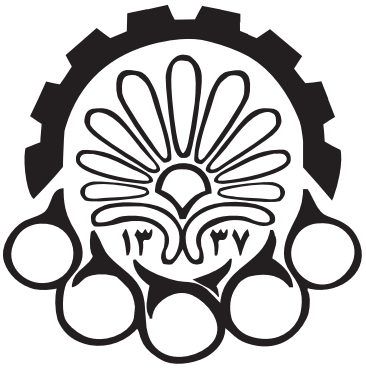


دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

پروژه ششم داده کاوی

گرایش بیوانفورماتیک

عنوان

مقایسه روش های مقابله با ناهنجاری در داده ها

نگارش

آرزو پاک سرشت

استاد راهنما

دکتر مهدی قطعی

دی ۱۴۰۰

چکیده

کلمات کلیدی:

فهرست مطالب

۱ ۲

۲ ۱۵

۳ ۱۶

۴ ۱۷

۵ نتیجه‌گیری ۱۸

فهرست تصاویر

۱.۱ . . . . . ۲

۲.۱ . . . . . ۳

۳.۱ . . . . . ۳

۴.۱ . . . . . ۴

۵.۱ . . . . . ۴

۶.۱ . . . . . ۵

۷.۱ . . . . . ۵

۸.۱ . . . . . ۵

۹.۱ . . . . . ۶

۱۰.۱ . . . . . ۶

۱۱.۱ . . . . . ۷

۱۲.۱ . . . . . ۷

۱۳.۱ . . . . . ۸
۱۴.۱ . . . . . ۸
۱۵.۱ . . . . . ۹
۱۶.۱ . . . . . ۹
۱۷.۱ . . . . . ۱۰
۱۸.۱ . . . . . ۱۰
۱۹.۱ . . . . . ۱۱
۲۰.۱ . . . . . ۱۱
۲۱.۱ . . . . . ۱۲
۲۲.۱ . . . . . ۱۲
۲۳.۱ . . . . . ۱۳
۲۴.۱ . . . . . ۱۳
۲۵.۱ . . . . . ۱۳
۲۶.۱ . . . . . ۱۴
۲۷.۱ . . . . . ۱۴

فهرست جداول

فصل ۱

شکل ۱.۱:

```
[ ] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    from sklearn.decomposition import PCA
    from sklearn.preprocessing import RobustScaler
    from sklearn.cluster import DBSCAN
    from sklearn.ensemble import IsolationForest
    from sklearn.metrics import classification_report, roc_auc_score, confusion_matrix
    from statsmodels.tsa.seasonal import seasonal_decompose
    import keras
    from keras import layers
    import utils
```

شکل ۲.۱:

```
[ ] SEED = 1291
```

```
satellite = pd.read_csv("satellite.mat.csv")
print(satellite.shape)
features = [c for c in satellite.columns if c.startswith("V")]
satellite.head()
```

```
(6435, 38)
```

	ID	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25
0	0	92	115	120	94	84	102	106	79	84	102	102	83	101	126	133	103	92	112	118	85	84	103	104	81	102	126
1	1	84	102	106	79	84	102	102	83	80	102	102	79	92	112	118	85	84	103	104	81	84	99	104	78	88	121
2	2	84	102	102	83	80	102	102	79	84	94	102	79	84	103	104	81	84	99	104	78	84	99	104	81	84	107
3	3	80	102	102	79	84	94	102	79	80	94	98	76	84	99	104	78	84	99	104	81	76	99	104	81	84	99
4	4	84	94	102	79	80	94	98	76	80	102	102	79	84	99	104	81	76	99	104	81	76	99	108	85	84	99

```
satellite.drop('ID',
  axis='columns', inplace=True)
```

```
[ ] satellite.describe()
```

شکل ۳.۱:

```
[ ] satellite.drop('ID',
      axis='columns', inplace=True)
```

```
satellite.describe()
```

	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9
count	6435.000000	6435.000000	6435.000000	6435.000000	6435.000000	6435.000000	6435.000000	6435.000000	6435.000000	6435.000000
mean	69.400000	83.594872	99.290598	82.592696	69.150272	83.243512	99.110645	82.497125	68.912354	82.893085
std	13.605871	22.882234	16.645944	18.897674	13.561197	22.886495	16.664088	18.940923	13.470599	22.862255
min	39.000000	27.000000	53.000000	33.000000	39.000000	27.000000	50.000000	29.000000	40.000000	27.000000
25%	60.000000	71.000000	85.000000	69.000000	60.000000	71.000000	85.000000	69.000000	60.000000	71.000000
50%	68.000000	87.000000	101.000000	81.000000	68.000000	85.000000	101.000000	81.000000	67.000000	85.000000
75%	80.000000	103.000000	113.000000	92.000000	80.000000	103.000000	113.000000	92.000000	79.000000	102.000000
max	104.000000	137.000000	140.000000	154.000000	104.000000	137.000000	145.000000	157.000000	104.000000	130.000000

شکل ۴.۱:

```
[ ] satellite.isnull().sum()
```

```
V0     0
V1     0
V2     0
V3     0
V4     0
V5     0
V6     0
V7     0
V8     0
V9     0
V10    0
V11    0
V12    0
V13    0
V14    0
V15    0
V16    0
V17    0
V18    0
V19    0
V20    0
V21    0
V22    0
V23    0
V24    0
V25    0
V26    0
V27    0
V28    0
V29    0
V30    0
V31    0
```

شکل ۵.۱:

Preprocessing

```
[109] scaler = RobustScaler()

      train_org_scaled = scaler.fit_transform(train_org)

      fig, ax = plt.subplots(figsize=(24,4))
      pd.DataFrame(train_org.values[-400:,1:5]).plot(ax=ax, legend=False)
      plt.show()

      fig, ax = plt.subplots(figsize=(24,4))
      pd.DataFrame(train_org_scaled[-400:,1:5]).plot(ax=ax, legend=False)
      plt.show()

      fig, ax = plt.subplots(figsize=(24,2))
      satellite["Y"].iloc[-400:].plot(ax=ax)
      plt.show()
```

شکل ۶.۱:

شکل ۷.۱:

```
pca = PCA()
df_pca = pd.DataFrame(pca.fit_transform(train_org_scaled))
df_pca
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1.309643	-4.479017	-0.625169	-0.999227	0.827347	1.512994	0.286183	0.215897	-0.180444	-0.075130	0.260824	-0.282958	0.316343
1	0.192888	-3.272389	-0.438224	-0.693622	0.398241	0.807270	0.145533	0.611144	-0.136862	-0.207024	0.162261	0.177802	0.099302
2	-0.319680	-2.681657	-0.527484	-0.194807	0.125304	0.210088	-0.012750	0.187102	-0.001479	-0.113041	0.051318	0.234624	0.160579
3	-0.541547	-2.421579	-0.496834	-0.148119	0.184124	0.022355	0.082030	-0.134200	0.144788	0.060903	-0.145898	0.125372	0.031639
4	-0.278470	-2.309121	-0.235061	-0.323044	0.186225	-0.282829	0.064878	-0.132609	-0.222949	0.049462	-0.185844	-0.017391	0.299554
...	...	...	...	...	...	...	...	...	...	...	...	...	...
5430	0.765504	0.286239	0.648973	-0.255301	0.167223	-0.156386	-0.019307	0.262706	-0.124209	-0.176798	-0.050155	0.100140	-0.273444
5431	0.331220	1.248151	0.439854	-0.240975	0.699843	0.195264	-0.129260	-0.124194	0.131252	0.032350	-0.170893	0.033729	-0.028670
5432	-0.497741	1.715838	0.517331	-0.317972	0.843332	0.227585	-0.467897	0.061113	0.080956	0.014107	0.099257	0.011984	-0.120246
5433	-0.431171	1.475170	0.488530	0.393579	0.538781	-0.923558	0.190830	0.743066	0.235887	0.147641	0.200122	0.176530	0.017991
5434	0.693649	1.405599	0.345372	0.599478	0.690461	-1.191941	0.363475	-0.079407	0.112838	0.339183	0.078035	0.174259	-0.002108

5435 rows × 35 columns

شکل ۸.۱:

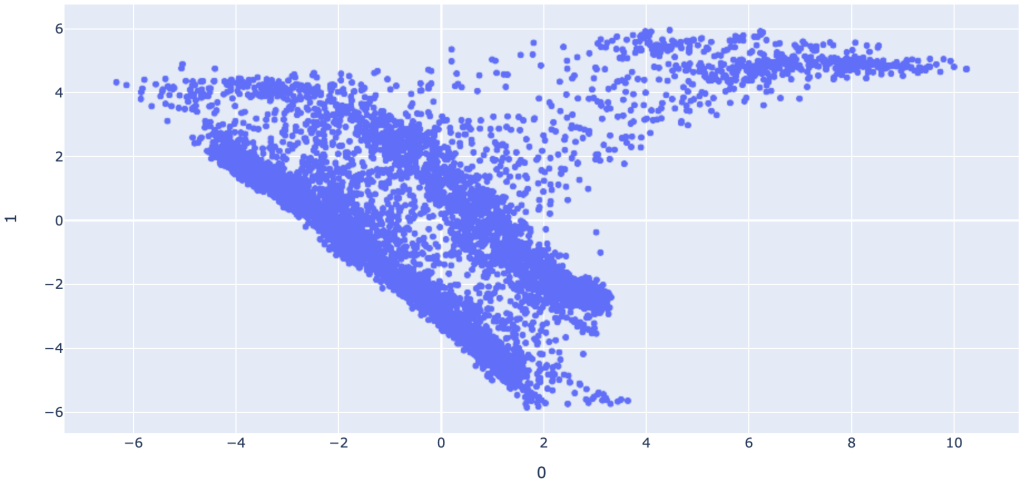
```
import plotly.express as px
from sklearn.decomposition import PCA

df = px.data.iris()
X =train_org_scaled

pca = PCA(n_components=2)
components = pca.fit_transform(X)

fig = px.scatter(components, x=0, y=1)
fig.show()
```

شکل ۹.۱:



شکل ۱۰.۱:

Outlier

```
import numpy as np
import pandas as pd
from scipy import stats
#import eif as iso
from sklearn import svm
from sklearn.cluster import DBSCAN
from sklearn.ensemble import IsolationForest
from sklearn.neighbors import LocalOutlierFactor
import matplotlib.dates as md
from scipy.stats import norm
%matplotlib inline
import seaborn as sns
sns.set_style("whitegrid") #possible choices: white, dark, whitegrid, darkgrid, ticks
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly import tools
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
pd.set_option('float_format', '{:f}'.format)
pd.set_option('max_columns',250)
pd.set_option('max_rows',150)
```

شکل ۱۱.۱:

Isolation Forest

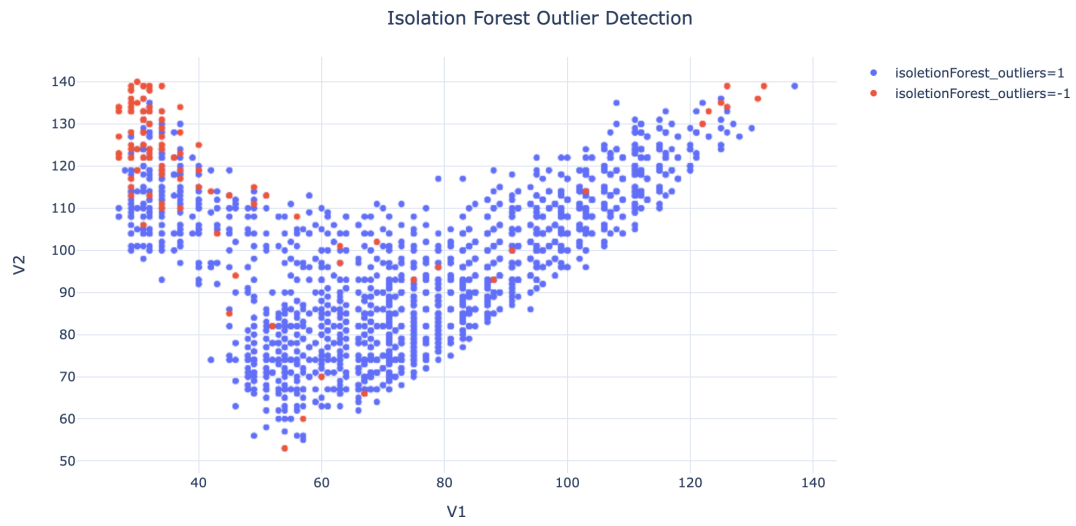
```
clf = IsolationForest(max_samples='auto', random_state = 1, contamination= 0.02)
preds = clf.fit_predict(X)
satellite['isoletionForest_outliers'] = preds
satellite['isoletionForest_outliers'] = satellite['isoletionForest_outliers'].astype(str)
satellite['isoletionForest_scores'] = clf.decision_function(X)
print(satellite['isoletionForest_outliers'].value_counts())
satellite[152:156]
```

V21	V22	V23	V24	V25	V26	V27	V28	V29	V30	V31	V32	V33	V34	V35	Y	isoletionForest_outliers	isoletionForest_scores
104	108	92	92	108	110	86	92	108	110	86	92	103	105	86	0	1	0.164733
109	108	89	92	108	110	86	92	103	105	86	87	103	105	83	0	1	0.167824
104	112	85	92	103	105	86	87	103	105	83	92	103	110	83	0	1	0.168027
104	104	81	87	103	105	83	92	103	110	83	92	103	110	86	0	1	0.167299

```
[54] x1='V1'
     x2='V2'
     X = satellite[[x1,x2]]
```

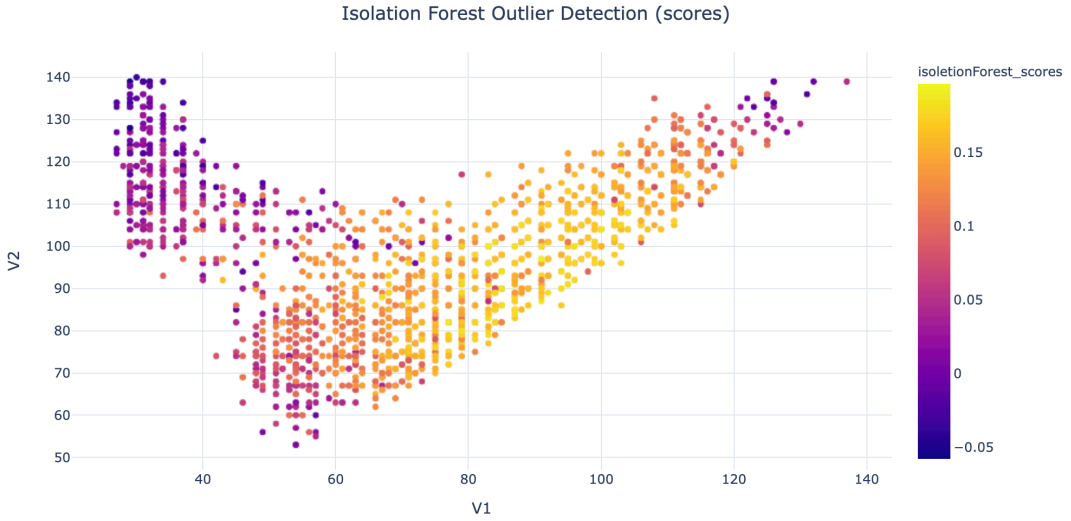
شکل ۱۲.۱:

```
[110] fig = px.scatter(satellite
                       , x=x1, y=x2, color='isoletionForest_outliers')
      fig.update_layout(title='Isolation Forest Outlier Detection', title_x=0.5, yaxis=dict(gridcolor = '#DFEAF4'), xaxis=dict
      fig.show()
```



شکل ۱۳.۱:

```
fig = px.scatter(satellite, x=x1, y=x2, color="isoletionForest_scores")
fig.update_layout(title='Isolation Forest Outlier Detection (scores)', title_x=0.5,yaxis=dict(gridcolor = '#DFEAF4'), xa
fig.show()
```

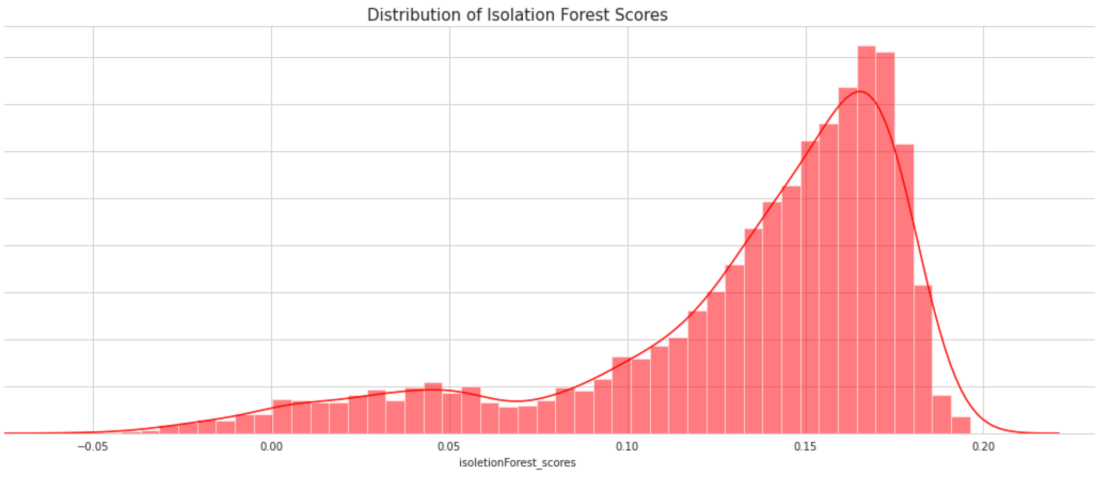


شکل ۱۴.۱:

```
[48] fig, ax = plt.subplots(figsize=(20, 7))
     ax.set_title('Distribution of Isolation Forest Scores', fontsize = 15, loc='center')
     sns.distplot(satellite['isoletionForest_scores'],color='red',label='if',hist_kws = {"alpha": 0.5});
```

```
/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:

a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-le
```



شکل ۱۵.۱:

Local Outlier Factor

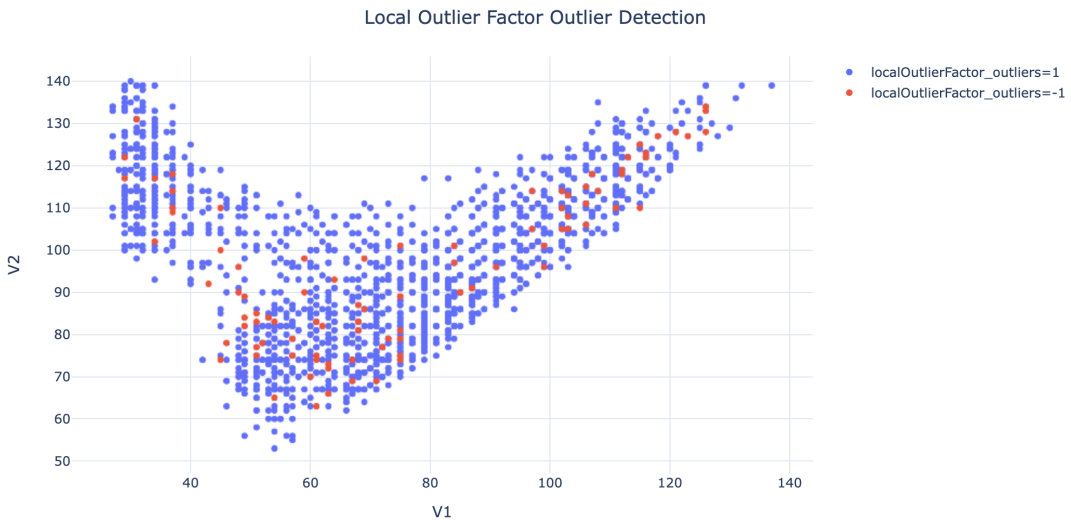
```
clf = LocalOutlierFactor(n_neighbors=11)
X =train_org_scaled
y_pred = clf.fit_predict(X)
satellite['localOutlierFactor_outliers'] = y_pred.astype(str)
print(satellite['localOutlierFactor_outliers'].value_counts())
satellite['localOutlierFactor_scores'] = clf.negative_outlier_factor_
satellite[152:156]
```

```
1      6338
-1       97
Name: localOutlierFactor_outliers, dtype: int64
```

	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25
152	88	107	109	92	88	107	109	87	88	107	109	87	90	104	112	89	86	104	108	89	90	104	108	92	92	108
153	88	107	109	87	88	107	109	87	88	107	109	87	86	104	108	89	90	104	108	92	90	109	108	89	92	108
154	88	107	109	87	88	107	109	87	88	103	109	87	90	104	108	92	90	109	108	89	86	104	112	85	92	103
155	88	107	109	87	88	103	109	87	93	103	109	87	90	109	108	89	86	104	112	85	86	104	104	81	87	103

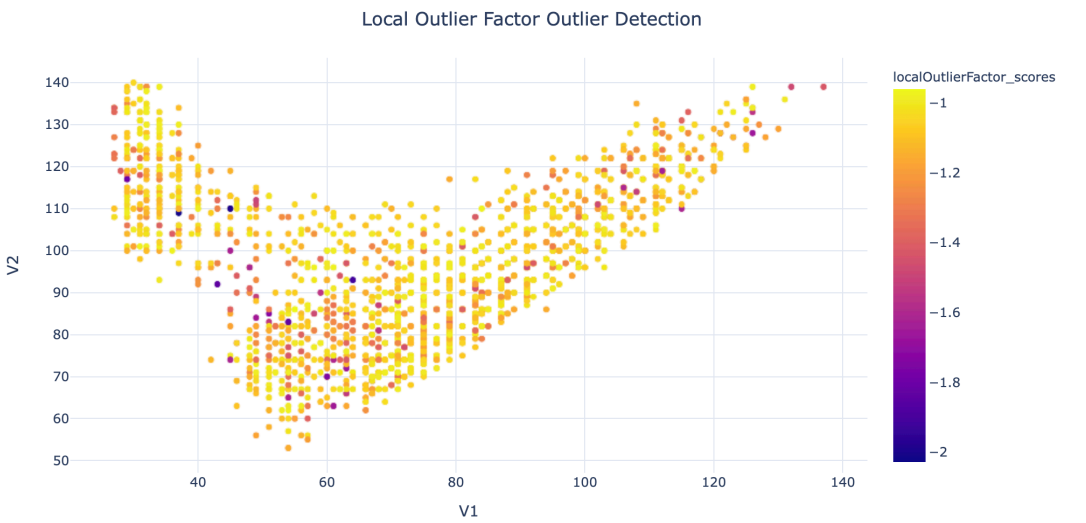
شکل ۱۶.۱:

```
fig = px.scatter(satellite, x=x1, y=x2, color='localOutlierFactor_outliers')
fig.update_layout(title='Local Outlier Factor Outlier Detection', title_x=0.5, yaxis=dict(gridcolor = '#DFEAF4'), xaxis=
fig.show()
```



شکل ۱۷.۱:

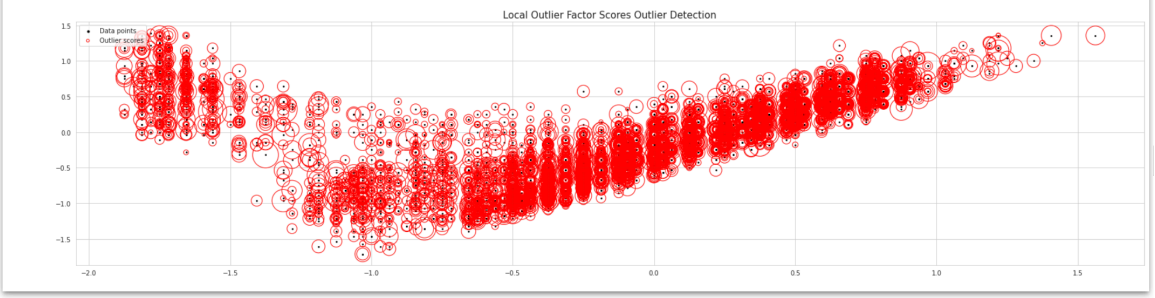
```
[118] fig = px.scatter(satellite, x=x1, y=x2, color="localOutlierFactor_scores")
      fig.update_layout(title='Local Outlier Factor Outlier Detection', title_x=0.5,yaxis=dict(gridcolor = '#DFEAF4'), xaxis=d
      fig.show()
```



شکل ۱۸.۱:

```
fig, ax = plt.subplots(figsize=(30, 7))
ax.set_title('Local Outlier Factor Scores Outlier Detection', fontsize = 15, loc='center')

plt.scatter(X[:, 0], X[:, 1], color='k', s=3., label='Data points')
radius = (satellite['localOutlierFactor_scores'].max() - satellite['localOutlierFactor_scores']) / (satellite['localOutl:
plt.scatter(X[:, 0], X[:, 1], s=2000 * radius, edgecolors='r', facecolors='none', label='Outlier scores')
plt.axis('tight')
legend = plt.legend(loc='upper left')
legend.legendHandles[0]._sizes = [10]
legend.legendHandles[1]._sizes = [20]
plt.show();
```

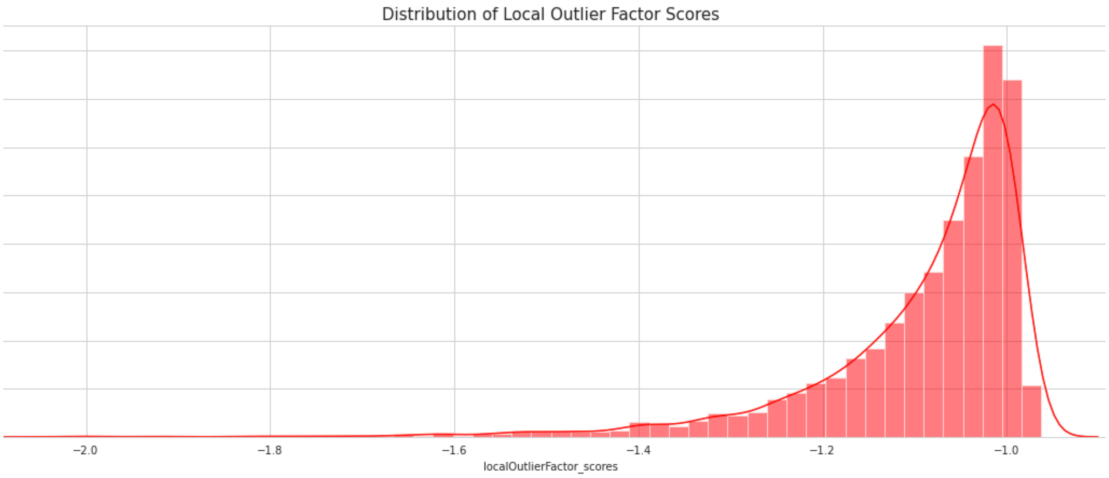


شکل ۱.۱۹:

```
[73] fig, ax = plt.subplots(figsize=(20, 7))
     ax.set_title('Distribution of Local Outlier Factor Scores', fontsize = 15, loc='center')
     sns.distplot(satellite['localOutlierFactor_scores'],color='red',label='eif',hist_kws = {"alpha": 0.5});
```

```
/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:

 is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figur
```



شکل ۱.۲۰:

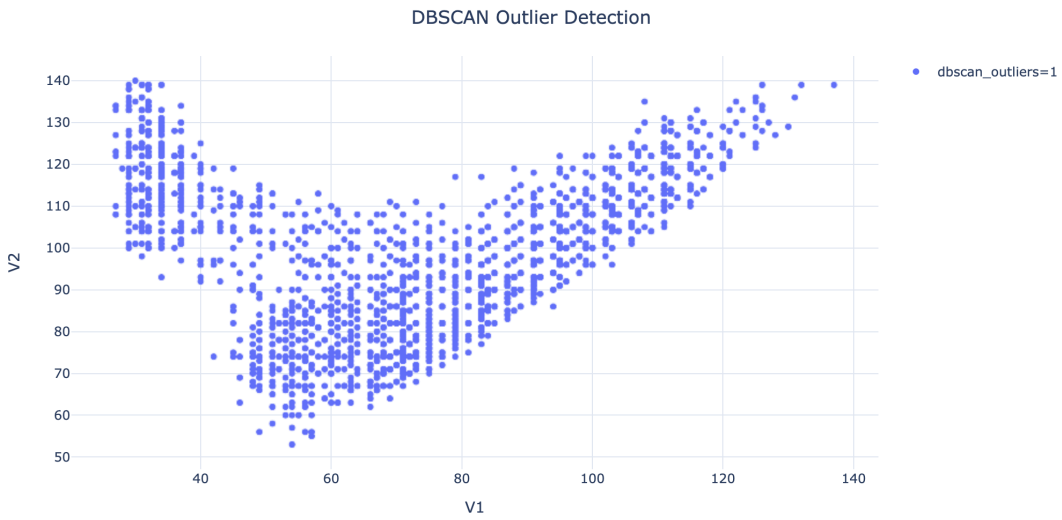
DBSCAN

```
from sklearn.cluster import DBSCAN
outlier_detection = DBSCAN(eps = 20, metric='euclidean', min_samples = 5,n_jobs = -1)
X =train_org_scaled
clusters = outlier_detection.fit_predict(X)
satellite['dbscan_outliers'] = clusters
satellite['dbscan_outliers'] = satellite['dbscan_outliers'].apply(lambda x: str(1) if x>-1 else str(-1))
print(satellite['dbscan_outliers'].value_counts())
```

```
1    6435
Name: dbscan_outliers, dtype: int64
```

شکل ۱.۲۱:

```
fig = px.scatter(satellite, x=x1, y=x2, color="dbscan_outliers")
fig.update_layout(title='DBSCAN Outlier Detection', title_x=0.5,yaxis=dict(gridcolor = '#DFEAF4'), xaxis=dict(gridcolor
fig.show()
```



شکل ۱.۲۲:

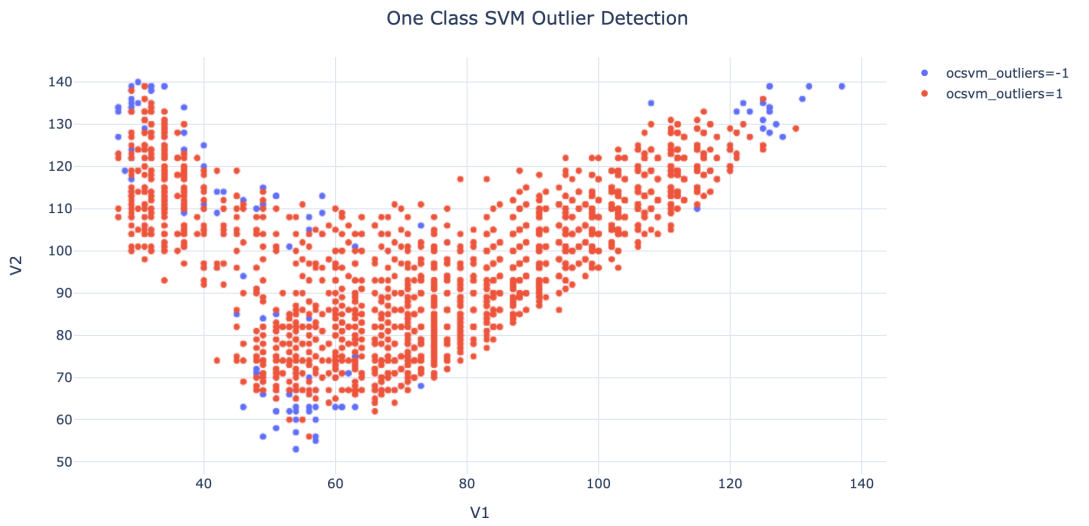
One Class SVM

```
clf = svm.OneClassSVM(nu=0.07,kernel='rbf',gamma='auto')
outliers = clf.fit_predict(X)
satellite['ocsvm_outliers'] = outliers
satellite['ocsvm_outliers'] = satellite['ocsvm_outliers'].apply(lambda x: str(-1) if x==-1 else str(1))
satellite['ocsvm_scores'] = clf.score_samples(X)
print(satellite['ocsvm_outliers'].value_counts())
```

```
1     5984
-1     451
Name: ocsvm_outliers, dtype: int64
```

شکل ۱.۲۳:

```
fig = px.scatter(satellite, x=x1, y=x2, color="ocsvm_outliers")
fig.update_layout(title='One Class SVM Outlier Detection', title_x=0.5,yaxis=dict(gridcolor = '#DFEAF4'), xaxis=dict(gri
fig.show()
```



شکل ۱.۲۴:

Ensemble

```
satellite['outliers_sum'] = satellite['isoletionForest_outliers'].astype(int)
+satellite['localOutlierFactor_outliers'].astype(int)
+satellite['dbscan_outliers'].astype(int)
+satellite['ocsvm_outliers'].astype(int)
```

```
satellite['outliers_sum'].value_counts()
```

```
 4    5904
 2     387
 0     142
-2       2
Name: outliers_sum, dtype: int64
```

شکل ۱.۲۵:

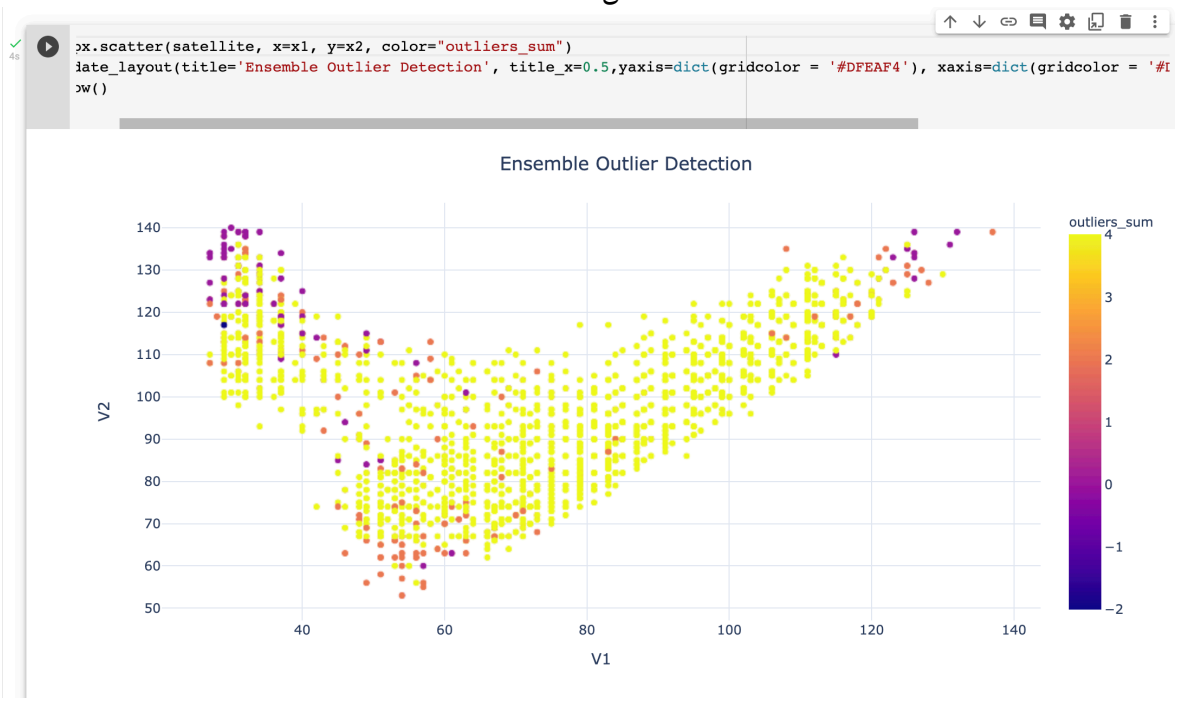
Ensemble

```
[142] satellite['outliers_sum'] = satellite['isoletionForest_outliers'].astype(int)
      +satellite['localOutlierFactor_outliers'].astype(int)
      +satellite['dbscan_outliers'].astype(int)
      +satellite['ocsvm_outliers'].astype(int)
```

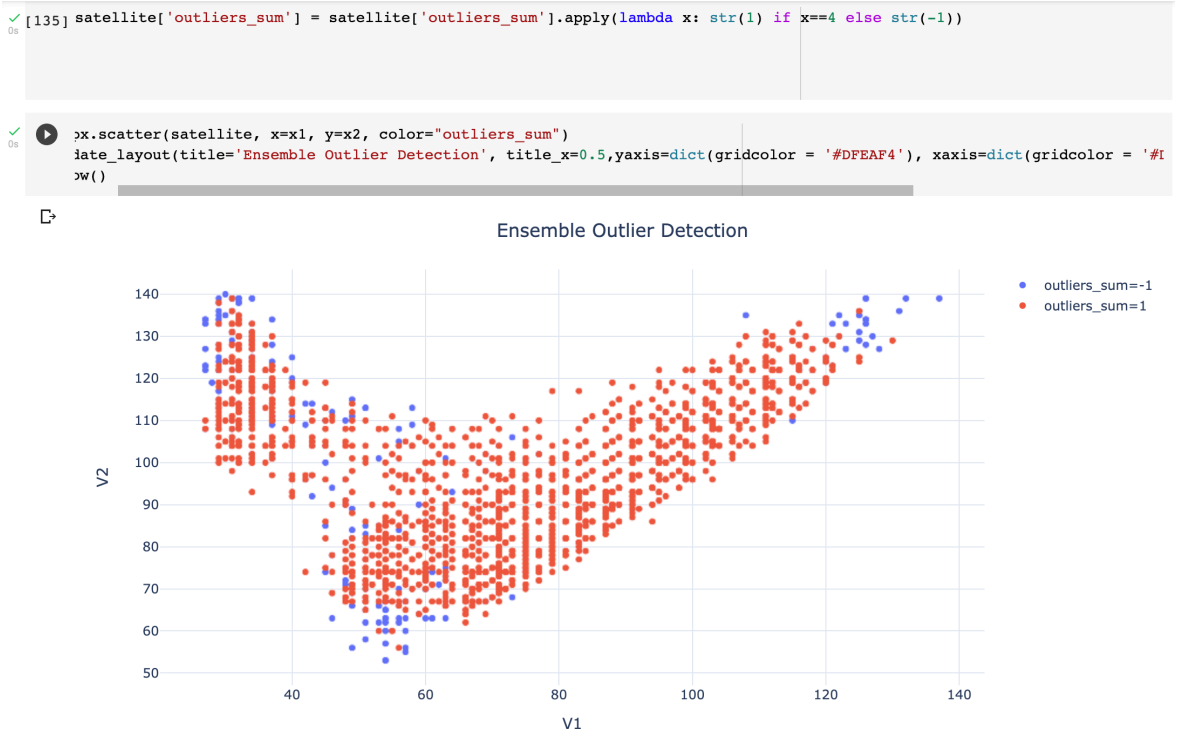
```
[143] satellite['outliers_sum'].value_counts()
```

```
 4    5904
 2     387
 0     142
-2       2
Name: outliers_sum, dtype: int64
```

شکل ۱.۲۶:



شکل ۱.۲۷:



منابع و مراجع

[1] https://www.kaggle.com/mineshjethva/anomaly-detection-lstm-isolation-forest/notebook

[2]https://github.com/Deffro/Data-Science-Portfolio/blob/master/Notebooks/Outlier%20Detection/Outlier%20Detection%20-%20Theory%2C%20Visualizations%20and%20Code.ipynb

[3]https://towardsdatascience.com/outlier-detection-theory-visualizations-and-code-a4fd39de540c

[4]https://towardsdatascience.com/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57

[5]https://towardsdatascience.com/visualizing-high-dimensional-data-f59eab85f08b

[6]https://plotly.com/python/pca-visualization/