Accern Technical Interview

I have attempted this question with three basic strategies.

The first strategy that I have used is to flush out the attributes with a very high proportion of null values and list them as null value anomalies.

The second strategy uses the concept of rolling variance over a 1000 records for the numerical attributes and sees for what time does this window have a higher variance than the average variance of 1000 records windows.

The third strategy is an ensemble of 4 classifiers, inducing instability and randomness in the ensemble with each classifier using a different clustering algorithm. The records which are at a greater distance from their cluster center than a given threshold have been classified as anomalous records. Finally, a weighted majority voting is carried out and record anomalies are listed. (For this part, in spite of trying various schemes like taking a representative sample and trying a rolling window for model training, the program was taking an extremely long time to execute. Thus, in order to display the functionalities of the code in a shorter time of 7 minutes or so, a separate file with only a subset of first 10000 records for ensemble input has been included. Please execute this file to see the effect of the created model).

The following parts contain part-wise description of the Python file and it's description

Part 1 - Null Values Anomalies

- Null values generally tend to add inefficiency and inaccuracy with the data interpretation. Thus, I have assumed that the data attributes with a very high proportion of null values (greater than 0.75) must be eliminated in order to improve further analysis.
- The proportion of null values for each column has been calculated as the ratio of the number of null values in that column to the number of total records in the data.
- The Python script lists the attributes with proportions of null values greater than 0.75 and subsequently deletes the attributes from the pandas dataframe.
- Thus, the Null Value Anomalies are these attributes and they are displayed to the output terminal for further reference to the users. The user can now check these anomalies and continue to ignore or take appropriate action.
- As for this purpose, the null values have been filled with the mode in case of numerical or categorical attributes. However, the id type variables have been marked as "id missing"

Part 2 - Normalizing the numeric attributes and finding their variance

- In the next part, all the attributes which are numeric in nature are considered and they are normalized using the principle (value - min(attribute) / max(attribute) - min(attribute)).
- This would help compare the further computations of variance amongst the attributes and other subsequent calculations in a normalized form. Also, the variance for each numerical attribute is also calculated.

Part 3 - Determining if there is an unusually high amount of variance amongst the numerical attributes in a certain time frame - attribute variance anomalies

- For this functionality, each numerical attribute is considered.
- For each, a rolling window of 1000 records is maintained. The mean of the variance of all rolling windows is calculated.
- If the variance of any window is 50% more than the collective mean of variance of all the windows, than this region is flagged and displayed as an anomaly for that attribute.
- The threshold condition is that the minimum size of the flagged window should be 5000, in order to prevent very small windows being erroneously displayed in the result.
- This analysis would point out the anomalies in attributes that have significantly higher variance as compared to the other timeframes considered for the variables.
- This could signify that the higher variance could be noise and this data may make the analysis for data prone to overfitting this window.

Part 4 - This part assign numerical values to certain string attributes to facilitate application of clustering and the formation of the set of attributes considered for clustering

- There are several categorical string attributes which can take any value from a fixed set.
- These attributes have been identified as the ones which are of string types and can take one value from a fixed set of values. These attributes are mapped to a representative integer where each value in the fixed set is associated with a particular integer.
- The threshold for the number of distinct values for a variable will be less than 1000, i.e. if the attribute takes discrete string values from a set of size less than 1000, it will be included in the clustering model calculation. Such attributes have been mapped to an integer, i.e. each distinct string value has been mapped to a particular integer in order to perform clustering analysis.

- The other type of attributes which are involved in the calculation of the clusters are the numerical attributes.
- All of the attributes used for clustering analysis have been normalized to prevent over-representation or excessive scarcity between data points just due to the magnitude of the attribute.
- Here, the ID type variables are excluded since they are unique for all fields and would lead to overfitting if included in the clustering algorithms. Also, the boolean field of first_mention is treated specially.

Part 5 - Using k-means algorithm to assign clusters to the records and find anomalies

- From the set of attributes obtained from Part 4, each of the attributes' data for the first 10000 records (for demonstration of the ensemble as described above) is extracted and is correspondingly appended into the model input
- A k-means clustering model with predefined number of clusters as 10 is trained on the model data. The cluster label for each record and the center of each of the 10 clusters is obtained and stored.
- For each record, the distance of the point from the allocated cluster's center is measured. If this distance is greater than a defined threshold (defined as 2), the the record is listed as a k-means anomaly.

Part 6 - Using mean shift clustering algorithm to assign clusters to the records and find anomalies

- Similarly as part 5, the first 10000 records (for demonstration) are clustered using mean shift algorithm, which induces randomness in the ensemble since the clustering technique is different and number of clusters is not predefined. (1. Fix a window around each data point. 2. Compute the mean of data within the window. 3. Shift the window to the mean and repeat till convergence.)
- For each record, the distance of the point from the allocated cluster's center is measured. If this distance is greater than a defined threshold (defined as 2), the the record is listed as a mean shift clustering anomaly.

Part 7 - Using mini batch k means clustering algorithm to assign clusters to the records and find anomalies

- Similarly as above,the first 10000 records (for demonstration) are clustered using mini batch k means algorithm, which again works in a different way as it a variant of k means which uses Mini-batches as subsets of the input data, randomly sampled in each training

iteration. In the first step, some samples are drawn randomly from the data set which form a mini-batch. These are then assigned to the nearest centroid. In the second step, the obtained centroids are subsequently updated. In contrast to k-means, this is done on a per-sample basis.

- Again, as above, for each record, the distance of the point from the allocated cluster's center is measured. If this distance is greater than a defined threshold (defined as 2), the the record is listed as a mini batch k means clustering anomaly.

Part 8 - Using Agglomerative clustering algorithm to assign clusters to the records and find anomalies

- This variation of clustering algorithm used is based on the concept of hierarchical clustering using a bottom up approach, i.e. each observation starts in its own cluster, and clusters are successively merged together like forming a tree. The merge strategy is made implementing the default ward metric, which aims minimizing the squared differences within all clusters.
- The centers for each of the allocated clusters is determined by mining the records assigned to each label and taking their mean.
- And as always, for each of the 10000 records, the distance from the allocated cluster's center is calculated, if it is greater than pre defined a threshold (2), the record is listed as agglomerative clustering anomaly.

Part 9 - Weighted Majority Voting to finally determine the records anomalies

- This is the final part, where the output of the ensemble is determined by weighted majority voting.
- I have assumed that the accuracy of mean shift and agglomerative clustering algorithms is higher than the k means and mini batch k means, since the latter two have predefined number of clusters and clustering technique that are prone to overfitting.
- Thus, I have assigned a weight of 2 to mean shift and agglomerative clustering anomalies and a weight of 1 to k means and mini batch k means anomalies. The anomalies' weights are summed up and if the total anomaly weight is greater than 3, than the record is classified as anomaly.

Certain Issues Faced

- The major issue I am facing is the speed of the execution of the Python file, especially the clustering models. This is the reason I have given two files - one which trains the clustering models on the entire data set (which takes a REALLY LONG time to execute) and another one which trains the clustering models on the first 10000 records.

- The other part I had a doubt with is whether to train data on the go in window sets and predict the data for the remaining records as test set as opposed to the approach I have actually adopted.