Assignment 3

Outlier Detection using LOF and Isolation Forest

SATELLITE DATASET

Presented to

Subair Abayomi Oloko

In Partial Fulfillment of the requirements for the course

BI and Data Analytics

CST8390

By

Abdullah Zeki Ilgun 040991363

Elaiza Rivera 040839516

April 7, 2021

Introduction

Anomaly detection or outlier detection has been of great interest to detect unusual behavior through the given datasets. Which have two important characteristics. First, that anomalies are different from the norm and second, that dataset are rare compared to the normal instances. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data (plos.org) In many real-world setups, outlier detection solely focused on the internal structure of the data and is commonly unlabeled, thus it is called unsupervised.

The satellite dataset is one of the examples which we will work on this research. Satellite dataset is the behaviour recorded from the satellite observations. The original dataset's task is to classify soil category from the captured images, which were capture in four different wavelengths of lights, two invisible light (green and red) and the two infrared images. With these factors soil types can be determined whether they are "red soil", "gray soil", "damp dray soil", "very damp gray soil" OpenML Satellite.

Compared to supervised anomaly detection, unsupervised tend to be the most flexible setup when detecting outliers, for does not need labels. Thus, this type of method was used commonly with the satellite datasets. It focuses more on the density and intrinsic properties of the given dataset rather than the clusters. With this assignment, our focus will then be recognizing which records deviate from the norms. We will compare our result from the study that was conducted and see how close we can get on this study.

Analyzing Data and its attributes

Since we are doing unsupervised learning, labelling of the columns in our record is not as important when doing supervised learning. Thus, the downloaded dataset with its original form was not labeled. Yet it is confusing when researchers do not have an idea what are these records are about. As mentioned above, these datasets are image behavior captured by satellites to classify soil types. Our task is to detect rare behaviours. For the sake of this study, us the researcher will label these columns with Label 1 through Label 36 with Label 37 as the classification whether the records are either a norm or an anomaly. Label 1 through Label 36 are numeric, and Label 37 is nominal datatypes.

Loading to Weka

Before loading the file to Weka, data for all the attributes are skimmed to make sure almost all of them are numeric. As there are no specific information about the attributes, they are simply labeled by “Label1”, “Label2” and so on till “Label31”. All the attributes except for the last one is set to be numeric, and the last one is set to be nominal with the values or ‘o’ and ‘n’. That attribute later will be used as the class attribute for the analysis. Below are the statistics for each attribute regarding their min, max, mean and standard deviation values respectively.

Statistics and Count

The table below tells us the minimum, maximum, mean and stddev of each attribute respectively.

Label No.	Stats	Minimum	Maximum	Mean	StdDev
1		41	104	73.092	11.876
2		28	137	91.578	16.039
3		62	139	99.569	15.611
4		44	144	79.827	13.514
5		42	104	72.958	11.722
6		29	137	91.426	15.823
7		62	139	99.358	15.586
8		46	157	76.644	13.447
9		42	104	72.608	11.716
10		29	130	90.862	16.071
11		60	140	98.978	15.588
12		34	150	79.473	13.623

13	41	104	73.026	11.859
14	27	137	91.602	15.793
15	62	139	99.454	15.735
16	44	150	79.651	13.538
17	41	104	72.956	11.728
18	27	130	91.63	15.565
19	62	139	99.328	15.686
20	44	146	79.546	13.565
21	42	104	72.657	11.659
22	29	130	91.149	15.71
23	60	138	99.023	15.712
24	34	150	79.375	13.66
25	40	104	72.641	11.979
26	27	131	91.044	16.218
27	60	139	99.245	15.722
28	44	151	79.696	13.75
29	41	104	72.591	11.815
30	27	130	91.188	15.916
31	62	139	99.207	15.736
32	42	151	79.615	13.626
33	40	104	72.358	11.711
34	27	130	90.857	15.921
35	50	138	98.985	15.712
36	29	147	79.493	13.707
37				

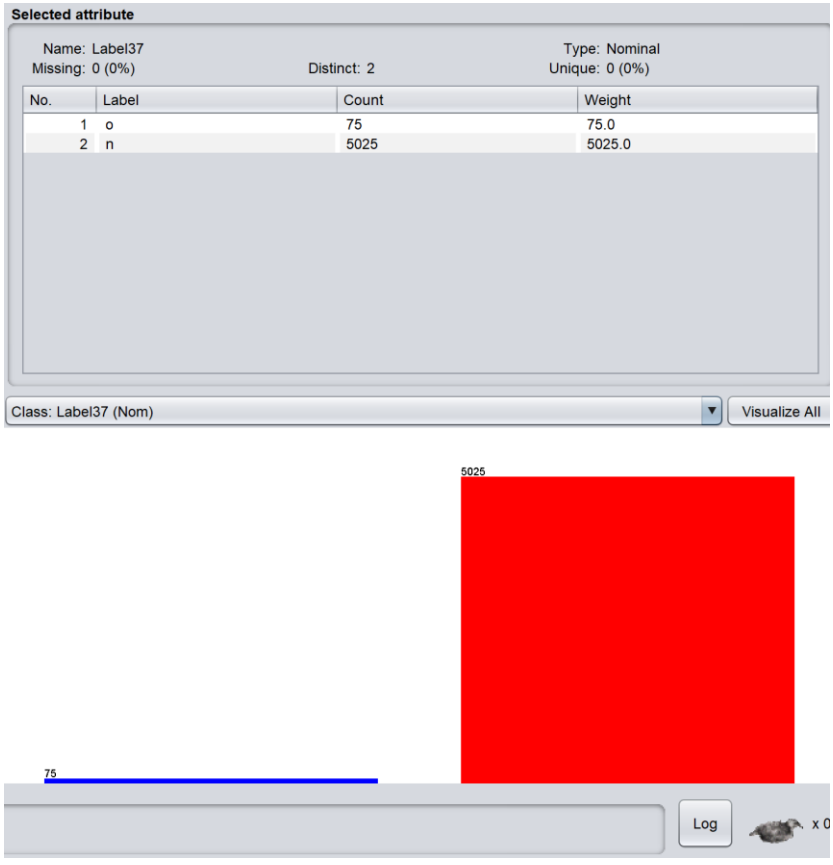
From the table, we can see that the minimum values are between 27 and 62, maximum values are 104 to 157, mean values are 72.358 to 99.569 and lastly the standard deviation values are having values from 11.611 to 16.218.

Preprocessing

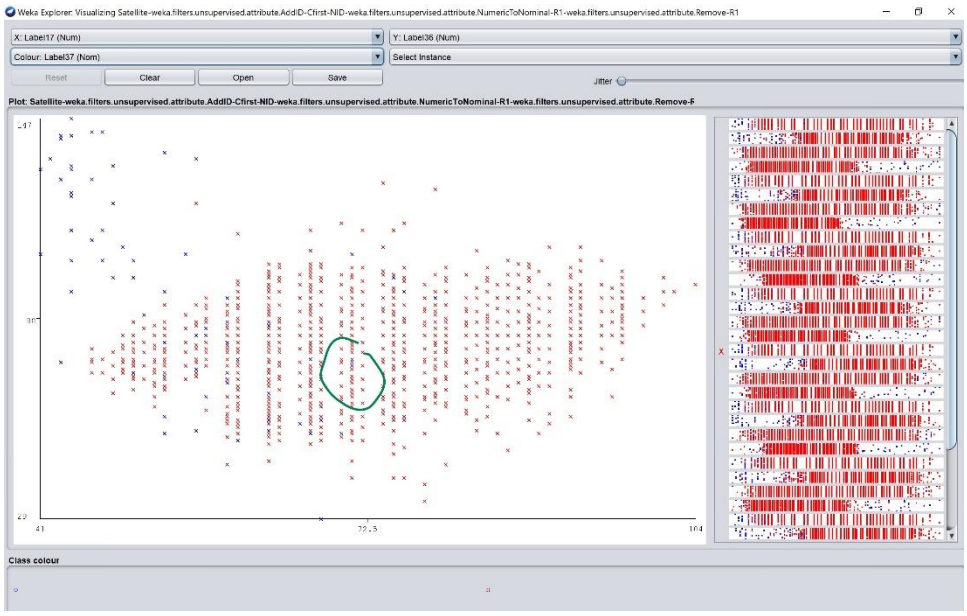
The data set was fairly clean when it is extracted from the website. It was tried to see if there are any duplicate rows on Excel. Firstly, the rows' all the values are combined into one row, then the cells with the duplicate values are formatted using Excels' formatting feature. After all there did not seem any duplicate rows in the set. There were also no missing or wrongly typed data as the file could already be loaded to Weka.

Data Visualization

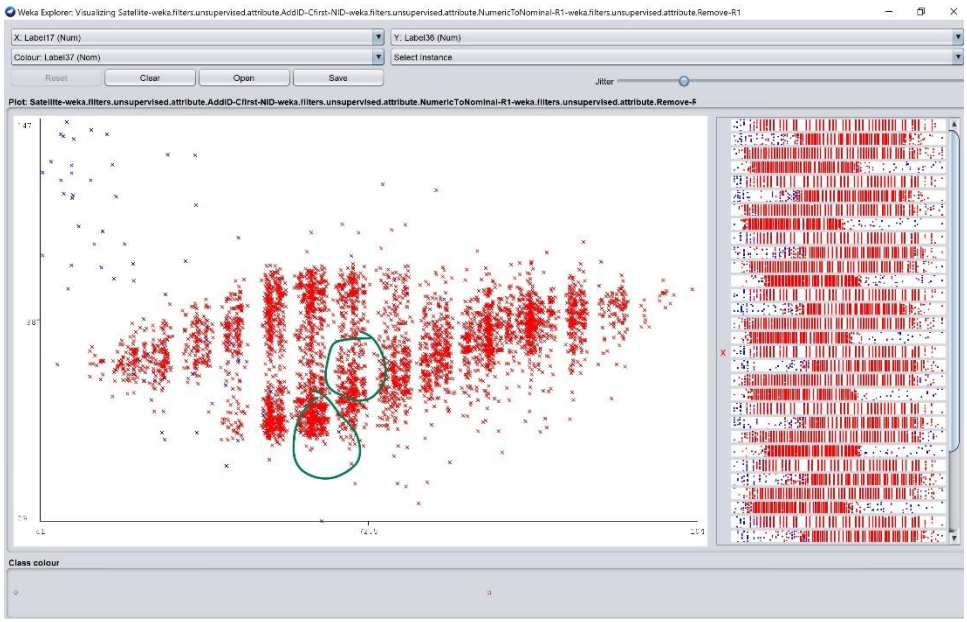
In this section we will investigate the outliers detected from the original data. To begin, we have 5025 normal data detected and 75 count for outliers. We will randomly select a set of coordinates to see where these outliers are seated basing on the X and Y points plotted using the Visualize tab on Weka.



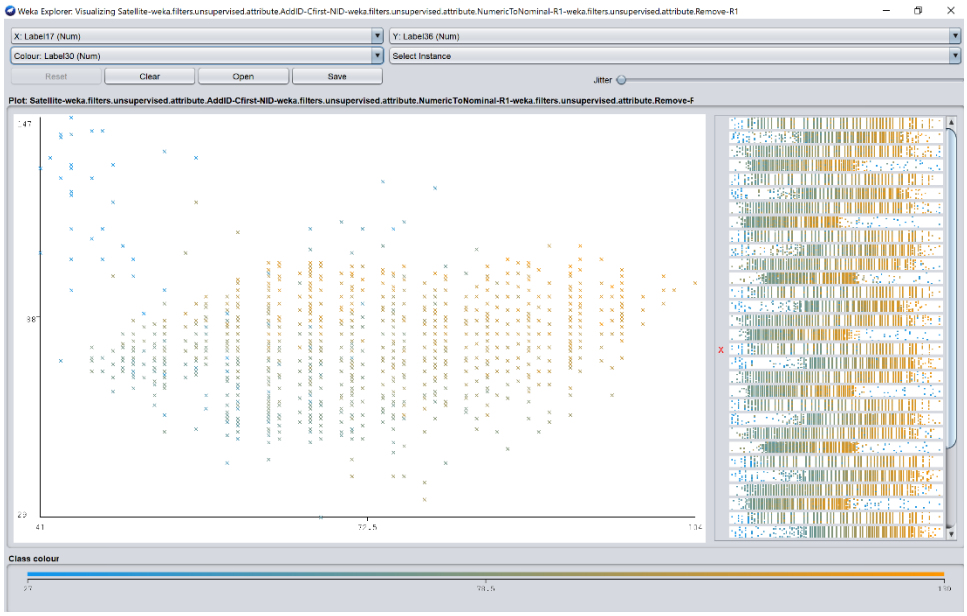
First chart we selected with X – Label 17 and Y – Label 36. The plots are not too crowded so we can see clearly where the outlier sits on the graph. We see that some outliers are sitting close to normal data. Comparing it to the outliers that obviously far from the norm.



If we increase the jitter to this chart, we can see that there are more normal data that is very close to the outlier. By looking only on their distance, how these points become an outlier when they are surrounded by the norm? Shouldn't it be that they are clustered?



When using another attribute, for example on this chart, instead of the Label 37 but Label 36. We see that some factor or value of the data can determine a reason as to why these points though close to the normal points are an outlier.



Clustering using K-Means

"kMeans" method was performed on the data set to cluster the values. The algorithm was run with different 'k' values, which means the number of clusters, to determine the most accurate result by seeing the different results. Testing method is set to be classes to clusters evaluation with Label37. Below table shows the number of iterations, the number and percentage of incorrectly clustered instances and clustered instances for each 'k' values.

K value	Stats	Number of Iterations	Incorrectly Clustered Instances (Num - %)	Clustered Instances
2		7	2374 – 46.549%	0 – 2435 (48%) 1 – 2665 (52%)
3		11	2914 – 57.1373%	0 – 2201 (43%)

			1 – 1735 (34%) 2 – 1164 (23%)
4	12	3287 – 64.451%	0 – 1786 (35%) 1 – 1716 (34%) 2 – 924 (18%) 3 – 674 (13%)
5	13	3757 – 73.6667%	0 – 1241 (24%) 1 – 1296 (25%) 2 – 1001 (20%) 3 – 650 (13%) 4 – 912 (18%)
6	11	4018 – 78.7843%	0 – 1056 (21%) 1 – 877 (17%) 2 – 774 (15%) 3 – 643 (13%) 4 – 910 (18%) 5 – 840 (16%)

As it has significantly more accuracy regarding correctly clustered instances, best number of clusters is selected to be 2. It can be seen that when there are more clusters, the accuracy is lower. The instances are generally clustered nearly homogenically.

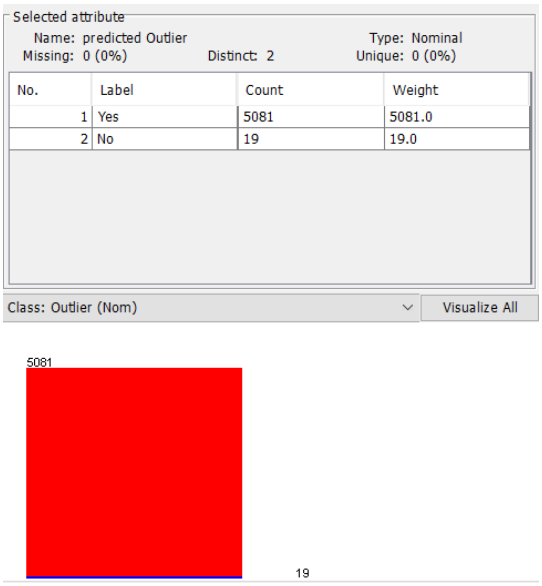
Clustering using farthestFirst

Alternative to kMeans, farthestFirst algorithm was also run of the set. The same k values were applied, and also for that method, it seems that when K value is 2, the accuracy is the best. The shifts between incorrectly clustered instances' percentage for each k value is lower than the shift in the results of kMeans. Also, the instances are not clustered homogenically according to the results.

K value	Stats	Incorrectly Clustered Instances (Num - %)	Clustered Instances
2		1639 – 32.1373%	0 – 3454 (68%) 1 – 1646 (32%)
3		1645 – 32.2549%	0 – 3430 (67%) 1 – 1640 (32%) 2 – 30 (1%)
4		1709 – 33.5095%	0 – 3371 (66%) 1 – 1544 (30%) 2 – 24 (0%) 3 – 161 (3%)
5		1713 – 33.5882%	0 – 3368 (66%) 1 – 1527 (30%) 2 – 19 (0%) 3 – 138 (3%) 4 – 48 (1%)
6		2572 – 50.4314%	0 – 2511 (49%) 1 – 857 (17%) 2 – 19 (0%) 3 – 86 (2%) 4 – 36 (1%) 5 – 1591 (31%)

Outlier Detection using LOF

Firstly, by using Excel's find and replace function, for Label37, all the values with 'o' were replaced with 'Yes', and 'n' are replaced with 'No'. Then, Outlier Detection was performed using LOF algorithm with 10-fold cross validation in Weka. Once the detection is done, the result was saved as another arff file to compare the outliers originally given in the data set with the predicted outlier results of the algorithm.



From the picture it can be seen that 5081 of the values are predicted to be outliers, and 19 of them are not.

Comparison of LOF predicted outlier to Outlier attribute

From the same LOF results Weka file, when the results and the original outlier values were compared by sorting the results by Outlier, it was seen that there were 65 of the instances were presented with 'Yes' in both predicted Outlier and Outlier columns.

: Label36 lumeric	37: prediction margin Numeric	38: predicted Outlier Nominal	39: **Outlier** Nominal
55.0	-0.922433	Yes	No
90.0	-0.974041	Yes	No
87.0	-0.945077	Yes	No
83.0	-0.838196	Yes	No
61.0	-0.953707	Yes	No
80.0	-0.949877	Yes	No
65.0	-0.834887	Yes	No
63.0	-0.877547	Yes	No
94.0	-0.931951	Yes	No
68.0	-0.771011	Yes	No
101.0	-0.972224	Yes	No

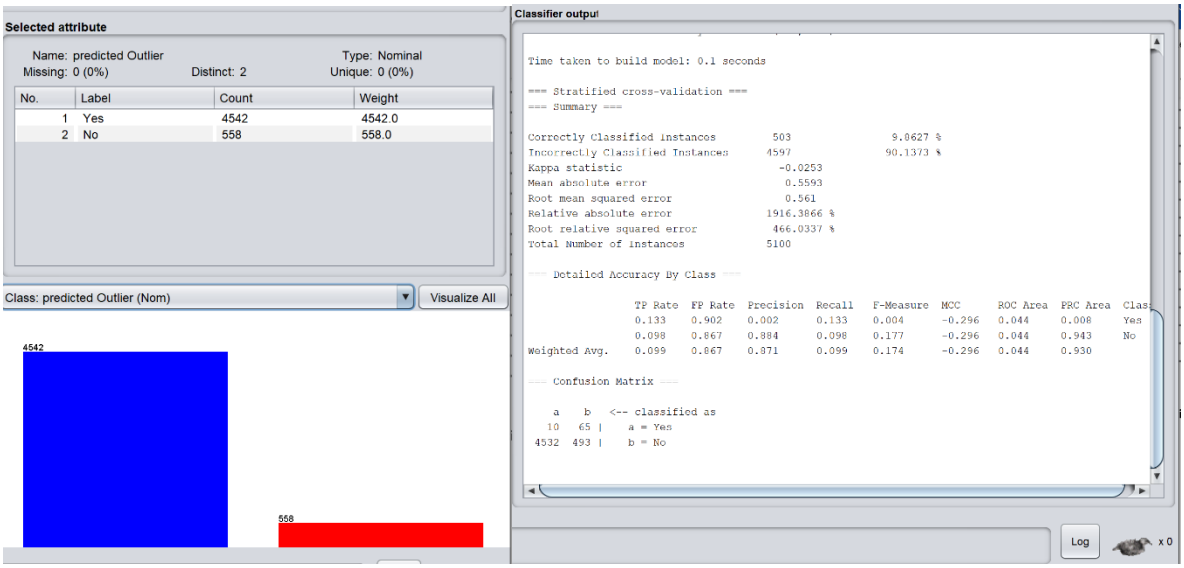
Finding the closest result to the given Outlier

In order to detect the outliers more properly, the number instances that were presented as Yes in both outlier columns should have been as close as possible to the number of original outliers. So, instead of using all the attributes in LOF algorithm, most attributes were randomly eliminated. Only Label1 and Label17 was left for the analysis. Then, the LOF method was performed, and the results were saved as the previous step. Finally, when it was looked at the file, there was seen that there were 74 of the attributes were presented as Yes in both columns which was is close enough to use these results.

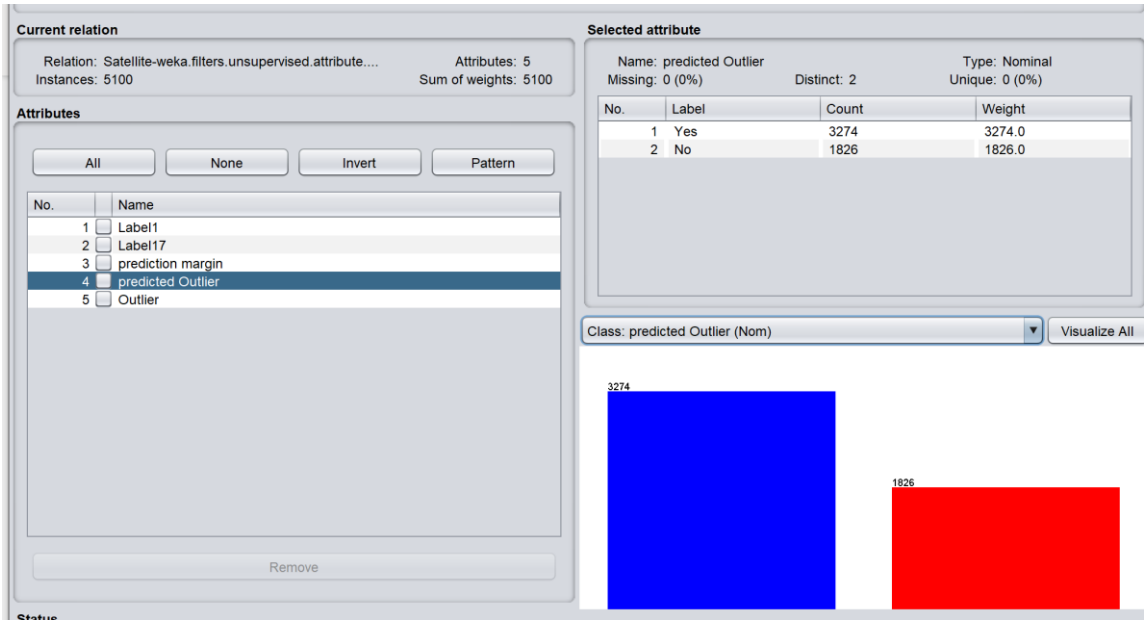
No.	1: Label1 Numeric	2: Label17 Numeric	3: prediction margin Numeric	4: predicted Outlier Nominal	5: **Outlier** Nominal
1	43.0	41.0	0.183597	Yes	Yes
2	56.0	60.0	0.810123	Yes	Yes
3	49.0	47.0	0.784766	Yes	Yes
4	60.0	68.0	0.231434	Yes	Yes
5	56.0	57.0	0.9855	Yes	Yes
6	63.0	71.0	0.977601	Yes	Yes
7	67.0	66.0	0.994861	Yes	Yes
8	44.0	41.0	0.1631	Yes	Yes
9	80.0	71.0	0.901552	Yes	Yes
10	46.0	46.0	0.627177	Yes	Yes
11	52.0	59.0	0.630442	Yes	Yes
12	72.0	53.0	0.424336	Yes	Yes
13	71.0	63.0	0.076934	Yes	Yes
14	44.0	43.0	0.308037	Yes	Yes
15	71.0	56.0	0.633967	Yes	Yes

Outlier Detection using Isolation Forest.

After performing LOF, Isolation Forest with 10-fold cross validation, was then performed using the original datasets. On the image below, we see that we see that opposite to the original outliers presented from the satellite dataset, our predicted outliers came back with 4542 records.



Another test was run for ISF and this time using only the attributes 1 & 17 to match the LOF test. Here Isolation forest considering only attributes 1 and 17 returns with 3274 as predicted outliers.



Comparison of ISF Predicted outliers and Outliers.

The first comparison is the test we did considering all the attributes. “Yes” value was replaced with 1 and “No” with 0. Then adding the columns predcted_Outlier and Outliers we got the IS_Result or the match, which gives us 24 matches for both outliers.

	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN
1	Label31	Label32	Label33	Label34	Label35	Label36	'predicti	'predicted Outlier'	Outlier	ISF_Result
16	90	83	59	57	97	86	0.032921	1	1	2
17	89	80	67	79	93	76	0.163939	1	1	2
24	94	76	57	81	90	76	0.06684	1	1	2
27	78	62	53	49	78	58	0.007084	1	1	2
38	114	97	74	88	110	94	0.040792	1	1	2
39	77	62	67	72	77	58	0.060585	1	1	2
48	89	71	67	75	77	62	0.063264	1	1	2
50	100	81	59	87	96	81	0.084887	1	1	2
54	100	92	56	53	108	107	0.061628	1	1	2
55	112	100	66	83	117	100	0.063113	1	1	2
67	67	50	60	62	67	54	0.010124	1	1	2
69	96	87	59	72	96	83	0.032282	1	1	2

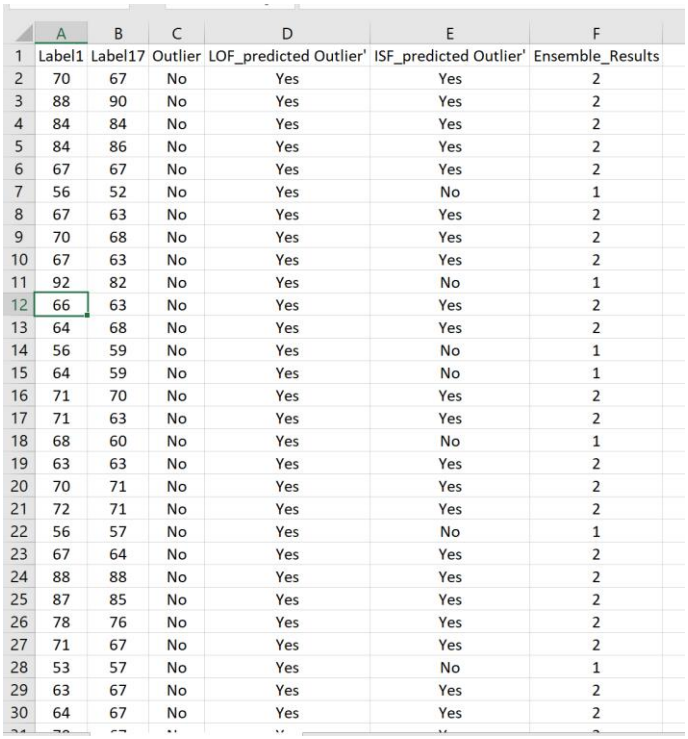
Second comparison is test we ran with only Label 1 and 17 as the attributes. Similar process was done to get the comparison. On this test run, we get 26 matching records.

	A	B	C	D	E	F
1	Label1	Label17	'predicti	'predicted Outlier'	Outlier	ISF_Result
510	63	71	0.031603	1	1	2
511	67	66	0.182985	1	1	2
1020	71	63	0.000581	1	1	2
1526	63	63	0.183175	1	1	2
1527	63	60	0.028784	1	1	2
2038	66	67	0.181154	1	1	2
2548	67	63	0.186459	1	1	2
3055	64	68	0.145298	1	1	2
3057	63	60	0.040274	1	1	2
4075	71	63	0.010155	1	1	2
4588	67	63	0.198567	1	1	2
4589	76	75	0.119892	1	1	2
5097	63	59	0.018859	1	1	2

Combined Result

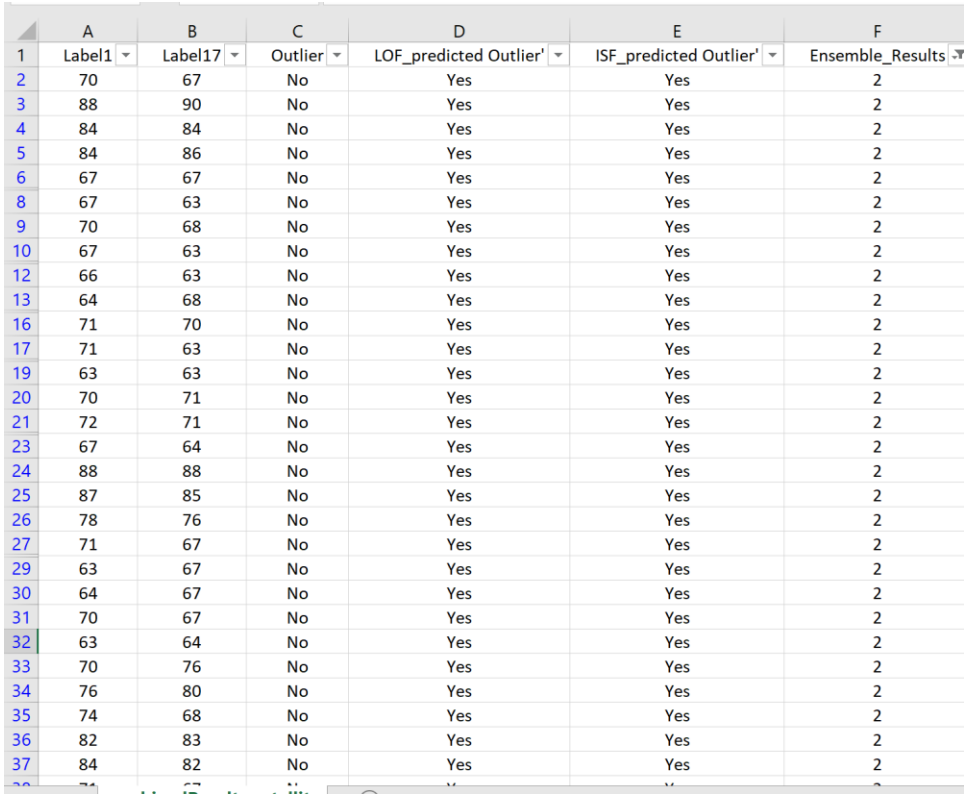
We have created a new excel file to merge the results of both LOF and Isolation forest. We will use both results from the dataset with only Label 1 and 17 as their attributes. Both were set to 10-fold cross validations. Three screen captures are presented.

The first image shows us the combination of both LOF_predicted and ISF_predicted Outlier with the Ensemble_Result that tells us if both method detected similar record as a rare record.



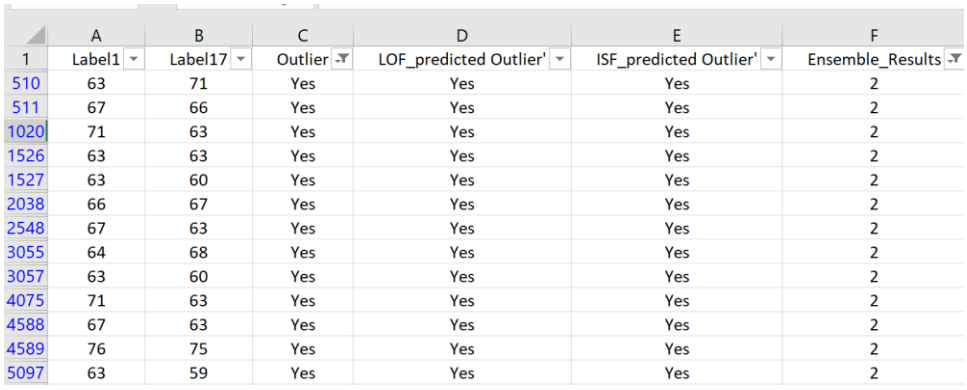
	A	B	C	D	E	F
1	Label1	Label17	Outlier	LOF_predicted Outlier'	ISF_predicted Outlier'	Ensemble_Results
2	70	67	No	Yes	Yes	2
3	88	90	No	Yes	Yes	2
4	84	84	No	Yes	Yes	2
5	84	86	No	Yes	Yes	2
6	67	67	No	Yes	Yes	2
7	56	52	No	Yes	No	1
8	67	63	No	Yes	Yes	2
9	70	68	No	Yes	Yes	2
10	67	63	No	Yes	Yes	2
11	92	82	No	Yes	No	1
12	66	63	No	Yes	Yes	2
13	64	68	No	Yes	Yes	2
14	56	59	No	Yes	No	1
15	64	59	No	Yes	No	1
16	71	70	No	Yes	Yes	2
17	71	63	No	Yes	Yes	2
18	68	60	No	Yes	No	1
19	63	63	No	Yes	Yes	2
20	70	71	No	Yes	Yes	2
21	72	71	No	Yes	Yes	2
22	56	57	No	Yes	No	1
23	67	64	No	Yes	Yes	2
24	88	88	No	Yes	Yes	2
25	87	85	No	Yes	Yes	2
26	78	76	No	Yes	Yes	2
27	71	67	No	Yes	Yes	2
28	53	57	No	Yes	No	1
29	63	67	No	Yes	Yes	2
30	64	67	No	Yes	Yes	2

Second image shows us the records that both LOF and Isolation forest predicted as an outlier. Ensemble results were 3268 counts. If compared to the Outlier detected from the raw data we still got a huge gap.



	A	B	C	D	E	F
1	Label1	Label17	Outlier	LOF_predicted Outlier'	ISF_predicted Outlier'	Ensemble_Results
2	70	67	No	Yes	Yes	2
3	88	90	No	Yes	Yes	2
4	84	84	No	Yes	Yes	2
5	84	86	No	Yes	Yes	2
6	67	67	No	Yes	Yes	2
8	67	63	No	Yes	Yes	2
9	70	68	No	Yes	Yes	2
10	67	63	No	Yes	Yes	2
12	66	63	No	Yes	Yes	2
13	64	68	No	Yes	Yes	2
16	71	70	No	Yes	Yes	2
17	71	63	No	Yes	Yes	2
19	63	63	No	Yes	Yes	2
20	70	71	No	Yes	Yes	2
21	72	71	No	Yes	Yes	2
23	67	64	No	Yes	Yes	2
24	88	88	No	Yes	Yes	2
25	87	85	No	Yes	Yes	2
26	78	76	No	Yes	Yes	2
27	71	67	No	Yes	Yes	2
29	63	67	No	Yes	Yes	2
30	64	67	No	Yes	Yes	2
31	70	67	No	Yes	Yes	2
32	63	64	No	Yes	Yes	2
33	70	76	No	Yes	Yes	2
34	76	80	No	Yes	Yes	2
35	74	68	No	Yes	Yes	2
36	82	83	No	Yes	Yes	2
37	84	82	No	Yes	Yes	2

Here as the final image, we will see how many from the outliers from the raw data that the two-outlier detection process will predict. With the help of the Outlier column form the original file, it helps us to filter the value “Yes”. Now we see that there are 13 records that the ensemble results that are outliers and was previously detected as an odd data.



	A	B	C	D	E	F
1	Label1	Label17	Outlier	LOF_predicted Outlier'	ISF_predicted Outlier'	Ensemble_Results
510	63	71	Yes	Yes	Yes	2
511	67	66	Yes	Yes	Yes	2
1020	71	63	Yes	Yes	Yes	2
1526	63	63	Yes	Yes	Yes	2
1527	63	60	Yes	Yes	Yes	2
2038	66	67	Yes	Yes	Yes	2
2548	67	63	Yes	Yes	Yes	2
3055	64	68	Yes	Yes	Yes	2
3057	63	60	Yes	Yes	Yes	2
4075	71	63	Yes	Yes	Yes	2
4588	67	63	Yes	Yes	Yes	2
4589	76	75	Yes	Yes	Yes	2
5097	63	59	Yes	Yes	Yes	2