피마인디언 분석

장태성

적용 라이브러리

```
%load_ext watermark
%watermark -v -p sklearn,numpy,scipy,matplotlib
import matplotlib.pyplot as plt
plt.rcParams['image.cmap'] = "gray"
from IPython.display import display
import pandas as pd
import numpy as np
import platform

import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

path = "c:/Windows/Fonts/malgun.ttf"
from matplotlib import font_manager, rc
if platform.system() == 'Darwin':
    rc('font', family='AppleGothic')
elif platform.system() == 'Windows':
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
else:
    print('Unknown system... sorry~~~~')

plt.rcParams['axes.unicode_minus'] = False
```

```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import plotly.offline as py
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import plotly.tools as tls
import plotly.figure_factory as ff
import plotly.express as px
py.init_notebook_mode(connected=True)
import squarify
```

```
import scipy.stats as ss
from scipy import interp
from scipy.stats import randint as sp_randint
from scipy.stats import uniform as sp_uniform
```

```
CPython 3.7.4
IPython 7.8.0

sklearn 0.0
numpy 1.16.5
scipy 1.3.1
matplotlib 3.1.3
```

```
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import GridSearchCV, cross_val_score, train_test_split, GridSearchCV, RandomizedSearchCV
from sklearn.metrics import precision_score, recall_score, confusion_matrix, roc_curve, precision_recall_curve, accuracy_score, roc_auc_s
import lightgbm as lgbm
from sklearn.ensemble import VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import roc_curve,auc
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_predict
from yellowbrick.classifier import DiscriminationThreshold
```

데이터 프레임 살펴보기

원본 데이터 프레임

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

data['Outcome'] != 0

분리

당뇨병 걸린 사람

not_zero_data

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
2	8	183	64	0	0	23.3	0.672	32	1
4	0	137	40	35	168	43.1	2.288	33	1
6	3	78	50	32	88	31.0	0.248	26	1
8	2	197	70	45	543	30.5	0.158	53	1
...	...	...	...	...	...	...	...	...	...
755	1	128	88	39	110	36.5	1.057	37	1
757	0	123	72	0	0	36.3	0.258	52	1
759	6	190	92	0	0	35.5	0.278	66	1
761	9	170	74	31	0	44.0	0.403	43	1
766	1	126	60	0	0	30.1	0.349	47	1

268 rows × 9 columns

분리

data['Outcome'] == 0

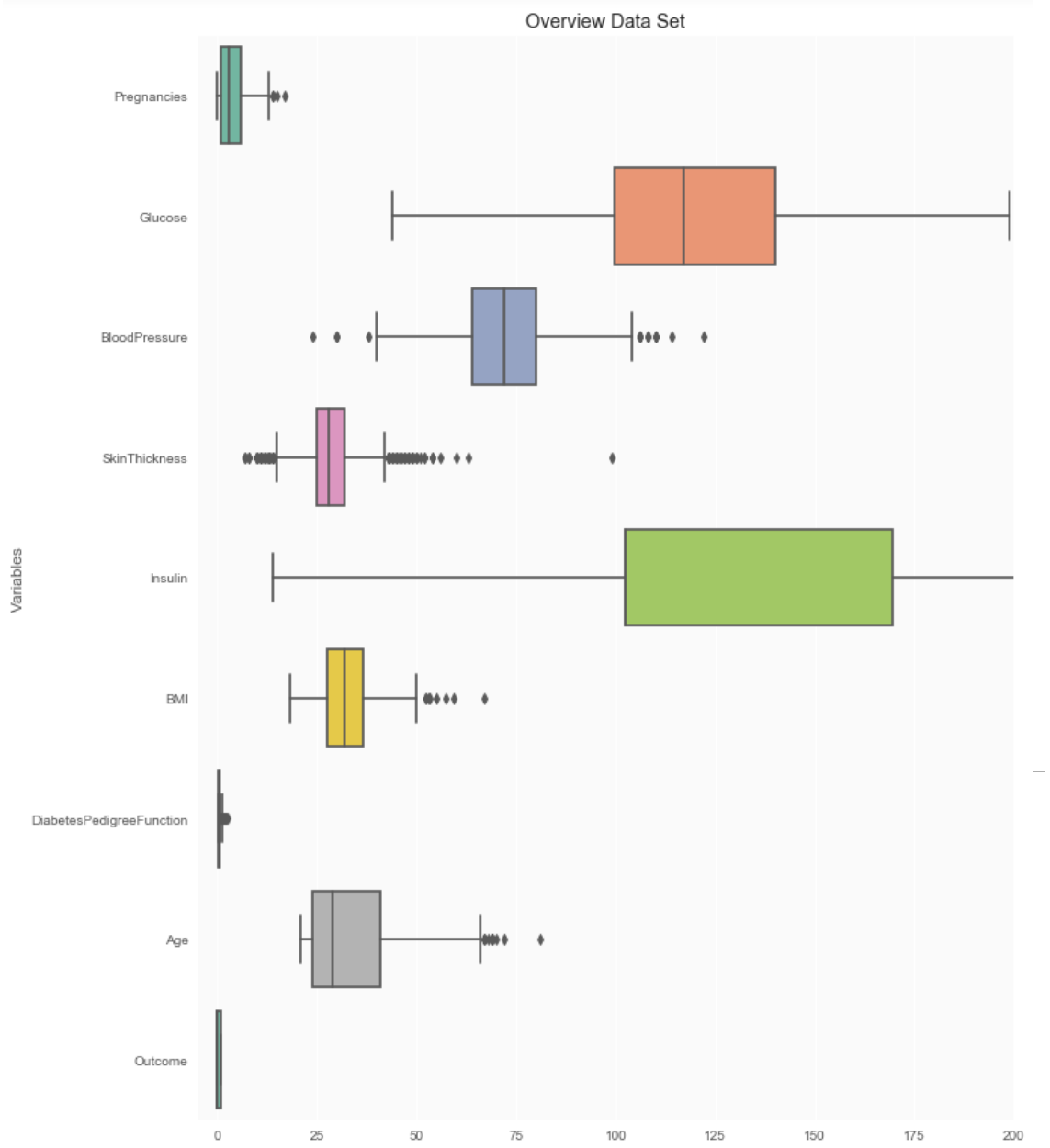
당뇨병 걸리지 않은 사람

include_zero_data

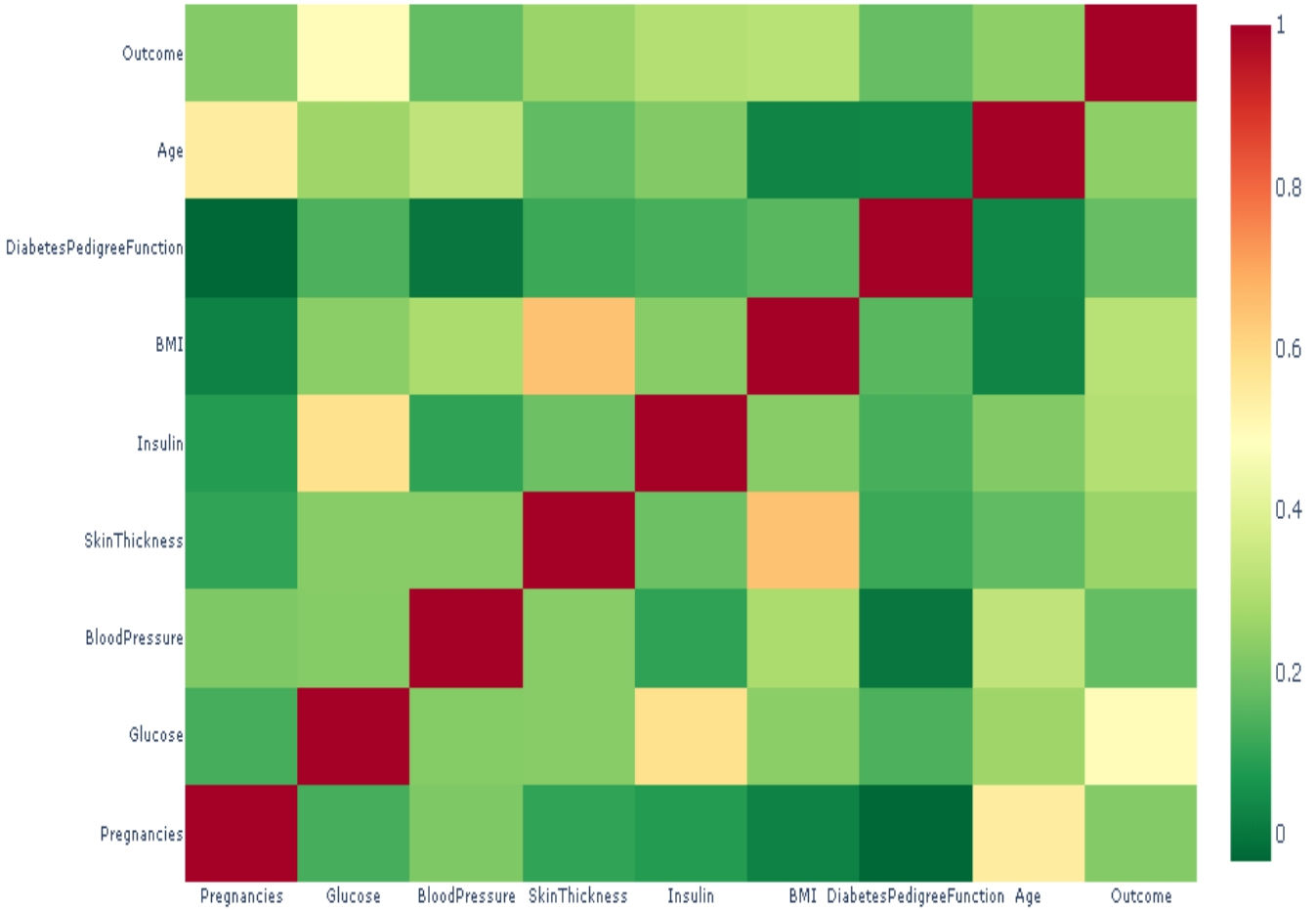
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	1	85	66	29	0	26.6	0.351	31	0
3	1	89	66	23	94	28.1	0.167	21	0
5	5	116	74	0	0	25.6	0.201	30	0
7	10	115	0	0	0	35.3	0.134	29	0
10	4	110	92	0	0	37.6	0.191	30	0
...	...	...	...	...	...	...	...	...	...
762	9	89	62	0	0	22.5	0.142	33	0
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
767	1	93	70	31	0	30.4	0.315	23	0

500 rows × 9 columns

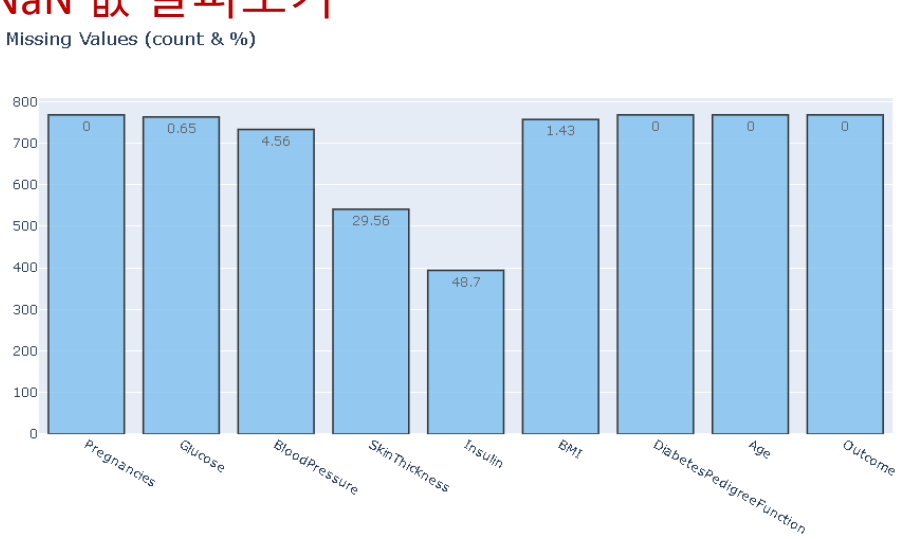
변수 분포 살펴보기



변수가 상관관계 분석-heatmap



NaN 값 살펴보기



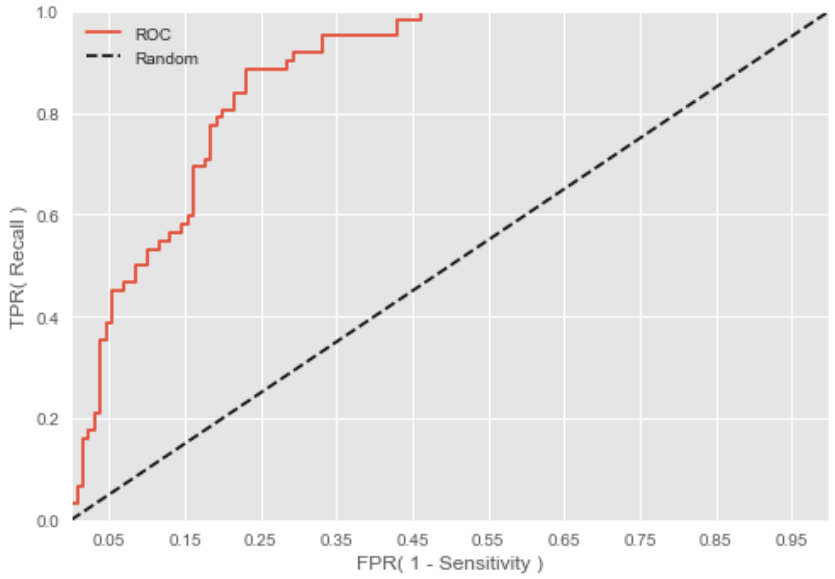
NaN 값을 Median으로 채운다.

적용한 코드

```
data.loc[(data['Outcome'] == 0 ) & (data[변수].isnull()), 변수] = 변수별 건강한 사람들의 median 값
data.loc[(data['Outcome'] == 1 ) & (data[변수].isnull()), 변수] = 변수별 당뇨병 환자들의 median 값
```

자료의 종합적 분석과 결과 해석

ROC AUC 값 : 0.8743



1. 원본 데이터를 불러온 후 'outcome' 컬럼에서 '0'이면 건강한 사람으로, '1'이면 당뇨병 걸린 사람으로 자료를 분리 한다.
2. 데이터 분리전의 데이터로 각 컬럼의 4분위표를 그리고 이상값이나 median, 25%, 50%, 75% 분포가 어떻게 되어있는지 등을 확인한다.
3. 각 컬럼간 상관관계 분석을하여 heatmap으로 도표를 그려 어떤 변수끼리 상관관계가 가장 큰지를 색깔로 확인한다.
4. 원본 데이터에서 'outcome' 컬럼을 제외한 나머지 변수들의 '0' 값을 np.nan으로 변환하여 nan값을 막대그래프로 확인, 앞서 나눴던 데이터 프레임으로 각 컬럼간 분포와 median 값을 구하여 nan 값을 media값으로 채워넣고, train, test 값으로 나눈 후 머신러닝 라이브러리로 분석하여 ROC 커브를 그리고 0.8743 점수를 얻었다.