ANALYSIS

Thought-Process:

The problem statement emphasizes on ranking of the features based on how much information it contributes towards classifying the sample as 1 or -1. This basically implies feature importance for a classification problem. The idea is to try out linear, non-linear and model-agnostic methods and compare their performances on the basis of strengths, weaknesses, accuracy and scalability in terms of feature importance and ranking. So, for linear method Logistic Regression Model is trained, for non-linear method Decision Tree Algorithms are trained and tested and Permutation Importance for a Model-Agnostic method.

Properties of artificially generated dataset:

Exploratory Data Analysis(EDA) is performed on the dataset with 400 rows.

a) It is first checked for outliers using the InterQuartile Range method which is a measure of dispersion similar to standard deviation or variance. There are also other methods like Z-score to check for outliers, but IQ method is much more robust against outliers. We observe that there are **no outliers** in the dataset by comparing the shape of the original dataset and the new dataset without the outliers (if any). The shape is the same!

b) Next step is to check for missing values, we do that with data.info() method. And there are **no missing values** in any of the columns. All are non-null!

c) The last step is to check for the correlation between the independent variables (features) and the dependent variable (target). We use seaborn library for this. We notice that there is **little or no correlation between the features and the target value**. Hence its a cleaned datset. We can go ahead can build our model!

Training of models:

Linear model: Logistic Regression

About: - Logistic Regression is used to find the best hyper plane in k-dimensional space that separates the 2 classes, ie, 1 and -1, hence minimizing logistic loss. Here, the k-dimensional is used to get feature importance. Large positive values contribute towards higher importance in the prediction of positive class (1) whereas high negative values contribute towards higher importance towards prediction of negative class (-1). The Sigmoid Gaussian Distribution (SGD) reduces loss by setting learning with large positive weights for features more important in predicting a data point to belong to the positive class and similarly for negative class, thereby reducing bias.

Method and results: Referring to the code results, we use a Logistic Regression Linear Model with a L2 regularisation (Ridge Regression- squared loss) and a balanced class weight that gives us approximately 90.8% accuracy. Upon ranking the feature importances (obtained with the **coefficient of the features** in the decision function), we get positive and negative importances where the highest positive contributes significantly towards predicting class 1 and the highest negative contributes significantly towards predicting class 0. No clear pattern of important and unimportant features can be identified from the results towards the predictive power as a whole. These coefficients here can be used directly as a crude type of feature importance.

Strength of model:

a) It performs well if data is linearly separable
b) Performs even on multi-class labels (more than 2).
c) It not only provides a measure of how appropriate a predictor (coefficient size) is, but also its direction of association (positive or negative) in terms of feature importance.

Weakness of model:

a) It assumes that the relationship between the dependent and independent variables is linear.
b) Interpretation is more difficult because the interpretation of the weights is multiplicative and not additive.
c) Logistic regression can suffer from complete separation. If there is a feature that would perfectly separate the two classes, the logistic regression model can no longer be trained. This is because the weight for that feature would not converge, because the optimal weight would be infinite.

<u>Non-Linear model: Decision Tree Algorithms</u>

About: - Decision Tree algorithms for a classification task offer important scores for maximizing the **Information Gain (IG)** using the impurity criterion Entropy or Gini. This same approach can be used for ensembles of decision trees, such as the random forest and stochastic gradient boosting algorithms. After being fit, the model provides a **feature_importances_** property that can be accessed to retrieve the **relative importance scores** for each input feature.

The overall importance of a feature in a decision tree can be computed in the following way: It goes through all the splits for which the feature was used and measure how much it has reduced the Gini index compared to the parent node. The sum of all importances is scaled to 100. This means that each importance can be interpreted as share of the overall model importance.

Strengths:

a) The tree structure is ideal for capturing interactions between features in the data.
b) No transformation of features needed.
c) Highly scalable and permits to compute variable explanation very easy.
d) Perform well on large datasets

Weaknesses:

a) Fails to deal with linear relationships.
b) Trees are also quite unstable. A few changes in the training dataset can create a completely different tree. This is because each split depends on the parent split. And if a different feature is selected as the first split feature, the entire tree structure changes. It does not create confidence in the model if the structure changes so easily.
c) Decision trees are prone to overfitting, especially when a tree is particularly deep.

Decision Tree Ensemble methods: Random Forests

About: - Random forests comprise multiple decision trees (Ensemble method), thus the feature importance of feature j is the normalized sum of I.G. brought about by feature j across all trees. Scikit-learn's random forest model has a feature_importance_ attribute that gives the value of **Gini impurity reduction caused by each feature across all levels normalized across trees**.

Strengths:

a) Learns most of the non-linear relationships between X and Y by themselves.
b) Highly scalable
c) Much more robust than a decision tree. They aggregate many decision trees to limit overfitting as well as error due to bias and therefore yield useful results.
d) Also perform well on large datasets. However, for very large data sets, the size of the trees can take up a lot of memory.

Weaknesses:

a) Model interpretability: Random forest models are not all that interpretable; they are like black boxes.

Model-Agnostic Methods: Permutation Feature Importance

About: - Permutation feature importance measures the increase in the prediction error of the model after we permuted the feature's values, which breaks the relationship between the feature and the true outcome. So, a feature is considered to be important if by shuffling its values, the model error increased because in this case the model relied on that feature for prediction.

Method: - Firstly a model is fit on the dataset. The model chosen here must be one that does not support native feature importance scores. The model is then used to make predictions on a dataset, although the values of a feature is shuffled. This is repeated for each feature in the dataset. Then this whole process is repeated 3, 5, 10 or more times. The result is a mean importance score for each input feature (and distribution of scores given the repeats). Here we use KNeighbours Classifier for training.

Strengths:

a) Retraining of model not required unlike some other models which require dropping a feature, retraining the model and comparing the model error.
b) Takes into account all interactions of other features.

Weaknesses:

a) It is unclear whether we have to use training data or test data for eature importance computation.
b) Results might vary greatly as it depends on shuffling of the feature, which adds randomness to the measurement.
c) Permutation feature importance is linked to the error of the model. This is not inherently bad, but in some cases its not exactly what you need. In some cases, you might prefer to know how much the model's output varies for a feature without considering what it means for performance. For example, you want to find out how robust your model's output is when someone manipulates the features. In this case, you would not be interested in how much the model performance decreases when a feature is permuted, but how much of the model's output variance is explained by each feature.
d) Adding a correlated feature can decrease the importance of the associated feature by splitting the importance between both features.
e) If features are correlated, the permutation feature importance can be biased by unrealistic data instances.

Boruta Feature Selection, Recursive Feature Elimination, SelectKBest Features methods can also be used but to select the top 10 features but they don’t really contribute towards the ranking of the features as is required in the problem statement.

Even a simple Neural Network model can be used for ranking but Neural Network is often seen as a black box, from which it is very difficult to extract useful information for another purpose like feature explanations.

Inferences:

a) Since the importance is model-based, these values vary across different algorithms. Weight values determined by SGD in Logistic Regression need not be the same as normalized Gini Impurity in case of Random Forest. In fact, it is not necessary that the relative values follow the same ordering across different algorithms. What feature may be more important for classification using LR need not be as important for RF and vice versa.
b) If XGboost or RandomForest gives more than 90% accuracy on the dataset, we can directly use their inbuilt feature_importance_ method.
c) If the dataset is not too large, use Boruta for feature selection.

Conclusion:

Considering the strengths, weaknesses of all the applied methods and comparing the accuracy, Random Forest achieved an accuracy of 98.34% and it would be reliable to go ahead with this method to ranking the sensor features. With this, sensor 8 is shown to be of utmost importance towards feature ranking.

Citations:

1] Saarela, M., Jauhiainen, S. Comparison of feature importance measures as explanations for classification models. *SN Appl. Sci.* **3,** 272 (2021). https://doi.org/10.1007/s42452-021-04148-9

[2] https://towardsdatascience.com/feature-importance-with-neural-network-346eb6205743

[3] https://christophm.github.io/interpretable-ml-book/tree.html