Outlier Detection model

Pradeepa Rathi Thiagarasu
www.linkedin.com/in/rathi-thiagu-9925a321

Understanding the problem:

Given an unsupervised data of the molecule behavior, the outliers need to be detected.

1. Why do we need to detect outliers?

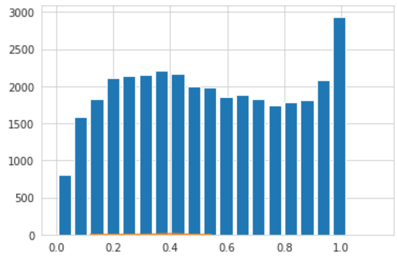
To find either system calibration issues, or drug candidates, or both.

2. Which and how many features can be considered to detect outliers? (**univariate** / **multivariate**)

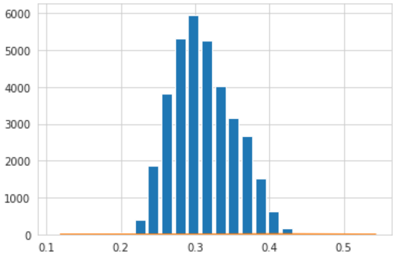
This is a multivariate outlier detection as a single datapoint has 1025 features, so a model needs to be trained to help in visualizing an outlier in the n dimensional space.

3. Assumption of a distribution(s) of values for the selected features? (**parametric** / **non-parametric**)

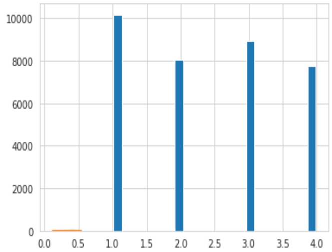
Based on the histogram of features, some of the features have normal /Gaussian distribution. There is no single distribution in this synthetic data.



HTS_1003



HTS_11



SYS_0

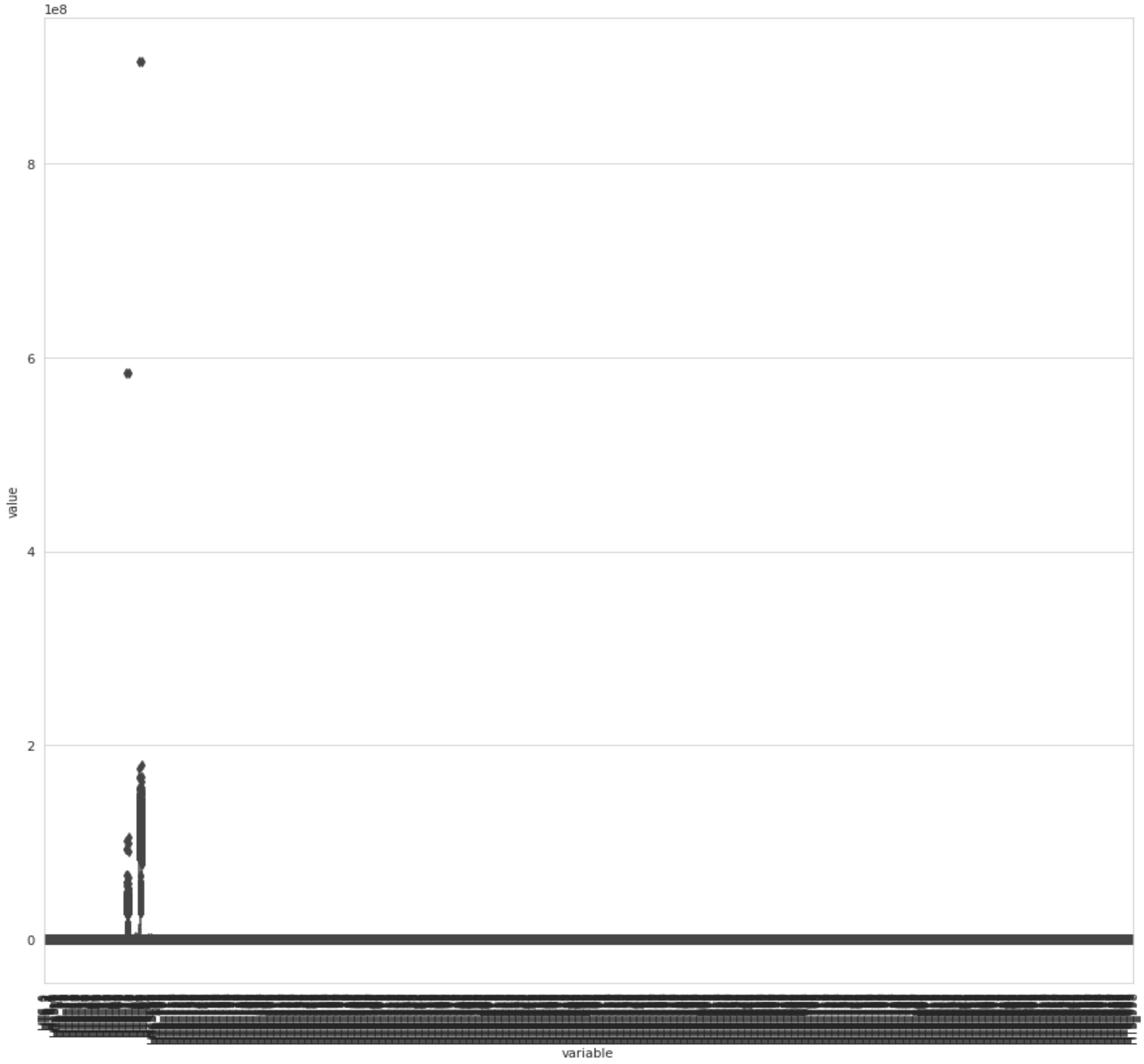
Understanding the data:

```
   No. of rows: 34960
No. of columns: 1025
```

There are 1025 features, and the last three columns represent the system states. First column is an unnamed column, and the second column is a categorical object that has the molecule names which can be dropped for outlier detection.

1025 -2 = 1023 features are used for data preprocessing.

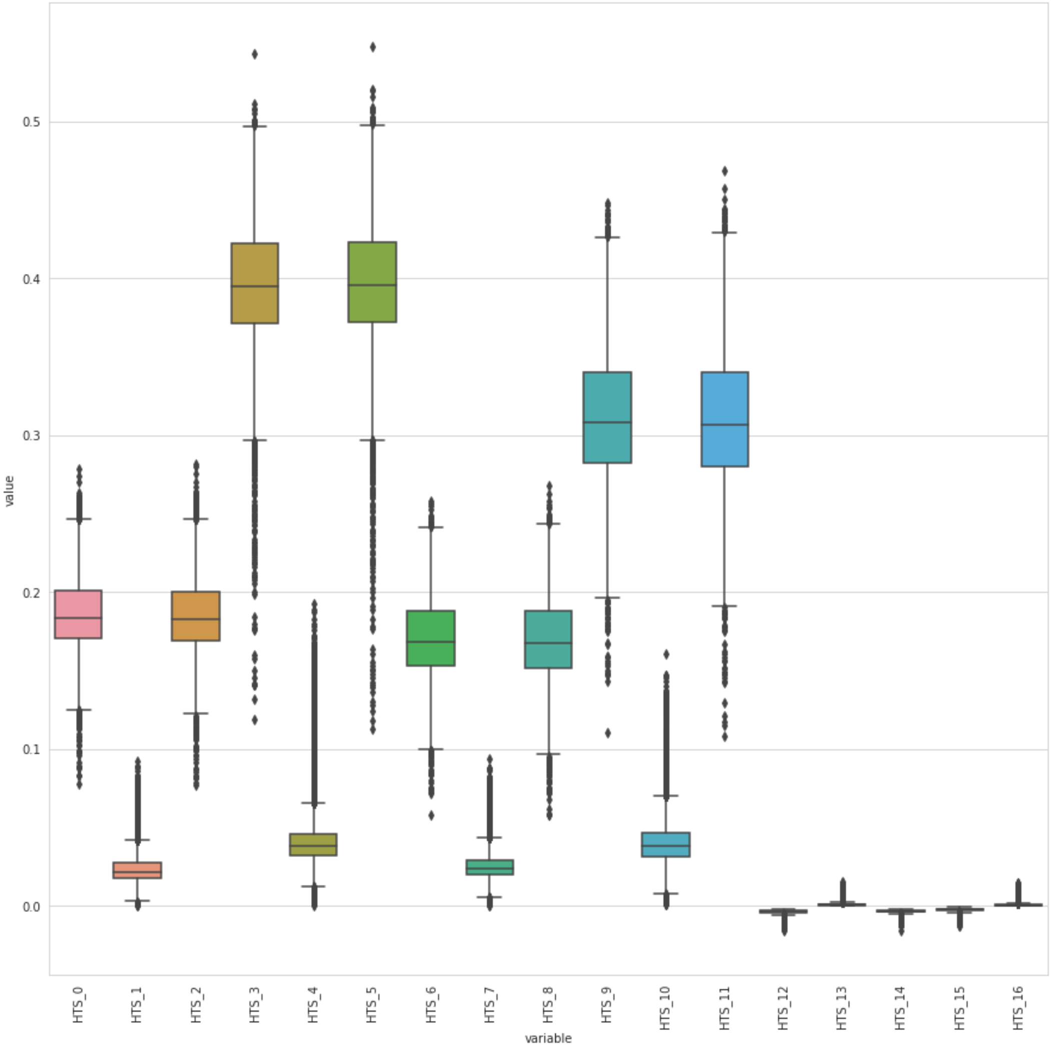
Box Plot of the feature represents there are outliers in the dataset.



This is a box plot for 1000 features. Though it is not readable it gives an idea that two features’ of 1023 features have different value range and have outliers.

Assumptions and Decisions made:

1. **Feature Selection**: Whether feature selection is to be done before outlier detection. Here outlier detection was done and then feature selection was done as I felt in this case all the features are necessary to label a datapoint as outlier.



Box plot for the first 18 features

2. **Feature scaling:** Robust Scaler was used for feature scaling. There was a difference in values after transforming. However, I have used data without scaling for outlier detection.
3. Handling of the NaN values

Preprocessing:

1. Found NaN values in the data

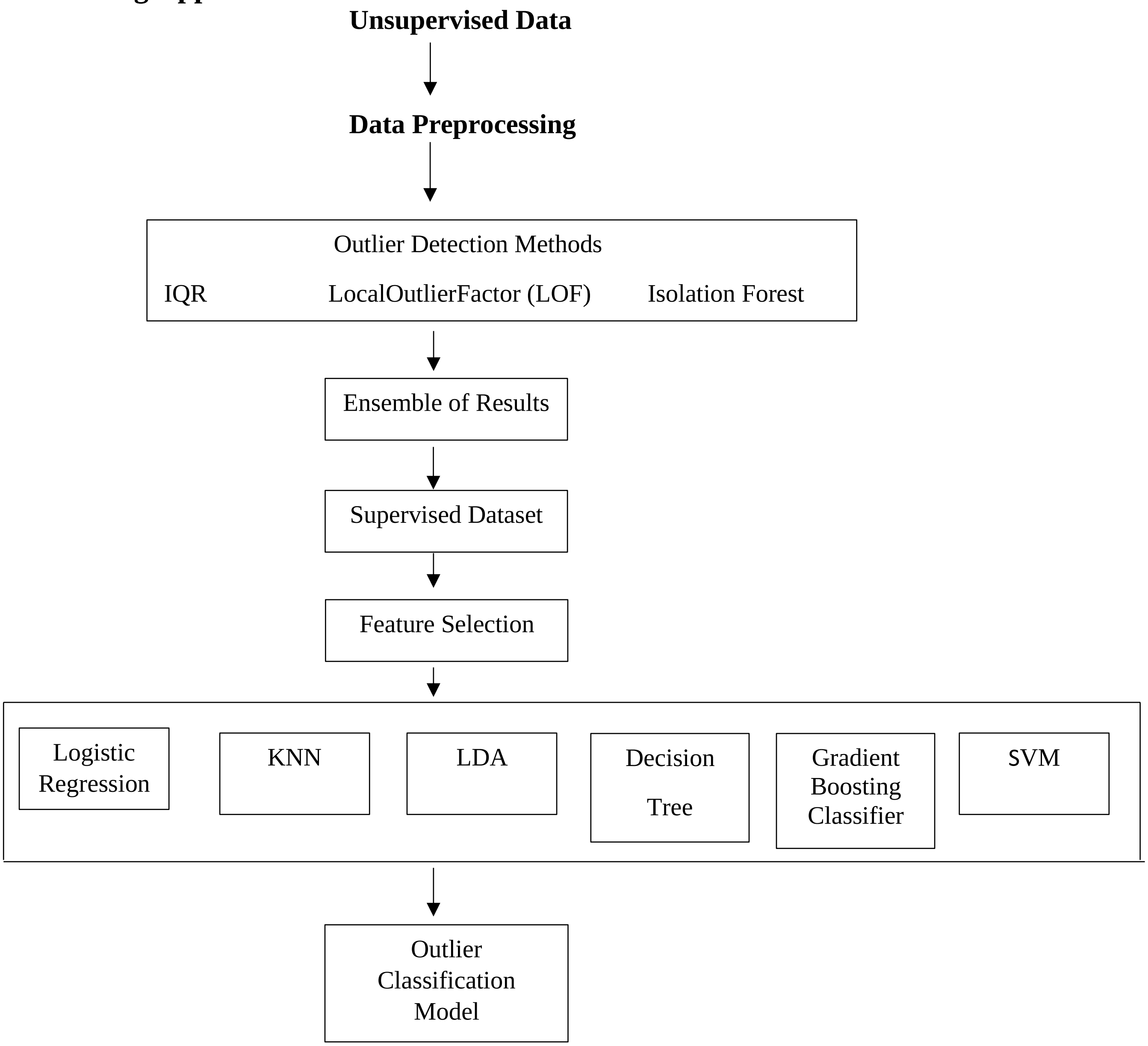
   No. of rows with NaN = 1374

   Rows having more than 20% NaN values were dropped.

   No. of rows dropped = 5

2. Missing values in a feature were filled with mean using SimpleImputer

Modelling Approach:



Above is the approach used to handle the given data and train a model. Initial Preprocessing steps were done, and the result was used for outlier detection. IQR (<3% and >97%), LOR and Isolation Forest were the three methods considered for detecting outliers in the data. Below are the outliers detected by all the three methods.

IQR - 3523
LOR - 609
Isolation Forest - 700

DBSCAN was also tried as it works well with unsupervised data with clustering. But the hyperparameters epsilon and min_samples were hard to decide without domain knowledge.

LOR and Isolation Forest gave outliers in a similar range.

The results of the three models were combined (ensemble) to form a label column where -1 represents outlier and 1 represents inlier.

```
1      31151
-1      3804
```

Now we have a supervised dataset and feature selection was done by evaluating correlation matrix. Features with more than 0.95 correlation value were removed. 1023 features were reduced to 384 features. Dataset shape after feature selection (34955, 384)

After that the dataset was split into train, validation, and test sets.

Train dataset :   **Validation dataset :**   **Test dataset :**

```
 1    19934         1    5000              1    6217
-1     2437        -1     593             -1     774
```

All the datasets were unbalanced. They can be made balanced with down sampling, but it was not done as I thought if the model learns the normal data’s behavior well then it can differentiate the outliers as well. Given time, I would also want to down sample, make the dataset balanced and train the ML algorithms to see if there is a performance improvement.

Logistic Regression, KNN, Linear Discriminant Analysis (LDA), Decision Tree, Gradient Boosting Classifier, SVM were tried.

Error Metric: Accuracy, Misclassification Error, Precision and Recall score

Analysis:

Test performance of the models were compared.

Model	Accuracy	Misclassification Error	Precision (-1)	Recall (-1)
Logistic Regression	0.9	669	0.91	0.15
KNN	0.9	712	0.58	0.28
Linear Discriminant Analysis (LDA)	0.95	381	0.89	0.61
Decision Tree	0.97	**224**	0.85	**0.86**
Gradient Boosting Classifier	0.92	591	0.99	0.24
SVM	Couldn't Run - Model was taking a long time to run			
DNN (Fully connected Neural Networks)	0.88	Model performed poor for the test data		

Out of all the error metrics for this analysis, **Recall score** is the one I believe should be considered for the best performing model. Recall score represents the ability of the model to detect the higher no. of outliers. We need the model to detect the greatest number of outliers. **Decision Tree** ML model gave the best recall score of **0.86**.

I would want to ensemble the best performing classification model, but for most of the models though the accuracy was good, they had very poor recall score, so the ensemble was not done.

DNN (Fully connected neural network) architecture was also trained but they were not able to classify the outliers. This could be because it's more complex for this problem, also there is not a lot of outlier data for training.

If the given is a supervised dataset, outlier detection models like IQR, LOF and Isolation Forest can be used to fit the data and evaluate using the test data with the same error metrics used here. The classification models here are trained based on the results from IQR, LOF and Isolation Forest model predictions which we don’t know if they are accurate. That’s the reason ensemble was used to make the decision based on majority.

References:

1. Ben Auffarth. (2020). Artificial Intelligence with Python Cookbook. Packt Publishing.

2. Geron. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O’Reilly Media, Incorporated.

3. Probabilistic and Statistical Models for Outlier Detection. (2013). In Outlier Analysis (pp. 41–74).

4.https://medium.com/swlh/all-you-need-to-know-about-handling-high-dimensional-data-7197b701244d