분류(classification) 문제 실습 - 항구의 기뢰 찾기

학습 목표

- urlopen를 이용하여 웹에서의 데이터를 가져오는 것을 실습해 본다.
- pandas를 이용하여 웹에서의 데이터를 가져오는 것을 실습해 본다.
- 데이터를 가져오고 이를 시각화를 통해 데이터를 이해해 본다.

학습 내용

- 웹에서의 데이터를 가져와 보기
- 웹의 데이터를 Pandas를 이용해 가져와 보기
- 데이터 인사이트를 확인하기 위한 데이터 시각화

- 데이터 셋 : UC Irvine Data Repository
  - https://archive.ics.uci.edu/ml/datasets.php (https://archive.ics.uci.edu/ml/datasets.php)
  - Connectionist Bench (Sonar, Mines vs. Rocks) Data Set
    - https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+%28Sonar%2C+Mines+vs.+Rocks% (https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+%28Sonar%2C+Mines+vs.+Rocks%
    - sonar.all-data

데이터 설명

- 데이터 셋 특성 : 다변수
- 행의 수 : 208개
- 열의 수 : 60개
- 데이터 설명
  - 다양한 각도와 다양한 조건에서 금속 실린더에서 수중 음파 탐지기 신호를 통해 얻은 다양한 패턴이 포함되어 있음.
  - 군사작전의 결과로 항구에 남아 있는 폭파되지 않은 기뢰를 찾기 위해 소나(Sonar, 수중 음파탐지기)를 이용할 수 있는지 확인하기 위한 어떤 실험으로 만들어짐.
  - 반 정도의 표본은 바위, 나머지 반은 기뢰 모양의 금속 원통을 나타냄.

01. 데이터 준비

In [4]:

```
from urllib.request import urlopen
import sys
```

In [9]:

```
target_url = "https://archive.ics.uci.edu/ml/machine-learning-databases/undocumented
data = urlopen(target_url)
data
```

Out[9]:

```
<http.client.HTTPResponse at 0x7fbe0745ddf0>
```

레이블을 리스트로, 속성을 리스트의 리스트로 확인

In [11]:

```
list_dat = []
labels = []
for line in data:
    #print(line)
    # 쉼표로 분리
    line = str(line)
    row = line.strip().split(",")
    list_dat.append(row)
```

In [14]:

```
len(list_dat), list_dat[0]
```

Out[14]:

```
(207,
 ["b'0.0453",
  '0.0523',
  '0.0843',
  '0.0689',
  '0.1183',
  '0.2583',
  '0.2156',
  '0.3481',
  '0.3337',
  '0.2872',
  '0.4918',
  '0.6552',
  '0.6919',
  '0.7797',
  '0.7464',
  '0.9444',
  '1.0000',
  '0.8874',
  '0.8024',
  '0.7818',
  '0.5212',
  '0.4052',
  '0.3957',
  '0.3914',
  '0.3250',
  '0.3200',
  '0.3271',
  '0.2767',
  '0.4423',
  '0.2028',
  '0.3788',
  '0.2947',
  '0.1984',
  '0.2341',
  '0.1306',
  '0.4182',
  '0.3835',
  '0.1057',
  '0.1840',
  '0.1970',
  '0.1674',
  '0.0583',
  '0.1401',
  '0.1628',
  '0.0621',
  '0.0203',
  '0.0530',
  '0.0742',
  '0.0409',
  '0.0061',
  '0.0125',
  '0.0084',
  '0.0089',
```

```
  '0.0048',
  '0.0094',
  '0.0191',
  '0.0140',
  '0.0049',
  '0.0052',
  '0.0044',
  "R\\n'"])
```

01. 데이터 준비 - pandas의 활용

In [60]:

```
import matplotlib.pyplot as plt
import pandas as pd
```

In [61]:

```
target_url = "https://archive.ics.uci.edu/ml/machine-learning-databases/undocumented
rocksVMines = pd.read_csv(target_url, header=None, prefix='V')
rocksVMines
```

Out[61]:

	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
0	0.0200	0.0371	0.0428	0.0207	0.0954	0.0986	0.1539	0.1601	0.3109	0.2111	0.1609	0.1582	0.2238
1	0.0453	0.0523	0.0843	0.0689	0.1183	0.2583	0.2156	0.3481	0.3337	0.2872	0.4918	0.6552	0.6919
2	0.0262	0.0582	0.1099	0.1083	0.0974	0.2280	0.2431	0.3771	0.5598	0.6194	0.6333	0.7060	0.5544
3	0.0100	0.0171	0.0623	0.0205	0.0205	0.0368	0.1098	0.1276	0.0598	0.1264	0.0881	0.1992	0.0184
4	0.0762	0.0666	0.0481	0.0394	0.0590	0.0649	0.1209	0.2467	0.3564	0.4459	0.4152	0.3952	0.4256
5	0.0286	0.0453	0.0277	0.0174	0.0384	0.0990	0.1201	0.1833	0.2105	0.3039	0.2988	0.4250	0.6343
6	0.0317	0.0956	0.1321	0.1408	0.1674	0.1710	0.0731	0.1401	0.2083	0.3513	0.1786	0.0658	0.0513
7	0.0519	0.0548	0.0842	0.0319	0.1158	0.0922	0.1027	0.0613	0.1465	0.2838	0.2802	0.3086	0.2657
8	0.0223	0.0375	0.0484	0.0475	0.0647	0.0591	0.0753	0.0098	0.0684	0.1487	0.1156	0.1654	0.3833

In [62]:

```
### 보이지 않는 행과 열 보기
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

In [63]:

```
rocksVMines
```

Out[63]:

	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
0	0.0200	0.0371	0.0428	0.0207	0.0954	0.0986	0.1539	0.1601	0.3109	0.2111	0.1609	0.1582	0.2238
1	0.0453	0.0523	0.0843	0.0689	0.1183	0.2583	0.2156	0.3481	0.3337	0.2872	0.4918	0.6552	0.6919
2	0.0262	0.0582	0.1099	0.1083	0.0974	0.2280	0.2431	0.3771	0.5598	0.6194	0.6333	0.7060	0.5544
3	0.0100	0.0171	0.0623	0.0205	0.0205	0.0368	0.1098	0.1276	0.0598	0.1264	0.0881	0.1992	0.0184
4	0.0762	0.0666	0.0481	0.0394	0.0590	0.0649	0.1209	0.2467	0.3564	0.4459	0.4152	0.3952	0.4256
5	0.0286	0.0453	0.0277	0.0174	0.0384	0.0990	0.1201	0.1833	0.2105	0.3039	0.2988	0.4250	0.6343
6	0.0317	0.0956	0.1321	0.1408	0.1674	0.1710	0.0731	0.1401	0.2083	0.3513	0.1786	0.0658	0.0513
7	0.0519	0.0548	0.0842	0.0319	0.1158	0.0922	0.1027	0.0613	0.1465	0.2838	0.2802	0.3086	0.2657
8	0.0223	0.0375	0.0484	0.0475	0.0647	0.0591	0.0753	0.0098	0.0684	0.1487	0.1156	0.1654	0.3833

In [64]:

```
print(rocksVMines.shape)
print(rocksVMines.columns)
print(rocksVMines.head())
print(rocksVMines.tail())
print()
print(rocksVMines.info())
```

```
(208, 61)
Index(['V0', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9',
'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V1
9', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'V2
9', 'V30',
       'V31', 'V32', 'V33', 'V34', 'V35', 'V36', 'V37', 'V38', 'V3
9', 'V40',
       'V41', 'V42', 'V43', 'V44', 'V45', 'V46', 'V47', 'V48', 'V4
9', 'V50',
       'V51', 'V52', 'V53', 'V54', 'V55', 'V56', 'V57', 'V58', 'V5
9', 'V60'],
      dtype='object')
       V0      V1      V2      V3      V4      V5      V6      V7
V8  \
0  0.0200  0.0371  0.0428  0.0207  0.0954  0.0986  0.1539  0.1601
0.3109
1  0.0453  0.0523  0.0843  0.0689  0.1183  0.2583  0.2156  0.3481
```

In [65]:

```
# V60 값 확인
rocksVMines.V60.value_counts()
```

Out[65]:

```
M    111
R     97
Name: V60, dtype: int64
```

- 데이터 탐색 확인
  - R : 바위, M : 기뢰
  - 행 208행, 열 61열
  - 값의 범위?

02 데이터 시각화

- 때때로 시각화가 숫자로 된 내용에서 얻기 어려운 데이터에 대한 인사이트를 제공한다.

평행 좌표계(parallel coordinates plot)

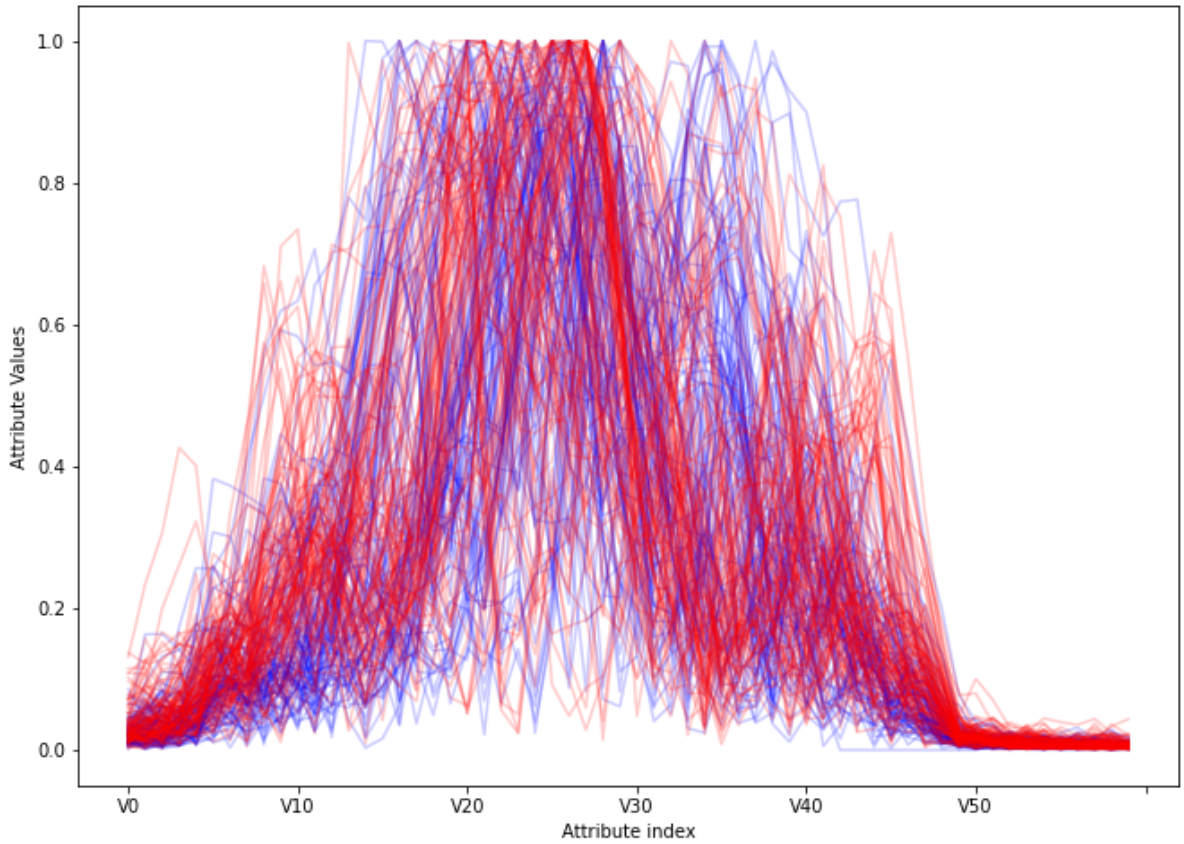
- 속성 개수가 작지 않은 문제에 유용하게 사용할 수 있는 시각화 방법 중 하나

In [66]:

```
plt.figure(figsize=(11,8))
for i in range(208):
    if rocksVMines.iat[i, 60] == "M":
        pcolor = "red"
    else:
        pcolor = "blue"

    dataRow = rocksVMines.iloc[i, 0:60] # 0~59열의 데이터
    dataRow.plot(color=pcolor, alpha=0.2)


plt.xlabel("Attribute index")
plt.ylabel(("Attribute Values"))
plt.show()
```



- 도표의 아래 부분에서는 푸른 선이 조금 나타난다.(파란색 : Rock)
- 속성 30~40사이의 경우에는 값들의 분포가 푸른선보다 붉은 선이 조금 높다.

각 특성(피처, 속성)간의 상호관계는 어떻게 될까?

- 각 속성 간의 상호관계 시각화

In [67]:

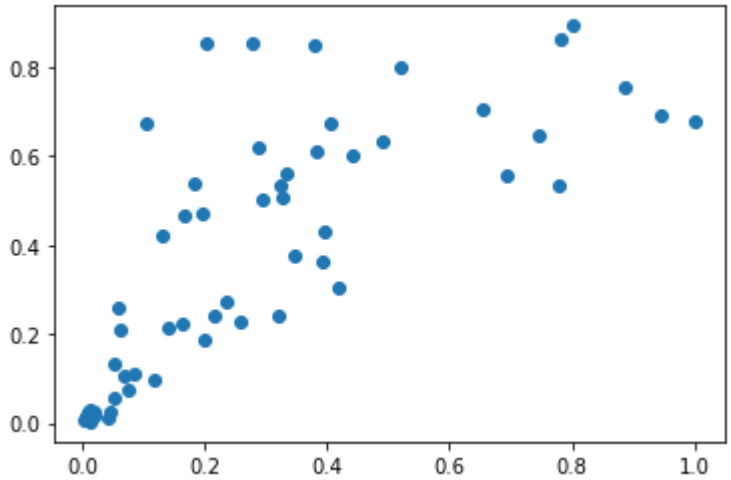
```
# 2번째 행과 3번째 행의 상호관계 시각화
dat_row2 = rocksVMines.iloc[1,0:60]
dat_row3 = rocksVMines.iloc[2,0:60]

print( type(dat_row2), type(dat_row3) )
plt.scatter(dat_row2, dat_row3)
```

```
<class 'pandas.core.series.Series'> <class 'pandas.core.series.Serie
s'>
```

Out[67]:

```
<matplotlib.collections.PathCollection at 0x7fbe0dbfd9a0>
```



실습. 2행과 5행 속성의 상호관계 시각화를 확인해 보자.

히트맵을 이용한 상관계수 시각화

In [72]:

```
dat_corr = rocksVMines.corr()
dat_corr
```

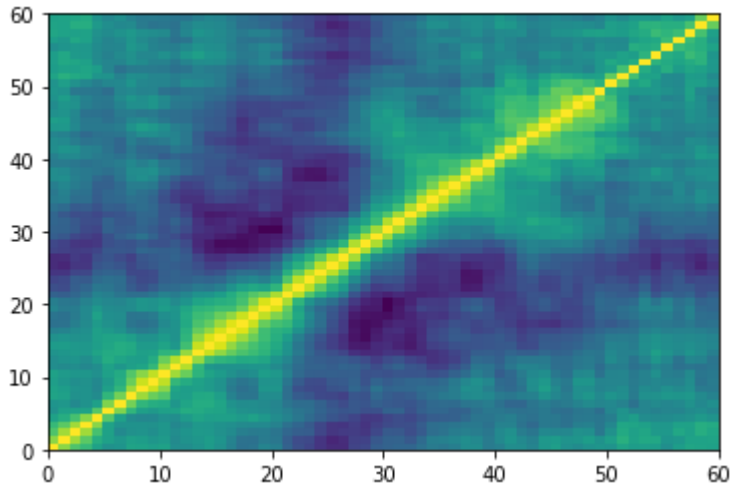
Out[72]:

	V0	V1	V2	V3	V4	V5	V6	V7	
V0	1.000000	0.735896	0.571537	0.491438	0.344797	0.238921	0.260815	0.355523	0.35
V1	0.735896	1.000000	0.779916	0.606684	0.419669	0.332329	0.279040	0.334615	0.31
V2	0.571537	0.779916	1.000000	0.781786	0.546141	0.346275	0.190434	0.237884	0.25
V3	0.491438	0.606684	0.781786	1.000000	0.726943	0.352805	0.246440	0.246742	0.24
V4	0.344797	0.419669	0.546141	0.726943	1.000000	0.597053	0.335422	0.204006	0.17
V5	0.238921	0.332329	0.346275	0.352805	0.597053	1.000000	0.702889	0.471683	0.32
V6	0.260815	0.279040	0.190434	0.246440	0.335422	0.702889	1.000000	0.675774	0.47
V7	0.355523	0.334615	0.237884	0.246742	0.204006	0.471683	0.675774	1.000000	0.77
V8	0.353420	0.316733	0.252691	0.247078	0.177906	0.327578	0.470580	0.778577	1.00
V9	0.318276	0.270782	0.219637	0.237769	0.183219	0.288621	0.425448	0.652525	0.87
V10	0.344058	0.297065	0.274610	0.271881	0.231684	0.333570	0.396588	0.584583	0.72
V11	0.210861	0.194102	0.214807	0.175381	0.211657	0.344451	0.274432	0.328329	0.36
V12	0.210722	0.249596	0.258767	0.215754	0.299086	0.411107	0.365391	0.322951	0.31
V13	0.256278	0.273170	0.291724	0.286708	0.359062	0.396233	0.409576	0.387114	0.32
V14	0.304878	0.307599	0.285663	0.278529	0.318059	0.367908	0.411692	0.391514	0.29
V15	0.239079	0.261844	0.237017	0.248245	0.328725	0.353783	0.363086	0.322237	0.24
V16	0.137845	0.152170	0.201093	0.223203	0.326477	0.293190	0.250024	0.140912	0.10
V17	0.041817	0.042870	0.120587	0.194992	0.299266	0.235778	0.208057	0.061333	0.02
V18	0.055227	0.040911	0.099303	0.189405	0.340543	0.226305	0.215495	0.061825	0.06
V19	0.156760	0.102428	0.103117	0.188317	0.285737	0.206841	0.196496	0.204950	0.26
V20	0.117663	0.075255	0.063990	0.142271	0.205088	0.174768	0.165827	0.208785	0.26
V21	-0.056973	-0.074157	-0.026815	0.036010	0.152897	0.123770	0.063773	0.023786	0.01
V22	-0.163426	-0.179365	-0.073400	-0.029749	0.073934	0.064081	0.009359	-0.092087	-0.15
V23	-0.218093	-0.196469	-0.085380	-0.102975	-0.000624	0.027026	0.011982	-0.124427	-0.18
V24	-0.295683	-0.295302	-0.214256	-0.206673	-0.067296	-0.043280	-0.057147	-0.196354	-0.19
V25	-0.342865	-0.365749	-0.291974	-0.291357	-0.125675	-0.100309	-0.126074	-0.203178	-0.13
V26	-0.341703	-0.337046	-0.263111	-0.294749	-0.169618	-0.129094	-0.179526	-0.233332	-0.11
V27	-0.224340	-0.234386	-0.256674	-0.256074	-0.214692	-0.118645	-0.116848	-0.120343	-0.02
V28	-0.199099	-0.228490	-0.290728	-0.300476	-0.283863	-0.156081	-0.129694	-0.139750	-0.09
V29	-0.077430	-0.115301	-0.197493	-0.236602	-0.273350	-0.151186	-0.068142	-0.017654	0.05
V30	-0.048370	-0.055862	-0.106198	-0.190086	-0.214336	-0.054136	-0.096945	-0.081072	-0.04
V31	-0.030444	-0.049683	-0.109895	-0.169987	-0.173485	-0.051934	-0.115871	-0.108115	-0.02

	V0	V1	V2	V3	V4	V5	V6	V7	
V32	-0.031939	-0.108272	-0.170671	-0.164651	-0.200586	-0.144391	-0.127052	-0.087246	-0.01
V33	0.031319	-0.004247	-0.099409	-0.083965	-0.140559	-0.070337	-0.077662	-0.014578	0.01
V34	0.098118	0.115824	0.017053	0.015200	-0.086529	-0.028815	-0.015531	0.035733	0.01
V35	0.080722	0.132611	0.053070	0.039282	-0.073481	-0.023621	0.002979	0.087187	0.03
V36	0.119565	0.169186	0.107530	0.063486	-0.064617	-0.064798	-0.001376	0.110739	0.11
V37	0.209873	0.217494	0.130276	0.089887	-0.008620	-0.048745	0.065900	0.186609	0.22
V38	0.208371	0.186828	0.110499	0.089346	0.063408	0.030599	0.080942	0.206145	0.21
V39	0.099993	0.098350	0.074137	0.045141	0.061616	0.081119	0.112673	0.184411	0.12
V40	0.127313	0.188226	0.189047	0.145241	0.098832	0.075797	0.041071	0.097517	0.01
V41	0.213592	0.261345	0.233442	0.144693	0.125181	0.048763	-0.028720	0.076054	-0.00
V42	0.206057	0.186368	0.113920	0.050629	0.063706	0.034380	-0.025727	0.114721	0.05
V43	0.157949	0.133018	0.071946	-0.008407	0.031575	0.048870	0.061404	0.135426	0.21
V44	0.279968	0.285716	0.180734	0.087824	0.089202	0.085468	0.110813	0.240176	0.32
V45	0.319354	0.304247	0.173649	0.080012	0.081964	0.029524	0.076537	0.169099	0.19
V46	0.230343	0.255797	0.179528	0.046109	0.041419	0.016640	0.098925	0.109744	0.08
V47	0.203234	0.265279	0.234896	0.121065	0.084435	0.067196	0.155221	0.222783	0.22
V48	0.247560	0.313995	0.223074	0.133294	0.088128	0.080729	0.194720	0.271422	0.22
V49	0.269287	0.245868	0.081096	0.077925	0.066751	0.017300	0.166112	0.191615	0.15
V50	0.254450	0.320538	0.238110	0.174676	0.115936	0.171767	0.184152	0.260692	0.17
V51	0.355299	0.434548	0.394076	0.374651	0.266617	0.252288	0.144051	0.219038	0.20
V52	0.311729	0.346076	0.332914	0.364772	0.314985	0.162404	0.046403	0.102447	0.10
V53	0.322299	0.383960	0.367186	0.334211	0.205306	0.164073	0.163074	0.234008	0.20
V54	0.312067	0.380165	0.289731	0.284955	0.196472	0.133464	0.195541	0.239551	0.17
V55	0.220642	0.262263	0.287661	0.280938	0.199323	0.166758	0.174143	0.276819	0.23
V56	0.313725	0.280341	0.380819	0.340254	0.219395	0.161333	0.186324	0.267212	0.19
V57	0.368132	0.353042	0.334108	0.344865	0.238793	0.203986	0.242646	0.287603	0.23
V58	0.357116	0.352200	0.425047	0.420266	0.290982	0.220573	0.183578	0.194400	0.09
V59	0.347078	0.358761	0.373948	0.400626	0.253710	0.178158	0.222493	0.146216	0.09

In [73]:

```
plt.pcolor(dat_corr)
plt.show()
```

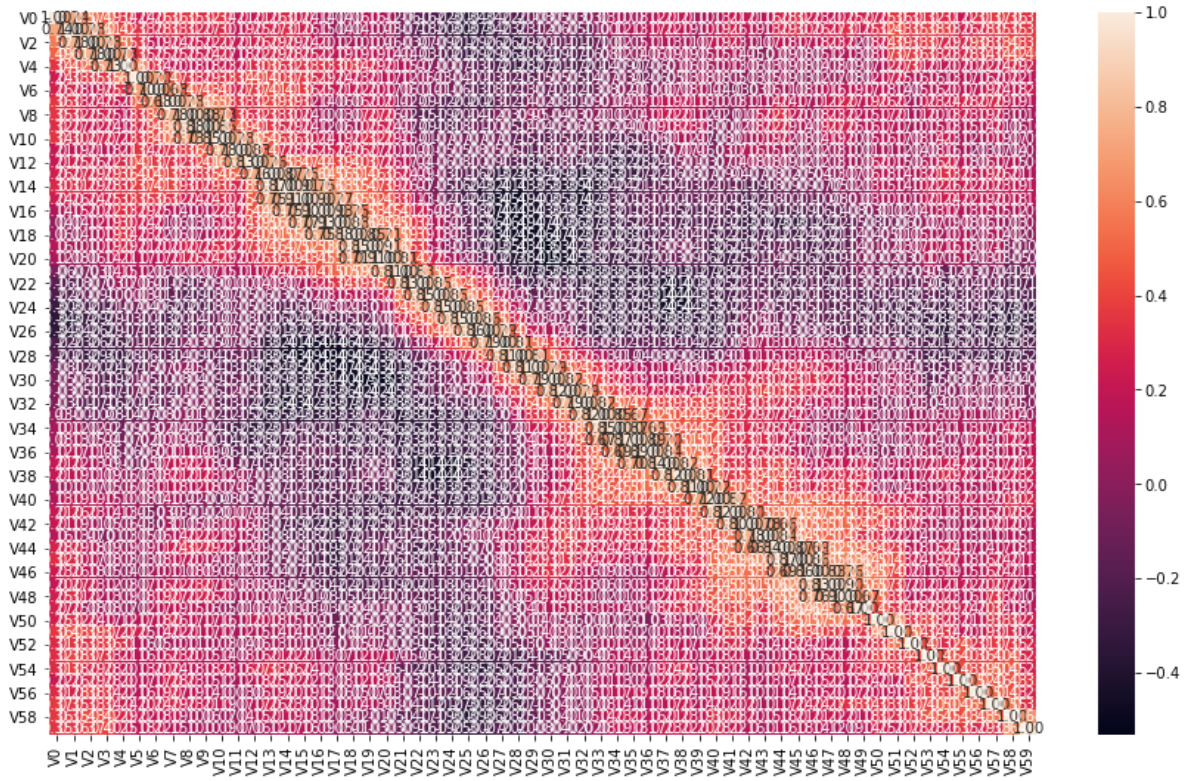


In [71]:

```
import seaborn as sns
```

In [80]:

```
plt.figure(figsize=(15,9))
sns.heatmap(dat_corr, annot=True, fmt='.2f', )
plt.show()
```



In [ ]:

```
### 첫번째에서 20번째 속성에 대한 상관관계를 확인
```

In [83]:

```
dat20_corr = rocksVMines.iloc[:,0:20].corr()
dat20_corr
```

Out[83]:

	V0	V1	V2	V3	V4	V5	V6	V7	V8
V0	1.000000	0.735896	0.571537	0.491438	0.344797	0.238921	0.260815	0.355523	0.353420
V1	0.735896	1.000000	0.779916	0.606684	0.419669	0.332329	0.279040	0.334615	0.316733
V2	0.571537	0.779916	1.000000	0.781786	0.546141	0.346275	0.190434	0.237884	0.252691
V3	0.491438	0.606684	0.781786	1.000000	0.726943	0.352805	0.246440	0.246742	0.247078
V4	0.344797	0.419669	0.546141	0.726943	1.000000	0.597053	0.335422	0.204006	0.177906
V5	0.238921	0.332329	0.346275	0.352805	0.597053	1.000000	0.702889	0.471683	0.327578
V6	0.260815	0.279040	0.190434	0.246440	0.335422	0.702889	1.000000	0.675774	0.470580
V7	0.355523	0.334615	0.237884	0.246742	0.204006	0.471683	0.675774	1.000000	0.778577
V8	0.353420	0.316733	0.252691	0.247078	0.177906	0.327578	0.470580	0.778577	1.000000
V9	0.318276	0.270782	0.219637	0.237769	0.183219	0.288621	0.425448	0.652525	0.877131
V10	0.344058	0.297065	0.274610	0.271881	0.231684	0.333570	0.396588	0.584583	0.728063
V11	0.210861	0.194102	0.214807	0.175381	0.211657	0.344451	0.274432	0.328329	0.363404
V12	0.210722	0.249596	0.258767	0.215754	0.299086	0.411107	0.365391	0.322951	0.316899
V13	0.256278	0.273170	0.291724	0.286708	0.359062	0.396233	0.409576	0.387114	0.329659
V14	0.304878	0.307599	0.285663	0.278529	0.318059	0.367908	0.411692	0.391514	0.299575
V15	0.239079	0.261844	0.237017	0.248245	0.328725	0.353783	0.363086	0.322237	0.241819
V16	0.137845	0.152170	0.201093	0.223203	0.326477	0.293190	0.250024	0.140912	0.100146
V17	0.041817	0.042870	0.120587	0.194992	0.299266	0.235778	0.208057	0.061333	0.027380
V18	0.055227	0.040911	0.099303	0.189405	0.340543	0.226305	0.215495	0.061825	0.067237
V19	0.156760	0.102428	0.103117	0.188317	0.285737	0.206841	0.196496	0.204950	0.266455

In [ ]: