МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа №3
по курсу «Методы машинного обучения»

Тема: «Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных»

ИСПОЛНИТЕЛЬ: _Бабин В.Е._____
ФИО
группа ИУ5-22М _____
подпись
"__"_____2020 г.

ПРЕПОДАВАТЕЛЬ: _____
ФИО
_____
подпись
"__"_____2020 г.

Москва - 2020

lab3

April 20, 2020

[9]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

[11]:
```
data = pd.read_csv('FIFA 2018 Statistics.csv', sep=",")
rows, columns = data.shape
print('rows = {}; cols = {};'.format(rows, columns))
data.head()
```

```
rows = 128; cols = 27;
```

[11]:

	Date	Team	Opponent	Goal Scored	Ball Possession %	\
0	14-06-2018	Russia	Saudi Arabia	5	40	
1	14-06-2018	Saudi Arabia	Russia	0	60	
2	15-06-2018	Egypt	Uruguay	0	43	
3	15-06-2018	Uruguay	Egypt	1	57	
4	15-06-2018	Morocco	Iran	0	64	

	Attempts	On-Target	Off-Target	Blocked	Corners	…	Yellow Card	\
0	13	7	3	3	6	…	0	
1	6	0	3	3	2	…	0	
2	8	3	3	2	0	…	2	
3	14	4	6	4	5	…	0	
4	13	3	6	4	5	…	1	

	Yellow & Red	Red	Man of the Match	1st Goal	Round	PSO	\
0	0	0	Yes	12.0	Group Stage	No	
1	0	0	No	NaN	Group Stage	No	
2	0	0	No	NaN	Group Stage	No	
3	0	0	Yes	89.0	Group Stage	No	
4	0	0	No	NaN	Group Stage	No	

	Goals in PSO	Own goals	Own goal Time
0	0	NaN	NaN
1	0	NaN	NaN

```
2             0        NaN            NaN
3             0        NaN            NaN
4             0        1.0           90.0

[5 rows x 27 columns]
```

<———————————————————> <———————————————————>

[15]:
```
#
#
num_cols = []
for col in data.columns:
    #
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / rows) * 100, 2)
        print('    {}.        {}.                {}, {}%.'.format(col, dt,␣
 ↪temp_null_count, temp_perc))
```

```
    1st Goal.        float64.                34, 26.56%.
    Own goals.        float64.                116, 90.62%.
    Own goal Time.        float64.                116,
90.62%.
```

[7]:
```
#
data_num = data[num_cols]
data_num
```

[7]:
```
     1st Goal  Own goals  Own goal Time
0        12.0        NaN            NaN
1         NaN        NaN            NaN
2         NaN        NaN            NaN
3        89.0        NaN            NaN
4         NaN        1.0           90.0
..        …        …            …
123       5.0        NaN            NaN
124       4.0        NaN            NaN
125       NaN        NaN            NaN
126      18.0        1.0           18.0
127      28.0        NaN            NaN

[128 rows x 3 columns]
```

[16]:
```
#
flt_index = data[data['1st Goal'].isnull()].index
flt_index
```

[16]:
```
Int64Index([  1,   2,   4,  12,  15,  16,  18,  23,  25,  35,  37,  38,  43,
             44,  47,  49,  62,  65,  72,  73,  74,  81,  82,  84,  88,  90,
             94, 105, 109, 112, 114, 116, 121, 125],
            dtype='int64')
```

[17]:
```
#
data[data.index.isin(flt_index)]
```

[17]:

	Date	Team	Opponent	Goal Scored	\
1	14-06-2018	Saudi Arabia	Russia	0	
2	15-06-2018	Egypt	Uruguay	0	
4	15-06-2018	Morocco	Iran	0	
12	16-06-2018	Peru	Denmark	0	
15	17-06-2018	Nigeria	Croatia	0	
16	17-06-2018	Costa Rica	Serbia	0	
18	17-06-2018	Germany	Mexico	0	
23	18-06-2018	Korea Republic	Sweden	0	
25	18-06-2018	Panama	Belgium	0	
35	20-06-2018	Morocco	Portugal	0	
37	20-06-2018	Saudi Arabia	Uruguay	0	
38	20-06-2018	Iran	Spain	0	
43	21-06-2018	Peru	France	0	
44	21-06-2018	Argentina	Croatia	0	
47	22-06-2018	Costa Rica	Brazil	0	
49	22-06-2018	Iceland	Nigeria	0	
62	24-06-2018	Poland	Colombia	0	
65	25-06-2018	Russia	Uruguay	0	
72	26-06-2018	Denmark	France	0	
73	26-06-2018	France	Denmark	0	
74	26-06-2018	Australia	Peru	0	
81	27-06-2018	Germany	Korea Republic	0	
82	27-06-2018	Mexico	Sweden	0	
84	27-06-2018	Serbia	Brazil	0	
88	28-06-2018	Japan	Poland	0	
90	28-06-2018	Senegal	Colombia	0	
94	28-06-2018	England	Belgium	0	
105	02-07-2018	Mexico	Brazil	0	
109	03-07-2018	Switzerland	Sweden	0	
112	06-07-2018	Uruguay	France	0	
114	06-07-2018	Brazil	Belgium	1	
116	07-07-2018	Sweden	England	0	
121	10-07-2018	Belgium	France	0	
125	14-07-2018	England	Belgium	0	

	Ball Possession %	Attempts	On-Target	Off-Target	Blocked	Corners	\
1	60	6	0	3	3	2	
2	43	8	3	3	2	0	

4	64	13	3	6	4	5
12	52	18	6	7	5	3
15	46	14	2	5	7	5
16	50	10	3	3	4	5
18	60	25	9	9	7	8
23	48	5	0	2	3	5
25	39	6	2	4	0	3
35	53	16	4	10	2	7
37	53	8	3	3	2	4
38	30	5	0	5	0	2
43	56	10	2	6	2	3
44	58	10	3	3	4	5
47	34	4	0	4	0	1
49	42	10	3	6	1	5
62	45	9	2	3	4	7
65	44	3	1	1	1	2
72	38	5	1	2	2	4
73	62	11	4	6	1	2
74	53	14	2	7	5	8
81	70	26	6	11	9	9
82	65	19	3	8	8	7
84	44	10	1	5	4	5
88	54	10	3	4	3	5
90	43	8	3	4	1	1
94	48	13	1	7	5	7
105	53	13	1	4	8	7
109	63	18	4	5	9	11
112	42	11	4	6	1	4
114	57	26	9	7	10	8
116	43	7	3	3	1	1
121	60	9	3	5	1	5
125	57	15	5	7	3	5

	…	Yellow Card	Yellow & Red	Red	Man of the Match	1st Goal	\
1	…	0	0	0	No	NaN	
2	…	2	0	0	No	NaN	
4	…	1	0	0	No	NaN	
12	…	1	0	0	No	NaN	
15	…	1	0	0	No	NaN	
16	…	2	0	0	No	NaN	
18	…	2	0	0	Yes	NaN	
23	…	2	0	0	No	NaN	
25	…	5	0	0	No	NaN	
35	…	1	0	0	No	NaN	
37	…	0	0	0	No	NaN	
38	…	2	0	0	No	NaN	
43	…	2	0	0	No	NaN	

44	…	3	0	0	No	NaN
47	…	1	0	0	No	NaN
49	…	0	0	0	No	NaN
62	…	2	0	0	No	NaN
65	…	1	1	0	No	NaN
72	…	1	0	0	No	NaN
73	…	0	0	0	Yes	NaN
74	…	4	0	0	No	NaN
81	…	0	0	0	No	NaN
82	…	3	0	0	No	NaN
84	…	3	0	0	No	NaN
88	…	1	0	0	No	NaN
90	…	1	0	0	No	NaN
94	…	0	0	0	No	NaN
105	…	4	0	0	No	NaN
109	…	2	0	1	No	NaN
112	…	2	0	0	No	NaN
114	…	2	0	0	No	NaN
116	…	2	0	0	No	NaN
121	…	3	0	0	No	NaN
125	…	2	0	0	No	NaN

	Round	PSO	Goals in PSO	Own goals	Own goal Time
1	Group Stage	No	0	NaN	NaN
2	Group Stage	No	0	NaN	NaN
4	Group Stage	No	0	1.0	90.0
12	Group Stage	No	0	NaN	NaN
15	Group Stage	No	0	1.0	32.0
16	Group Stage	No	0	NaN	NaN
18	Group Stage	No	0	NaN	NaN
23	Group Stage	No	0	NaN	NaN
25	Group Stage	No	0	NaN	NaN
35	Group Stage	No	0	NaN	NaN
37	Group Stage	No	0	NaN	NaN
38	Group Stage	No	0	NaN	NaN
43	Group Stage	No	0	NaN	NaN
44	Group Stage	No	0	NaN	NaN
47	Group Stage	No	0	NaN	NaN
49	Group Stage	No	0	NaN	NaN
62	Group Stage	No	0	NaN	NaN
65	Group Stage	No	0	1.0	23.0
72	Group Stage	No	0	NaN	NaN
73	Group Stage	No	0	NaN	NaN
74	Group Stage	No	0	NaN	NaN
81	Group Stage	No	0	NaN	NaN
82	Group Stage	No	0	1.0	74.0
84	Group Stage	No	0	NaN	NaN

```
88      Group Stage   No             0        NaN            NaN
90      Group Stage   No             0        NaN            NaN
94      Group Stage   No             0        NaN            NaN
105     Round of 16   No             0        NaN            NaN
109     Round of 16   No             0        NaN            NaN
112  Quarter Finals   No             0        NaN            NaN
114  Quarter Finals   No             0        1.0           13.0
116  Quarter Finals   No             0        NaN            NaN
121    Semi- Finals   No             0        NaN            NaN
125       3rd Place   No             0        NaN            NaN

[34 rows x 27 columns]
```

[18]:
```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

[19]:
```
strategies=['mean', 'median','most_frequent']
```

[21]:
```
def test_num_impute_col(dataset, column, strategy_param):
    temp_data = dataset[[column]]

    indicator = MissingIndicator()
    mask_missing_values_only = indicator.fit_transform(temp_data)

    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(temp_data)

    filled_data = data_num_imp[mask_missing_values_only]

    return column, strategy_param, filled_data.size, filled_data[0],␣
↪filled_data[filled_data.size-1]
```

[22]:
```
test_num_impute_col(data, '1st Goal', strategies[0])
```

[22]:
```
('1st Goal', 'mean', 34, 39.45744680851064, 39.45744680851064)
```

[23]:
```
test_num_impute_col(data, 'Own goals', strategies[1])
```

[23]:
```
('Own goals', 'median', 116, 1.0, 1.0)
```

[24]:
```
test_num_impute_col(data, 'Own goal Time', strategies[2])
```

[24]:
```
('Own goal Time', 'most_frequent', 116, 90.0, 90.0)
```

[27]:
```
import requests
import io


url = "https://datahub.io/core/airport-codes/r/airport-codes.csv"
s = requests.get(url).content
data = pd.read_csv(io.StringIO(s.decode('utf-8')), error_bad_lines=False)
data
```

[27]:

	ident	type	name	\
0	00A	heliport	Total Rf Heliport	
1	00AA	small_airport	Aero B Ranch Airport	
2	00AK	small_airport	Lowell Field	
3	00AL	small_airport	Epps Airpark	
4	00AR	closed	Newport Hospital & Clinic Heliport	
…	…	…	…	
56055	ZYYK	medium_airport	Yingkou Lanqi Airport	
56056	ZYYY	medium_airport	Shenyang Dongta Airport	
56057	ZZ-0001	heliport	Sealand Helipad	
56058	ZZ-0002	small_airport	Glorioso Islands Airstrip	
56059	ZZZZ	small_airport	Satsuma IÅ jima Airport	

	elevation_ft	continent	iso_country	iso_region	municipality	\
0	11.0	NaN	US	US-PA	Bensalem	
1	3435.0	NaN	US	US-KS	Leoti	
2	450.0	NaN	US	US-AK	Anchor Point	
3	820.0	NaN	US	US-AL	Harvest	
4	237.0	NaN	US	US-AR	Newport	
…	…	…	…	…	…	
56055	0.0	AS	CN	CN-21	Yingkou	
56056	NaN	AS	CN	CN-21	Shenyang	
56057	40.0	EU	GB	GB-ENG	Sealand	
56058	11.0	AF	TF	TF-U-A	Grande Glorieuse	
56059	338.0	AS	JP	JP-46	Mishima-Mura	

	gps_code	iata_code	local_code	coordinates
0	00A	NaN	00A	-74.93360137939453, 40.07080078125
1	00AA	NaN	00AA	-101.473911, 38.704022
2	00AK	NaN	00AK	-151.695999146, 59.94919968
3	00AL	NaN	00AL	-86.77030181884766, 34.86479949951172
4	NaN	NaN	NaN	-91.254898, 35.6087
…	…	…	…	…
56055	ZYYK	YKH	NaN	122.3586, 40.542524
56056	ZYYY	NaN	NaN	123.49600219726562, 41.784400939941406
56057	NaN	NaN	NaN	1.4825, 51.894444
56058	NaN	NaN	NaN	47.296388888900005, -11.584277777799999
56059	RJX7	NaN	NaN	130.270556, 30.784722

```
[56060 rows x 12 columns]
```

[29]:
```
#
#
rows, columns = data.shape
print('rows = {}; cols = {}'.format(rows, columns))
cat_cols = []
for col in data.columns:
    #
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        cat_cols.append(col)
        temp_perc = round((temp_null_count / rows) * 100.0, 2)
        print('     {}.        {}.                 {}, {}%.'.format(col, dt,␣
 ↪temp_null_count, temp_perc))
```

```
rows = 56060; cols = 12
     continent.        object.                 28063, 50.06%.
     iso_country.        object.                 246, 0.44%.
     municipality.        object.                 5789,
10.33%.
     gps_code.        object.                 14989, 26.74%.
     iata_code.        object.                 46825, 83.53%.
     local_code.        object.                 26984, 48.13%.
```

[30]:
```
#
def impute_objects(column_to_impute):
    imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
    data_imp2 = imp2.fit_transform(data[[column_to_impute]])
    return data_imp2
```

[45]:
```
data_imp1 = impute_objects('continent')
np.unique(data_imp1)
```

[45]:
```
array(['AF', 'AN', 'AS', 'EU', 'OC', 'SA'], dtype=object)
```

[46]:
```
data_imp2 = impute_objects('municipality')
np.unique(data_imp2)
```

[46]:
```
array(['"Big" Rock Flat', "'S Gravenvoeren", '108 Mile', …, 'Å½atec',
       'Å½ilina', 'Å½ocene'], dtype=object)
```

[47]:
```
data_imp3 = impute_objects('local_code')
np.unique(data_imp3)
```

[47]:
```
array(['-', '00A', '00AA', …, 'ZUN', 'ZWH', 'ZZV'], dtype=object)
```

<———————————————————————> <———————————————————————>

[35]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

[48]:
```
cat_enc = pd.DataFrame({'c1':data_imp1.T[0]})
cat_enc
```

[48]:

	c1
0	EU
1	EU
2	EU
3	EU
4	EU
…	..
56055	AS
56056	AS
56057	EU
56058	AF
56059	AS

[56060 rows x 1 columns]

[49]:
```
le = LabelEncoder()
cat_enc_le = le.fit_transform(cat_enc['c1'])
```

[50]:
```
cat_enc['c1'].unique()
```

[50]:
```
array(['EU', 'OC', 'AF', 'AN', 'AS', 'SA'], dtype=object)
```

[51]:
```
np.unique(cat_enc_le)
```

[51]:
```
array([0, 1, 2, 3, 4, 5])
```

- one-hot encoding

[66]:
```
cat_enc3 = pd.DataFrame({'c3':data_imp3.T[0]})
ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(cat_enc3[['c3']])
```

[67]:
```
cat_enc3.shape
```

[67]:
```
(56060, 1)
```

[68]:
```
cat_enc_ohe.shape
```

[68]:
```
(56060, 27747)
```

[69]:
```
cat_enc_ohe
```

[69]:
```
<56060x27747 sparse matrix of type '<class 'numpy.float64'>'
        with 56060 stored elements in Compressed Sparse Row format>
```

[70]:
```
cat_enc_ohe.todense()[0:10][0:10]
```

[70]:
```
matrix([[0., 1., 0., …, 0., 0., 0.],
        [0., 0., 1., …, 0., 0., 0.],
        [0., 0., 0., …, 0., 0., 0.],
        …,
        [0., 0., 0., …, 0., 0., 0.],
        [0., 0., 0., …, 0., 0., 0.],
        [0., 0., 0., …, 0., 0., 0.]])
```

[71]:
```
np.unique(cat_enc3)
```

[71]:
```
array(['-', '00A', '00AA', …, 'ZUN', 'ZWH', 'ZZV'], dtype=object)
```

Pandas get_dummies - one-hot

[72]:
```
cat_enc2 = pd.DataFrame({'c2':data_imp2.T[0]})
pd.get_dummies(cat_enc2).head()
```

[72]:

	c2_"Big" Rock Flat	c2_'S Gravenvoeren	c2_108 Mile	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	c2_26Â°56â 44.84â S 28Â°42â 47.79â E	c2_3 Marias	c2_;ifi,bwe	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	c2_<Olmos	c2_A. IbaÃ±ez	c2_Aachen	c2_Aalborg	…	c2_Å tip	\
0	0	0	0	0	…	0	
1	0	0	0	0	…	0	
2	0	0	0	0	…	0	
3	0	0	0	0	…	0	

```
4          0              0          0           0  …         0

   c2_Å tÄ tÃ  c2_Å umperk  c2_Å umvald  c2_Å»ernica  c2_Å»Å obek  \
0            0            0            0            0            0
1            0            0            0            0            0
2            0            0            0            0            0
3            0            0            0            0            0
4            0            0            0            0            0

   c2_Å½abljak Airport  c2_Å½atec  c2_Å½ilina  c2_Å½ocene
0                    0          0           0           0
1                    0          0           0           0
2                    0          0           0           0
3                    0          0           0           0
4                    0          0           0           0

[5 rows x 27510 columns]
```

[74]:
```
pd.get_dummies(data[['municipality']], dummy_na=True).head()
```

[74]:
```
   municipality_"Big" Rock Flat  municipality_'S Gravenvoeren  \
0                             0                             0
1                             0                             0
2                             0                             0
3                             0                             0
4                             0                             0

   municipality_108 Mile  municipality_26Â°56â 44.84â S 28Â°42â 47.79â E  \
0                      0                                              0
1                      0                                              0
2                      0                                              0
3                      0                                              0
4                      0                                              0

   municipality_3 Marias  municipality_;ifi,bwe  municipality_<Olmos  \
0                      0                      0                    0
1                      0                      0                    0
2                      0                      0                    0
3                      0                      0                    0
4                      0                      0                    0

   municipality_A. IbaÃ±ez  municipality_Aachen  municipality_Aalborg  …  \
0                       0                    0                     0  …
1                       0                    0                     0  …
2                       0                    0                     0  …
3                       0                    0                     0  …
4                       0                    0                     0  …
```

```
   municipality_Å tÄ tÃ  municipality_Å umperk  municipality_Å umvald  \
0                      0                      0                      0
1                      0                      0                      0
2                      0                      0                      0
3                      0                      0                      0
4                      0                      0                      0

   municipality_Å»ernica  municipality_Å»Å obek  \
0                      0                      0
1                      0                      0
2                      0                      0
3                      0                      0
4                      0                      0

   municipality_Å½abljak Airport  municipality_Å½atec  municipality_Å½ilina  \
0                              0                    0                     0
1                              0                    0                     0
2                              0                    0                     0
3                              0                    0                     0
4                              0                    0                     0

   municipality_Å½ocene  municipality_nan
0                     0                 0
1                     0                 0
2                     0                 0
3                     0                 0
4                     0                 0

[5 rows x 27511 columns]
```

<———————> <———————>

[75]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

MinMax

[82]:
```
data = pd.read_csv('FIFA 2018 Statistics.csv', sep=",")
data.head()
```

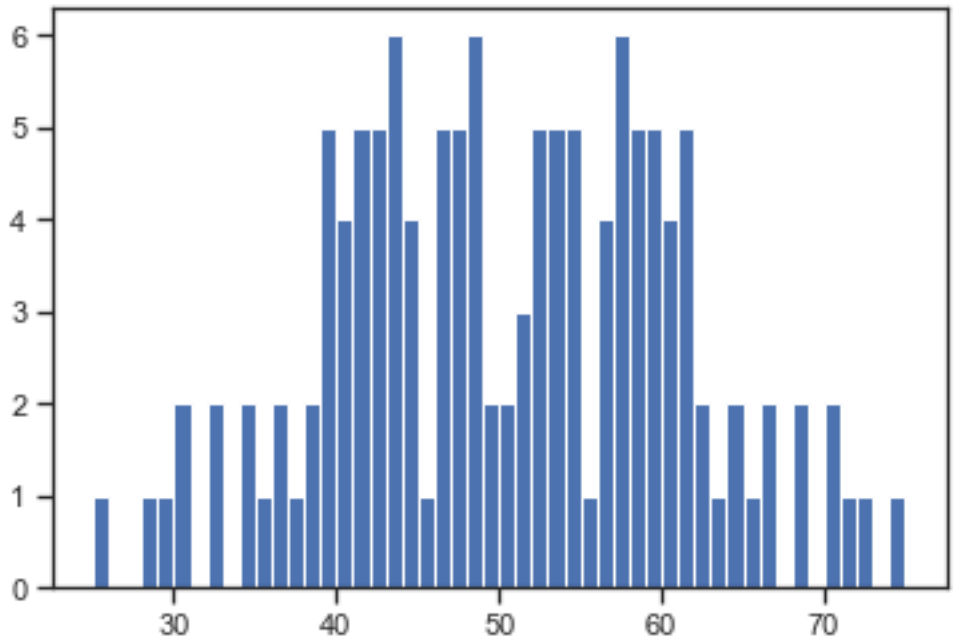
[82]:
```
         Date          Team      Opponent  Goal Scored  Ball Possession %  \
0  14-06-2018        Russia  Saudi Arabia            5                 40
1  14-06-2018  Saudi Arabia        Russia            0                 60
2  15-06-2018         Egypt       Uruguay            0                 43
3  15-06-2018       Uruguay         Egypt            1                 57
4  15-06-2018       Morocco          Iran            0                 64

   Attempts  On-Target  Off-Target  Blocked  Corners  …  Yellow Card  \
```

```
0        13          7           3        3        6  …            0
1         6          0           3        3        2  …            0
2         8          3           3        2        0  …            2
3        14          4           6        4        5  …            0
4        13          3           6        4        5  …            1

   Yellow & Red  Red  Man of the Match  1st Goal        Round  PSO  \
0             0    0               Yes      12.0  Group Stage   No
1             0    0                No       NaN  Group Stage   No
2             0    0                No       NaN  Group Stage   No
3             0    0               Yes      89.0  Group Stage   No
4             0    0                No       NaN  Group Stage   No

   Goals in PSO  Own goals  Own goal Time
0             0        NaN            NaN
1             0        NaN            NaN
2             0        NaN            NaN
3             0        NaN            NaN
4             0        1.0           90.0

[5 rows x 27 columns]
```

[83]:
```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['Ball Possession %']])
```

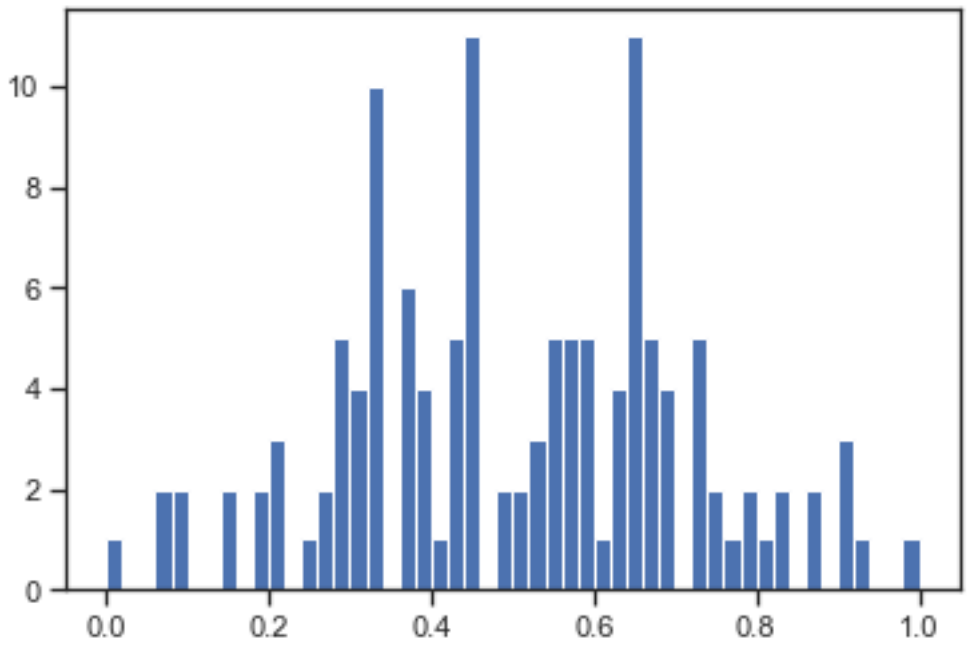
[84]:
```
plt.hist(data['Ball Possession %'], 50)
plt.show()
```

[84]:



[85]:
```
plt.hist(sc1_data, 50)
plt.show()
```

[85]:

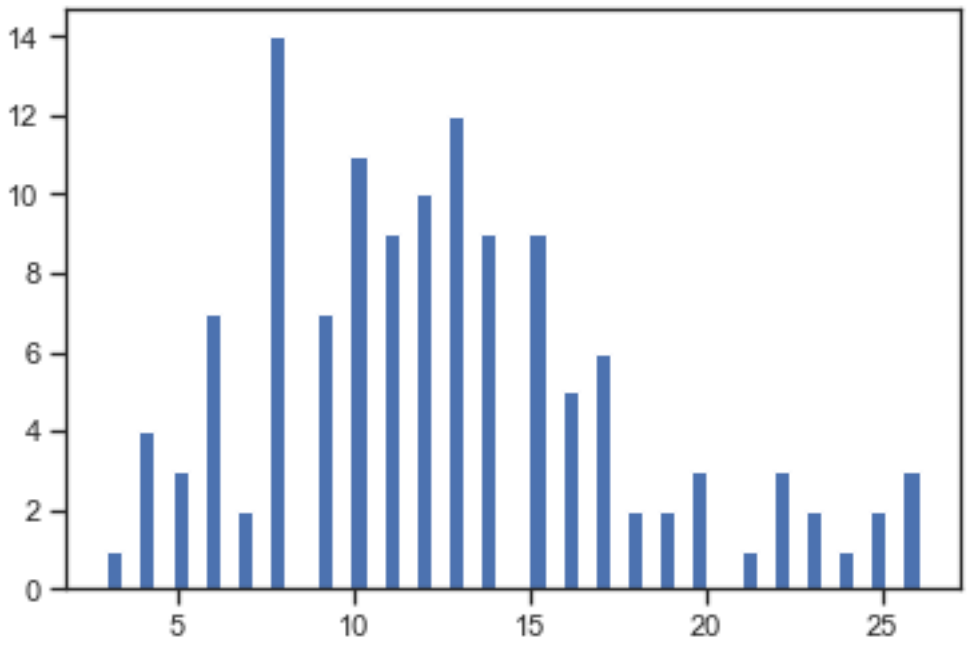


Z- - StandardScaler

[86]:
```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['Attempts']])
```

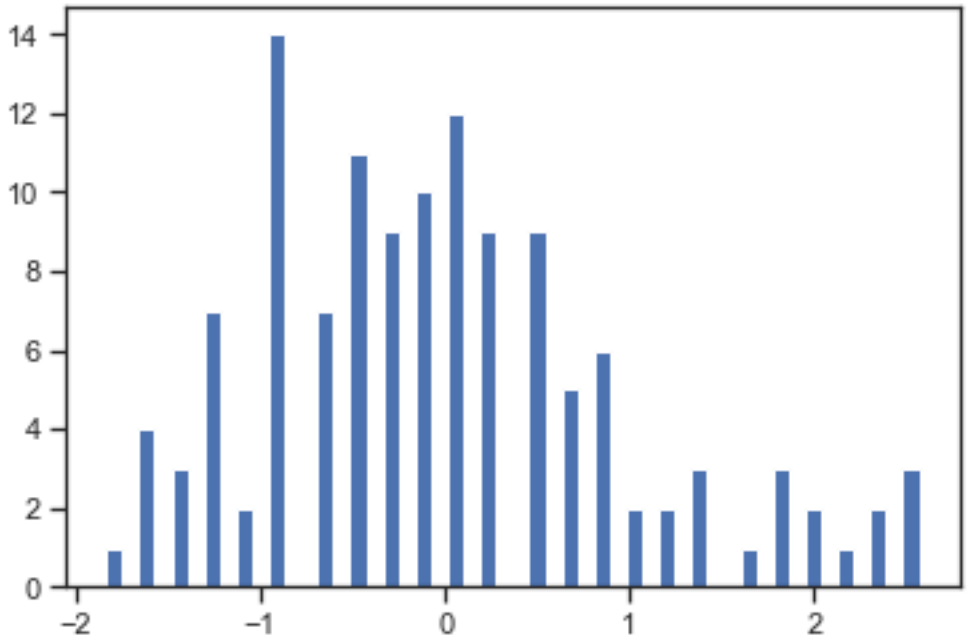
[88]:
```
plt.hist(data['Attempts'], 50)
plt.show()
```

[88]:



[87]:
```
plt.hist(sc2_data, 50)
plt.show()
```

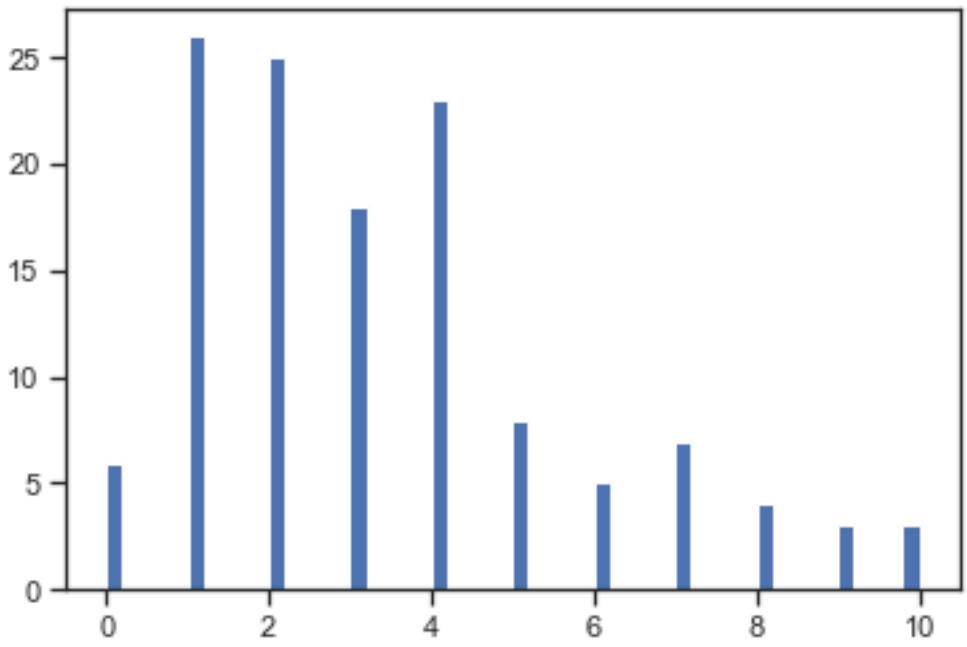
[87]:



[89]:
```
sc3 = Normalizer()
sc3_data = sc3.fit_transform(data[['Blocked']])
```

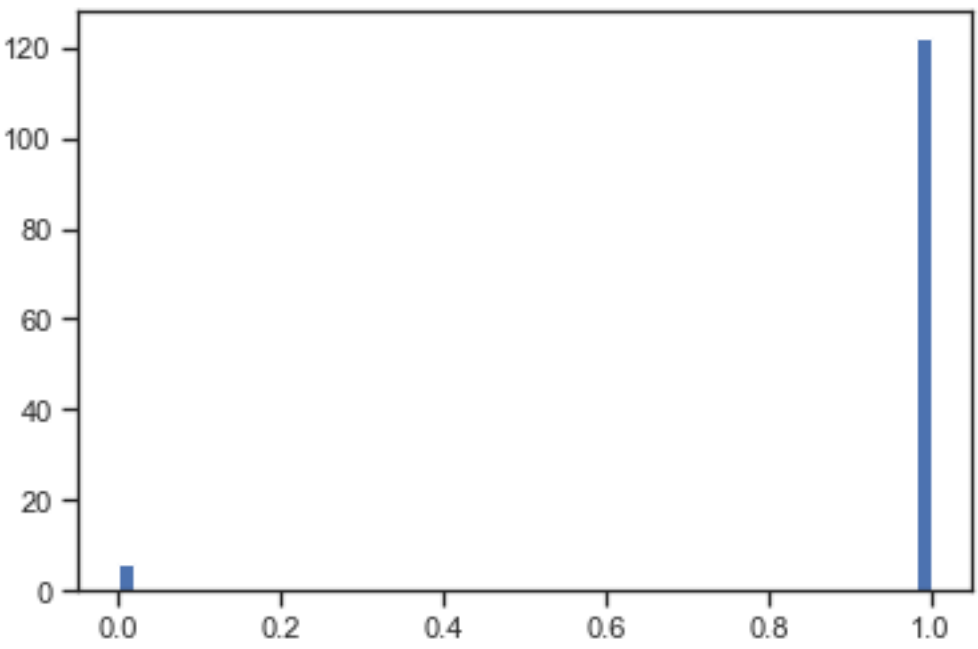
[90]:
```
plt.hist(data['Blocked'], 50)
plt.show()
```

[90]:



[91]:
```
plt.hist(sc3_data, 50)
plt.show()
```

[91]:



[0]: