.

이유를 살펴보면, 앙상블은 서로 다른 모델은 서로 다른 분류 경계면/예측 곡선을 생성하는 것을 이용하여, 한 모델이 잘 분류하지 못하는 문제를 합해 잘 분류할 수 있게끔 만들어주는 장치이다. CART Bagging은 단순히 한 모델만 여러 개 생성했기 때문에 분류 경계면이 크게 달라지지 않는 반면 Random Forest는 Bagging에 더해 Randomly Chosen Predictor Variables 기능까지 있기 때문에 CART Bagging에 비해 다양성을 갖추게 되고, 효과적인 분류 경계면을 구성한 것으로 추측된다. 하지만, 높은 편차를 보이는 것을 확인할 수 있다. 이는 Random Forest 특성에 기인한다. 여러 조합이 갖춰지기 전까지는 들쑥날쑥 성능이 변하는 것이다. 하지만, 단일 모델에 비해서는 조금 아쉬운 모습을 보인다.

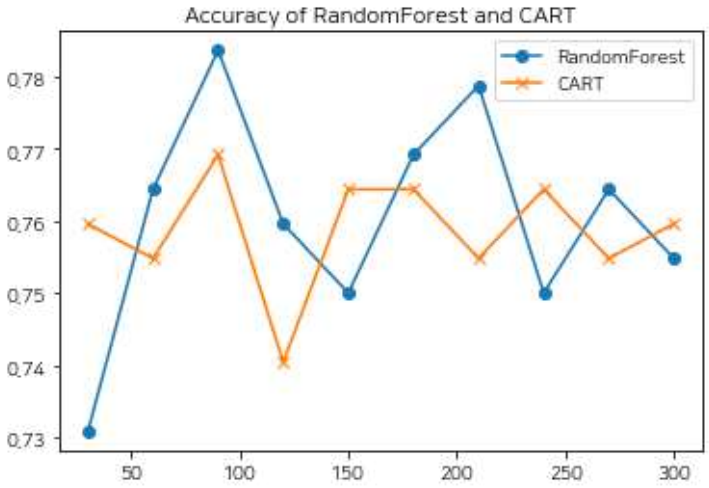

그림 13: RandomForest Bagging과 CART Bagging의 Accuracy

Adaboost

Adaboost 모델에 적용할 하이퍼파라미터를 n_estimators, Learning Rate, Stump Tree의 Depth 총 3가지를 선정하였다.

- n_estimators: Bootstrap 수
- Learning Rate: 이전 Weak Learner의 가중치를 제어
- Max_depth: Stump Tree의 높이

AdaBoost	하이퍼파라미터의 탐색 범위				
n_estimators	30	60	90	120	150
Learning Rate	0.1	0.5	1		
Max Depth	1	2	3		

최적의 하이퍼파라미터 조합을 보았을 때 n_estimators가 120, Learning Rate이 1, Max Depth가 1인 조합이 선정되었다. AdaBoost의 Learning Rate가 1인 경우, 모델이 이전 Weak Learner의 오차를 완전히 보정하기 위해 해당 Learner의 가중치를 그대로 사용한다. 이는 모델의 복잡도를 증가시키고, 과적합 가능성이 높아진다. n_estimators가 큰 값이고, Learning Rate가 0.1인 것을 살펴보았을 때 여러 Stump Tree를 이용해서 모델을 구성하는 것이 검증데이터에서 가장 좋은 성능을 보인 것을 확인할 수 있다.

GradientBoost

GradientBoost 모델에 적용할 하이퍼파라미터를 n_estimators, Learning Rate, Stump Tree의 Depth 총 3가지를 선정하였다.

- n_estimators: Bootstrap 수
- Learning Rate: 각 트리의 기여도를 Learning Rate만큼 축소
- Max_depth: Stump Tree의 높이

AdaBoost	하이퍼파라미터의 탐색 범위				
n_estimators	30	60	90	120	150
Learning Rate	0.1	0.5	1		
Max Depth	1	2	3		

최적의 하이퍼파라미터 조합을 보았을 때 n_estimators가 60, Learning Rate이 0.1, Max Depth가 1인 조합이 선정되었다. GradientBoosting의 Learning Rate에 따라 각 트리의 기여 정도를 조절하는데, 1일 경우 모델의 복잡도를 증가시키고, 과적합 가능성이 높아진다. n_estimators가 작은 값에 속하고, Learning Rate가 0.1인 것을 살펴보았을 때 Depth가 깊어지거나, n_estimators의 수를 무작정 크게 하기보다는 데이터셋 분류에서 못하는 것만 여러 번 반복하는 모델을 구축하는 것이 검증데이터에서 가장 좋은 분류 정확도를 나타낸다고 할 수 있다.

V. Conclusion

Model	TPR(Recall)	Precision	TNR	ACC	BCR	F1
Logistic Reg	0.54	0.73	0.90	0.77	0.69	0.62
KNN	0.61	0.63	0.82	0.75	0.70	0.62
SVM	0.59	0.74	0.89	0.79	0.73	0.66
SVM(Kernel)	0.55	0.70	0.88	0.76	0.69	0.61
Full Tree	0.58	0.61	0.81	0.73	0.68	0.59
Post-Pruning	0.66	0.69	0.85	0.78	0.75	0.68
Pre-Pruning	0.56	0.65	0.84	0.75	0.69	0.60
ANN	0.61	0.74	0.89	0.79	0.73	0.67
CART Bagging	0.62	0.67	0.84	0.76	0.72	0.64
RandomForest	0.64	0.69	0.85	0.78	0.74	0.66
AdaBoost	0.61	0.69	0.86	0.77	0.72	0.65
GradientBoost	0.46	0.72	0.91	0.75	0.65	0.56

Post Pruning이 TPR, BCR, F1-Measure에서 1등을 기록하며 영예의 베스트 모델로 선정되었다. 이후 뒤를 이어 ANN이 Precision과 Accuracy에서 1등을 기록해 2번째 베스트 모델로 선정되었다. ANN만 단독으로 분석했을 당시에는 좋지 않은 성능을 가진 것 같다고 판단하였는데 Post-Pruning Tree가 압도적인 것이었고 ANN도 준수한 성능을 가지고 있는 것을 결론에서 확인할 수 있었다.

Ensemble 모델 실험을 통해 단순히 Bootstrap으로 모델의 수를 늘리는 것이 가장 좋은 게 아니라 다양성을 부여하는 장치가 갖춰져야 효과적인 성능이 나타난다고 확인할 수 있었다.

Random Forest의 특성에 대해 다시 한 번 정리하자면,

1. 앙상블 기법 : Random Forest는 Decision Tree를 앙상블 하는 형태로 작동하며, 다수의 Decision Tree를 조합함으로써 각 트리의 다양한 특성과 Decision Node를 반영할 수 있다.

2. BootStrap : Random Forest는 Bootstrap을 통해 데이터를 학습하게 된다. 즉, 다양한 데이터 구성을 이용하여 학습을 진행하기 때문에 모델의 다양성이 높다.
3. 특성 무작위성: Random Forest는 각 Decision Node에서 무작위로 특성을 추출하여 분기한다. 즉, 이는 상관관계가 높은 특성들에 과적합되는 문제를 방지한다.

이러한 특성을 이용하기 때문에 가장 높은 정확도를 기록할 것이라고 예상하였는데 예상과 다른 결과가 도출되었다.

이렇게 총체적으로 프로젝트를 선정하고 실습하는 과정을 통해 수업에서 배운 다양한 모델들에 대한 이해를 더욱 깊게 하고 직접적으로 활용할 수 있는 능력을 함양할 수 있어서 매우 알찼다.