SENSOR DATA ANALYSIS

NOTE: This document uses the information from the **jupyter notebook** file provided with this folder. You can view either of them but **I would strongly suggest to have a look at it for visualizations.** Both of them use the same text.

About Code: I have created a custom based pipeline to automate the process. Of course several checks can be implemented for handling complex dataset and various file formats. But this is beyond the scope of this particular task. The structure is designed such that the work can be easily extended and upon running the project, an output file can be generated with ranking results.

Before implementing fancy machine learning algorithms right away, it is important to understand the properties of the data.

PROPERTIES

1. **Size:** There are 400 samples with 10 sensor features occupying 34.5 KB of space. The data is extremely small.

2. **Class balance:** The class distribution is **well balanced** with 200 instances of +1 class and other 200 instances of -1 class. This is an ideal scenario. Refer Figure 1 and 2.

	sensor0	sensor1	sensor2	sensor3	sensor4	sensor5	sensor6	sensor7	sensor8	sensor9
class_label										
-1.0	200	200	200	200	200	200	200	200	200	200
1.0	200	200	200	200	200	200	200	200	200	200

Figure 1: Class balance distribution among features.

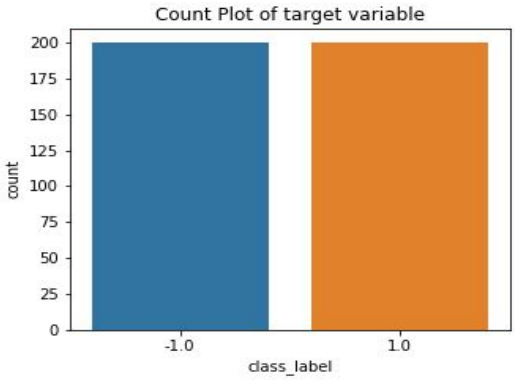


Figure 2: Cunt plot for the classes present in the dataset.

SENSOR DATA ANALYSIS

3. **Data:** The fields **do not have any null values**. There are no categorical or temporal data present. The sensor data points are between 0 and 1 hence there is **no need to normalize** the features.

4. **Correlation:** It describes the statistical relationship between the two variables. **Pearson correlation coefficient** was calculated in this case because it tells if any linear relationship between the two variables exist. The coefficients are in the range of +1 and -1. **Sensor 8 has highest positive correlation** with the target variable followed by **Sensor 4**. **Sensor 1 has the highest negative** correlation with the target variable. **Sensor 6** has almost negligible correlation with the dependent variable. Refer Figure 3.

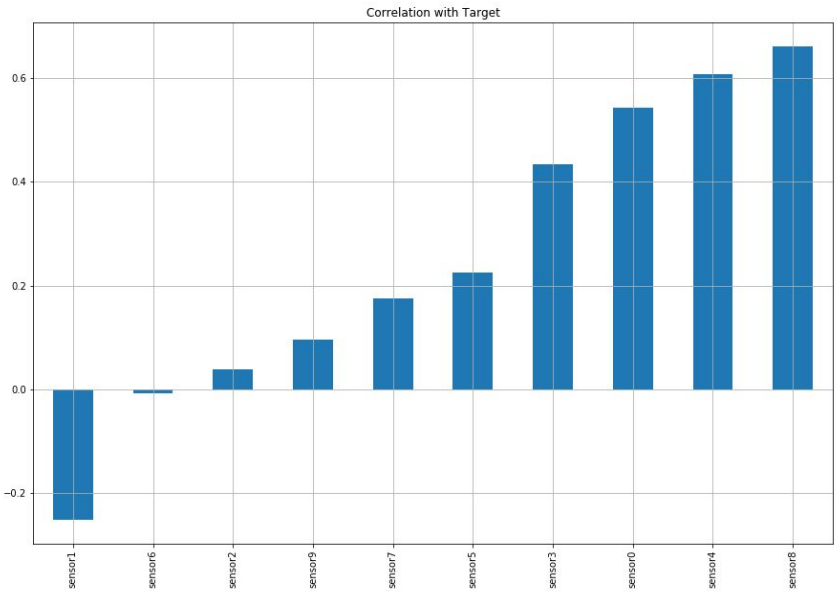


Figure 3: Ranking the sensors from least to highest (left-to-right) in terms of correlation.

5. **Descriptive Statistics:** Mean value of all the sensors is ~0.50 and standard deviation is ~0.28. Refer Figure 4. There is **no need to standardize data** because the data points are on the same scale with close mean and standard deviation. On an average 25% of the data is below 25th quartile, 50% of the data is below 50th quartile and 75% of the data is below 75th quartile. **The data is well distributed.**

SENSOR DATA ANALYSIS

	sensor0	sensor1	sensor2	sensor3	sensor4	sensor5	sensor6	sensor7	sensor8	sensor9
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	0.523661	0.509223	0.481238	0.509752	0.497875	0.501065	0.490480	0.482372	0.482822	0.541933
std	0.268194	0.276878	0.287584	0.297712	0.288208	0.287634	0.289954	0.282714	0.296180	0.272490
min	0.007775	0.003865	0.004473	0.001466	0.000250	0.000425	0.000173	0.003322	0.003165	0.000452
25%	0.299792	0.283004	0.235544	0.262697	0.249369	0.269430	0.226687	0.242848	0.213626	0.321264
50%	0.534906	0.507583	0.460241	0.510066	0.497842	0.497108	0.477341	0.463438	0.462251	0.578389
75%	0.751887	0.727843	0.734937	0.768975	0.743401	0.738854	0.735304	0.732483	0.740542	0.768990
max	0.999476	0.998680	0.992963	0.995119	0.999412	0.997367	0.997141	0.998230	0.996098	0.999465

Figure 4: Descriptive Statistics about the data.

6. **Kurtosis:** All the sensors data distribution have negative kurtosis which implies that all of them have **lighter tails and flatter peak** than the normal distribution. Refer Figure 5. This is also a characteristic of beta distribution with first and second shape parameters equal to 2.

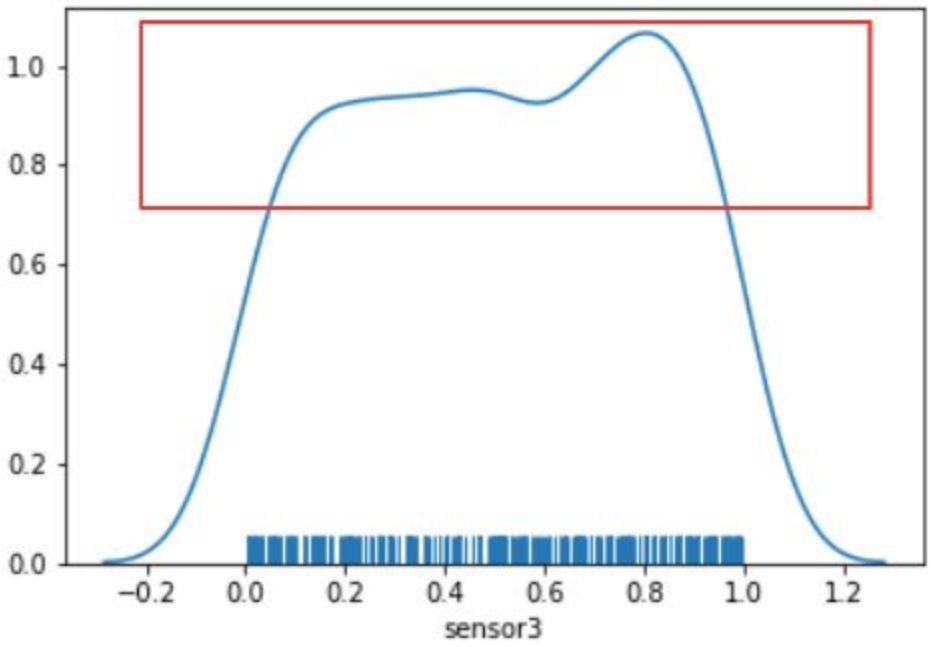


Figure 5: Dist-plot for Sensor 3. Notice the plateau peak and lighter tail. All the other sensors also exhibit similar behaviour.

7. **Skewness:** Sensors-0,1,3,4 and 9 are negatively skewed with a long tail in negative direction. But since the skewed coefficients are nearly close to 0 we can ignore this. Rest of the sensors are positively skewed and again the coefficients are close to 0 and can be ignored.

8. **Outliers:** There are no outliers in the data. Refer Figure 6.

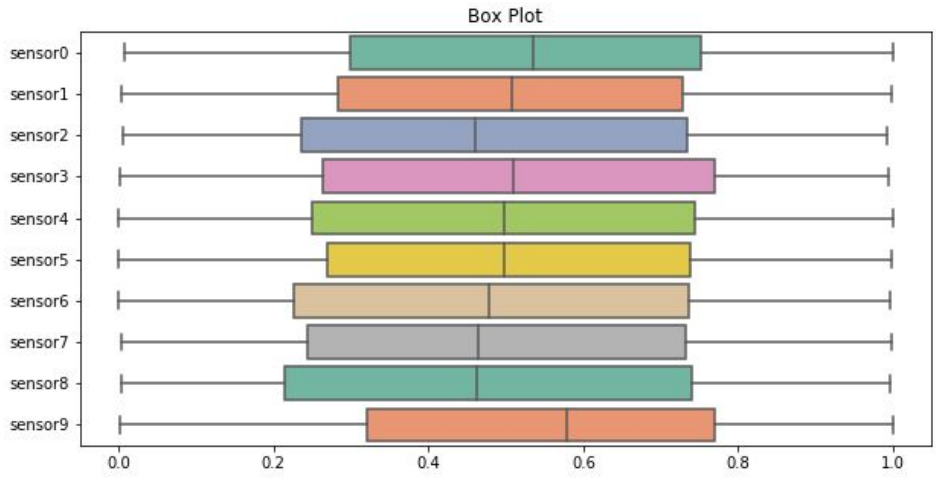


Figure 6: Box plot for all the sensors. No data points are placed beyond the whiskers hence there are no outliers.

9. **Normality test:** None of the sensor data is normally distributed. I have used shapiro-wilk test here since this test is suitable especially for small datasets(usually under thousand sample instances). It is also useful for detecting the potential outliers and we can visualize them using qqplots(not used here).

FEATURE IMPORTANCE/SELECTION METHODS USED

Ranking input features/variables according to their importance is extremely essential in order to avoid overfitting, improve the accuracy with right feature subsets and reduce the number of features in order to train the model faster. There are three ways to do so:

1. **Filter method:** this method does not involve any machine learning algorithm and is treated as a part of the pre-processing step.
2. **Wrapper method:** this method includes selecting subsets of features and then evaluating the algorithm according to their predictive performance.
3. **Embedding method:** this includes the feature selection step during the training time of the algorithm and takes the classifier into account.

Note: I have used one method each from the three categories. I will briefly describe the reasons below. First I will describe the advantages, disadvantages and scalability of the three methods. At the end I will compare the results from the three methods and give insights.

SENSOR DATA ANALYSIS

1. Filter Method: chi2(univariate selection method):

This method is applied to test the independence of two events. In this case it is applied to test the independence of feature variable with the target variable. Results are ranked in descending order according to their importance scores. The most important feature appears at the top.

Advantages:

1. Very fast and computational.
2. Not dependant on any machine learning algorithm.
3. Feature selection/ranking is carried out only once and then only we move to the machine learning algorithm part.
4. Better computational complexity as compared to wrapper methods.

Disadvantages:

1. The correlation between the feature and the target ignores any inter-feature relationship.
2. Lack of feature dependencies can degrade the classifier's performance.
3. Do not take into account the interaction with classifier.

Scalability:

1. It can be easily scaled to a very high-dimensional dataset because there is no dependency on the classifier and the filter method is generally regarded as a part of the pre-processing step.

My conclusion: I would use this method as a part of the pre-processing step and will use the rankings produced by this method only if I see that all the features are independent of each other. Generally the correlation values among features between -0.1 and +0.1 are said to be weak correlations. In this dataset usually all the features have weak positive or negative correlations. It can be safe to assume that all the features are not correlated with each other and we can use the filter method such as chi2test to rank them in accordance to feature importance. But still I won't rely on a single method, especially a method which does not include classifier to generate the importance scores.

2. Wrapper Method: Since this method relies on selecting the variables according to their importance and then building the model and finally picking up the model performing best, I start with **simple logistic regression** for classification. It is a go-to method and it widely used as baseline method.

Next I assess the wrapper method called **Recursive Feature Elimination (RFE)** which is a feature selection approach. It works by recursively removing weaker attributes and fitting a model on specified number of attributes which contribute most to the performance metric.

Observation: Accuracy score when no feature selection/ranking method is used is 92.5% which is great for this dataset. Refer Figure 7. This is as good as using all the features to fit the model. But when I use only the variable sensor8 (highest correlation), the predictive accuracy drops to 85%. For high dimensionality datasets, it is efficient to select relevant subset of features to increase the computational speed and to boost model performance. For this, RFE used with Logistic Regression is great as it greedily selects specified number of best features which give highest performance metric score on the test set.

	Model	Precision	Recall	F1 Score	Accuracy
0	Logistic Regression	0.947368	0.900	0.925	0.925
1	Logistic Regression RFE	0.868421	0.825	0.850	0.850

Figure 7: Results for wrapper method used.

Advantages:

1. This method involves the interaction between feature subset selection and the model selection.
2. This method accounts to the inter-feature-dependencies which is not the case with the chi2 feature selection method.
3. It is quite easy to implement the model.

Disadvantages:

1. Has a high risk of overfitting.
2. It can be computationally intensive for high-dimensional dataset.

Scalability:

This method is good for small datasets with less amount of features but it would take a lot of time to train for high dimensional dataset.

My conclusion: For our sensors dataset, I would prefer using Logistic Regression with RFE since it is easy to implement for such a small dataset(10 attributes). But I would not use this method

on large datasets and would rather prefer embedding method such as random forests, decision trees for scalability and prevention of overfitting.

3. Embedding Method: Extra tree classifier

These methods combine the properties of filter and wrapper methods. It is implemented by algorithms that have their own built-in feature selection methods. It builds multiple trees and splits nodes using random subsets of features. It does not bootstrap observations and nodes are split on random decisions.

Advantages:

1. Random cut-points have excellent variance reduction effect.
2. Reduced computational cost as compared to the random forest in terms of best split.
3. Decreases the variance because trees are constructed in a random fashion.
4. Accounts the classifier during the training.

Disadvantages:

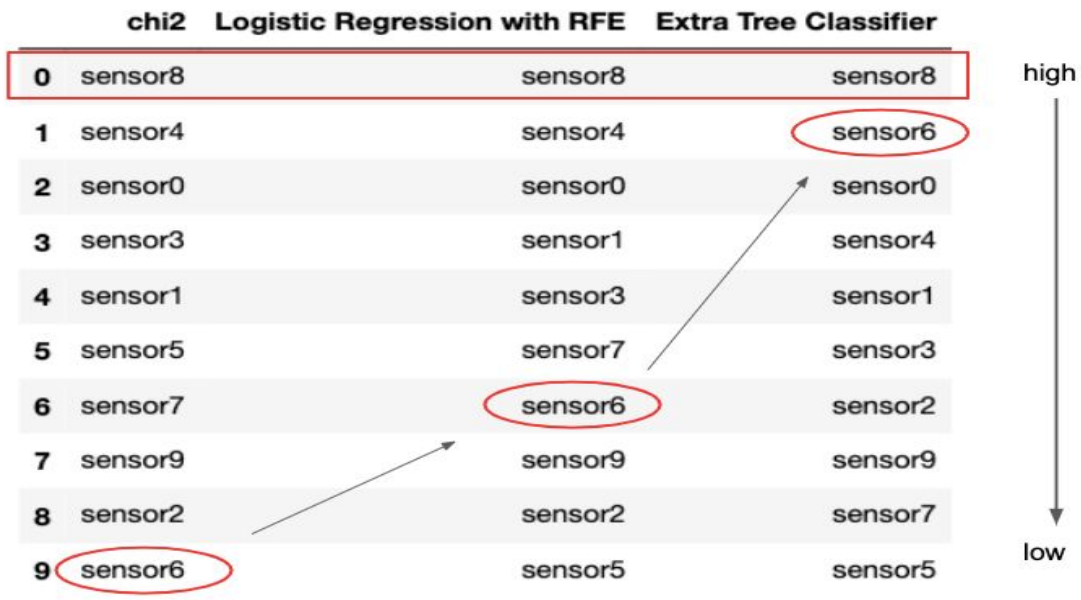
1. When the data is super noisy, creating random splits may not be a good idea.
2. Reduction in variance comes with increase in bias.

Scalability

1. Can be easily scaled to large datasets. Probably, it may take a long time to train. But in my opinion Extra Random Tree Classifier has already been proven to give state-of-the-art performance.

Results

(Please see the figure on the next page.)



	chi2	Logistic Regression with RFE	Extra Tree Classifier
0	sensor8	sensor8	sensor8
1	sensor4	sensor4	sensor6
2	sensor0	sensor0	sensor0
3	sensor3	sensor1	sensor4
4	sensor1	sensor3	sensor1
5	sensor5	sensor7	sensor3
6	sensor7	sensor6	sensor2
7	sensor9	sensor9	sensor9
8	sensor2	sensor2	sensor7
9	sensor6	sensor5	sensor5

Figure 8: Results generated by the three methods used ranking from high to low.

Insights from the three methods:

Refer Figure 8.

1. **Sensor 8** has the highest importance out of all the other features. (highlighted in red rectangle)
2. Correlation of one feature with the target variable does not imply that the same variable will have an important impact in the prediction power of the model. For example, **sensor 6** is shown to have lowest association with the target variable but in extra tree classifier, it holds the second highest importance among all the other features. This shows that the standalone importance of sensor 6 is the lowest but it can hold a high importance in the presence of other variables. This also proves that filter methods alone may give misleading results/degraded performance (highlighted in red circles).

   Similar behavior is exhibited by Sensor 5 but in reverse direction. It is 6th important feature according to chi2 but least important when we account for inter-feature interactions using the second and the third method.

3. **Sensors 8,4,0,1,3** continue to hold the importance in the top 5 ranks in all the three methods.
4. The difference in the results between Logistic Regression with RFE method and Extra Tree Classifier method is because RFE method greedily selects the features and then builds the model iteratively. At the end the best model is picked up which has the highest performance score. But the feature selection is a part of the training process in Extra Tree Classifier and

trees naturally rank the features in terms of node impurity. Greatest decrease in impurity happens at the start of the tree and the lowest happens at the nodes.

Conclusion for this use-case:

1. Filter methods and embedding methods are easily scalable. For a large dataset I would prefer Extra tree classifier or some tree based method such as random forest as a feature ranking algorithm and would use filter method as a part of the pre-processing step.
2. For this dataset I would use Logistic regression with RFE because there are only 10 features. However if the same dataset is used with huge amount of sensor attributes, then I would use tree based algorithm as a method to solve this use-case because of stability, efficiency and its state-of-the-art results proven by researchers.

Alternative Methods

Since there is not a universal perfect solution for this problem and various algorithms can be used to rank the features according to their importance, below I discuss some alternative methods:

1. Filter Method: **variance score, fisher score, information gain can also be used instead of chi2 test.**
2. Wrapper Method: **forward stepwise selection and backward selection method can be used.**
3. Embedding Method: **Random forest, decision trees, L1/L2 regularization method can also be used.**

I would not make the text here long since a lot of discussion can be held for each of the algorithms. I will explain 3 alternative methods from each category.

1. **Variance score:** Variance of each feature is used to rank the variables of a dataset to account their importance. It is scalable and easy to use. But this does not account for the inter feature dependency and also the association with target variable since the variance calculated will be confined to a standalone feature only. For example, if sensor-n has a variance of 6 then it does not tell any association with the target variable. Hence I chose chi2 method.
2. **Forward stepwise selection:** It is comparatively a faster approach as compared to considering all the possible combinations of features to build an efficient model. Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one at a time until all of the predictors are in the model. This method will slow down in a high dimensional dataset and will face scalability issues.
3. **L1/L2 regularization:** This method can be used with any classification technique such as SVM, Logistic regression etc. It solves overfitting problem. It penalizes the objective function during the training time hence the model can take a bit more time to train on large datasets. But this method accounts the classifier for feature importance which is great. It is also scalable.