COMP4434 Big Data Analytics

Mini project
QSAR Oral Toxicity Classification

GROUP 35

CHIU WING KI (21037233D)

LAM WAI TSUN (21019252D)

LEUNG CHUN KIT (21018713D)

LEUNG HEI YIU (21034263D)

1. Objective

The oral toxicity data set from the UCI machine learning repository are used in this project to distinguish and predict the presence or absence of severe toxicity in oral medicines by using different machine learning methods. The main purpose of this project is to create a classifier that can identify oral medicines with a higher potential for toxicity, which can then be subjected to further testing. By using such a classifier, the testing costs can reduce, as only those medicines with a higher likelihood of toxicity will be selected for further testing. However, it is also important to avoid any negative side effects that could arise from taking highly toxic medicines due to incorrect identification. Several machine learning models are built in order to compare the performance and find the best model as the classifier. At the same time, in order to address the imbalanced data, resampling techniques will be used to balance the number of positive and negative data. Also, it is an attempt to reduce the dimensionality of the data by using dimensionality reduction techniques such as K-selection from sklearn, LDA, ICA, and PCA. Finally, the recognition performance of different models is mainly compared through F2 score, so as to select the most suitable model.

2. Analysis method

2.1. Description of the Dataset

The data set contains 8992 chemicals, and 1024 binary attributes, which represent the corresponding number of molecular fingerprints of chemicals. The attributes of molecular fingerprints will be used to classify chemicals into 2 categories, that is very toxic (positive) and not very toxic (negative), both are present in the 1025th attribute as response variable.

2.2. Processing of data

The values of all attributes in the dataset are binary, so after importing the dataset, the values of the attributes used to represent the classification results need to use text instead of numbers 1 or 0. That is, convert 1 to the string "positive" and 0 to the string "negative".

The next step is to drop all the duplicate data that exists in the data set. This step is very important since if there is any duplicate data in the data set, not only will it take longer to run the programming, but it also increases the chance of errors, resulting in inaccurate final classification results.

After the cleansing part, the whole dataset needs to split into a training dataset and testing dataset for estimate the performance of different machine learning algorithms when they are used to make predictions. Set the first 1024 attributes of molecular fingerprints as predictor or independent variables (X), and the last attribute as response or dependent variable (Y), then split the cleaned datasets into train dataset and test dataset by using train_test_split method from library sklearn.model_selection in Python. To ensure execution of the same train and test sets, the random state is set to be 1 and testing size is 0.2. The training and test sets were then split eight to two, with 6,812 chemical samples in the training set and 1,703 in the testing set. Through calculation, it is found that the percentages of oral medicines containing very toxicity are 8.54% and 8.04% in the training set and test set, respectively, and there is little difference between the two.

2.3. Evaluation Criteria

In this report, there are 4 performance indexes used to evaluate the classification results, including:

F2 score

Since the cost of misclassifying toxic medicines to non-toxic medicines has a higher cost. We willing to increase the recall rate for certain levels and sacrifice some precision rate. F2-score is used as the scoring function for grid search with cross-validation. This scoring function measures precision and recall, with recall being 5 times more important than precision. The F2 score is the weighted harmonic mean of the precision and recall of the model, and unlike the F1 score, this score gives more weight to the recall than to the precision. The larger the value of F2 score, the better the model's correct prediction of positive samples. The F2 score is ideal for use in situations where it is desired that the prediction captures all oral drugs that contain toxicity, and the formula of the score is as follow:

F2 (β = sqrt(1/5)) = $(1+\frac{1}{5})\frac{prescision\times recall}{\left(\frac{1}{5}\times prescision\right)+recall}$

Recall score

The recall score in the f2 score is mainly focused on this report because this score is used for positive observations correctly identified from all actual positive cases (both true positives and false negatives) of the model. As oral medicines have a high risk of being very toxic, the consequences of classifying oral medicines that are not very toxic as very toxic are far less severe than classifying medicines that are very toxic as not being very toxic. The same with F2 score, the larger the value of recall score, the better the model can make correct predictions in positive sample.

Precision score

The precision score is the positive observations that the model correctly identified from all the observations it marked as positive, including true positives and false positives. It is no more important than the recall score in identifying all oral drugs with toxicity, so it is not the main criterion for comparing model performance.

Accuracy

Accuracy, which is one metric for evaluating the performance of a classification models, is used to calculate the percentage of correct predictions in the testing data. The higher accuracy the better the performance of a model.

2.4. Classification Model

A total of 11 models are used for classifying the toxicity in oral medicine, including Decision

Tree Classifier, K-Nearest Neighbors Classifier, Support Vector Classification, Gradient Boosting Classifier, Gaussian Naive Bayes, Logistic Regression, Linear Discriminant Analysis, AdaBoost classifier, XGBoost Classifier, Linear Regression, and multi-layer backpropagation.

2.5. Model Building

Once the data preprocessing is complete, the models are built using the original imbalanced dataset. Subsequently, resampling techniques are applied to the dataset, and the models are trained again. Then, compare the performance of the models with and without resampling. Also, we are keen to examine the effect of dimensionality reduction. Hence, several dimensionality reduction methods are applied and compared to determine their impact on the performance of the models.

3. Introduction of Classification Model

3.1. Decision Tree Classifier

A decision tree classifier is a decision tree used for classification, which is based on the attribute values in the dataset to make class label or discrete predictions. It may have many branches.

3.2. K-Neighbors Classifier (KNN)

The k-Nearest Neighbors algorithm, also known as KNN, is a nonparametric, supervised learning classifier that uses proximity to classify or predict groups of individual data points. It is usually used for classification because it can judge the category of new data by finding the category of K neighbors (data) near the new data.

After import KNeighborsClassifier method from "sklearn.neighbors" library, and finding the best neighbor parameters, which is one in this project, fitting the training data into the model and then predicting the target values for the test data can be performed.

3.3. Support Vector Classifier (SVC)

Support vector classifiers (SVC), which is a supervised learning method, are machine learning algorithms that analyze data for classification and regression analysis. SVM is a supervised learning method that looks at data and classifies it into one of two categories. It has relative advantages for data sets containing small samples, nonlinearity, high dimensionality, and local minimum points, so the QSAR model-based oral toxicity dataset may be suitable for using this classifier.

After importing the SVC method from "sklearn.svm" library, the kernel in SVC model is needed to be found. By collaborating with the kernel function, SVC can achieve better performance in classification problems, so the kernel trick is a very important part of SVC learning. The role of the Kernel trick in machine learning is that when different types of data cannot be separated by a linear classifier in the original space, the data after nonlinear projection can be more separated in a higher-dimensional space. Therefore, this is more suitable for the high-dimensional data set in this project.

3.4. Gradient Boosting Classifier (GB)

A gradient boosting classifier is a classifier that uses gradient boosting, a machine learning technique for regression and classification problems. Its purpose is to combine relatively weak predictive models to build a stronger predictive model. The classifier builds the model in stages, that is, combining weak learners into a strong learner in an iterative manner. As each weak learner is added, a new model is fitted to provide more accurate estimates of the response variable.

Learning rate is one of the key hyperparameters of gradient boosting classifiers, which affects the performance of the model. Each learner added to the model will modify the entire model, and the magnitude of the modification is controlled by the learning rate. The lower the learning rate, the slower the model learns, and vice versa. In the QSAR based oral toxicity dataset, it was found that 0.8 is the best learning rate when using the GB model.
By this method it shows that for the GB model the accuracy is about 91.7205%

3.5. Guassian Naive Bayes (NB)

Gaussian Naive Bayes (GNB) is a classification technique based on probabilistic methods and the Gaussian distribution, also known as the normal distribution. GNB assumes that each predictor variable has an independent ability to predict the output variable, and the combination of predictions of all predictor variables is the final prediction, which returns the probability that the dependent variable is classified into each group. Then the final classification result is assigned to the group with higher probability. Before prediction and classification, GaussianNB method also needed to be imported from "sklearn.naive_bayes"library.

3.6. Logistic Regression (LR)

When there are multiple variables in the logistic regression model, imposing penalties on the model can shrink the coefficients of variables that contribute less to zero, thereby reducing overfitting. The most commonly used penalty regressions include Ridge regression. It is in the case that all variables are included in the model, the coefficient of the variable that will contribute less will be close to zero.
Lasso regression. It is to force the coefficients of some variables with small contribution to zero so that only the most important variables are present in the final model.

3.7. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis is a robust classification. It is a model that is normally used for doing classification, dimension reduction and data visualization. The LDA method normally can give out the robust, decent and interpretable classification result. The Linear Discriminant Analysis can compute the algorithm easily. It's effectiveness in improved with the amount of the data which can also find the correlation between data.

3.8. AdaBoost classifier

AdaBoost which is also called Adaptive Boosting. This estimator normally used with decision tree model, but it normally only contain one spilt. Also the algorithm is used to give equal weight to all the data. Then the higher weight data which means it was wrongly classified. After that the model will be keep on training until the lowest error model received.

3.9. XGB Classifier (XGB)

The XGB classifier is also called the Extreme Gradient Boosting. The algorithm is used train the model and find the patten of the data then label the feature of the model. So to use the model to predict the data on the train data set. Compare to other models the result produce of the XGB classifier is fast.

3.10. Linear Regression (LR)

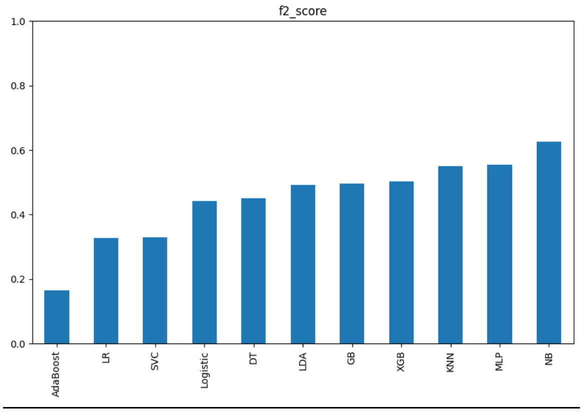
Linear Regression is the basic kind od the prediction analysis. So that to examine how the outcome of the predictor. To find the outcome of the variable. Then to estimate the data by using the formular $y = mx+y$, the c is the constant, m is the regression coefficient, y is the estimated dependent variable score, and x is the score on the independent variable. This can analysis the strength of the predictor, effectiveness of forecast and the trend of forecasting.

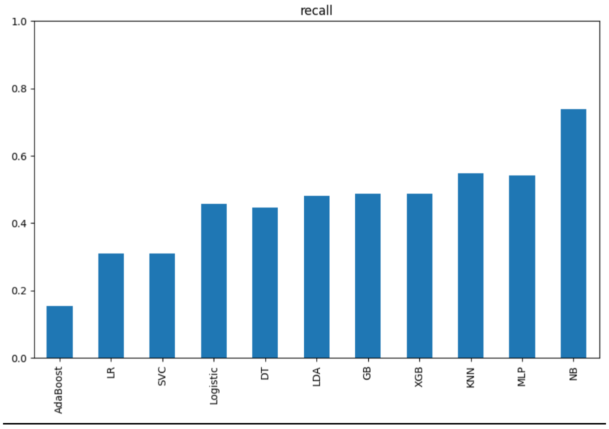
3.11. Multi-layer backpropagation (MLP)

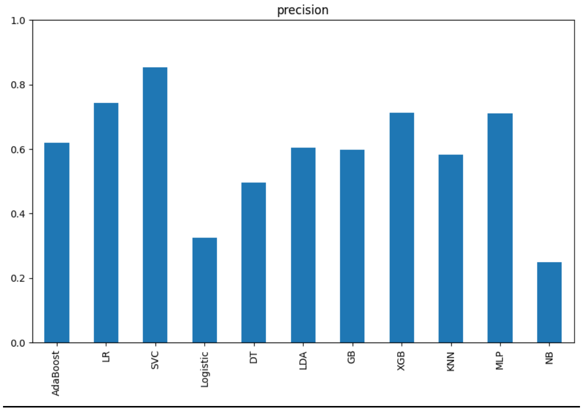
Multi-layer perceptron (MLP) is a kind of artificial neural network. The information flows in one direction, from input to output. There are no feedback loops. The input layer receives the input data, and the output layer produces the network's output. The hidden layers perform computations and transform the input into a form that is more easily processed by the output layer.

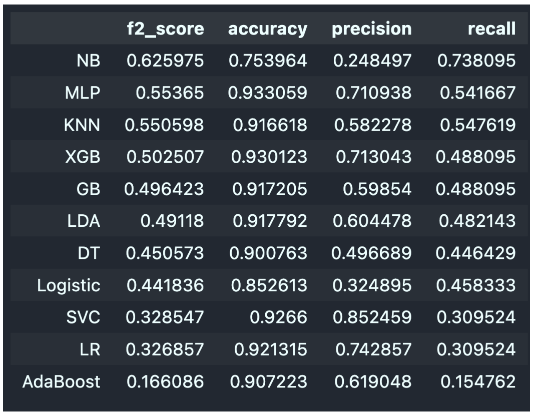
The neurons in each layer are connected to the neurons in the adjacent layer through weighted connections. The weights are adjusted during the training process using backpropagation algorithm. It adjust the weight of the model form the error.

Result with original dataset









	f2_score	accuracy	precision	recall
NB	0.625975	0.753964	0.248497	0.738095
MLP	0.55365	0.933059	0.710938	0.541667
KNN	0.550598	0.916618	0.582278	0.547619
XGB	0.502507	0.930123	0.713043	0.488095
GB	0.496423	0.917205	0.59854	0.488095
LDA	0.49118	0.917792	0.604478	0.482143
DT	0.450573	0.900763	0.496689	0.446429
Logistic	0.441836	0.852613	0.324895	0.458333
SVC	0.328547	0.9266	0.852459	0.309524
LR	0.326857	0.921315	0.742857	0.309524
AdaBoost	0.166086	0.907223	0.619048	0.154762

Based on the results obtained, the Naive Bayes model performed the best, with the highest F2-score and recall. It was able to correctly identify 73% of the toxic medicines in the test data. Also, we observed that recall and precision are inversely proportional to each other in general.

4. Resampling

As can be seen in the above section, the proportion of highly toxic oral drugs in the training set and test set is only about 8%, which shows the problem of data imbalance in the data set. To solve this problem, resampling method can be used to improve the identification accuracy of the population and estimate any uncertainty in the population. In this part, undersampling and oversampling techniques are used to resample and balance the number of positive and negative data, so that the ratio of the oral medicines that consist very toxic to those that do not consist of very toxic oral medicines can reach one-to-one.

4.1. Undersampling

One way to randomly resample an imbalanced dataset is to remove examples from the majority class, known as undersampling. It removes examples from the majority class, and this process can be repeated until the desired class distribution is achieved, such as an equal number of positive and negative samples in the Oral Toxicity data set. The random undersampling technique can be implemented by using the RandomUnderSampler imbalanced-learn class after splitting the data into training and test sets, and setting the sampling_strategy parameter to 1, which means "majority", to determine the class with the most samples as majority class.

4.2. Oversampling

Another way to randomly resample an imbalanced dataset is to replicate examples from the minority class, known as oversampling. It involves randomly copying examples from the minority class (i.e., no very toxic oral drugs) and adding them to the training dataset, repeating this step until the two datasets in the training set become balanced. Like undersampling, oversampling can be achieved by using the RandomOverSampler class and setting the sampling_strategy parameter to auto, which means "minority", to automatically balance the minority class with the majority class.

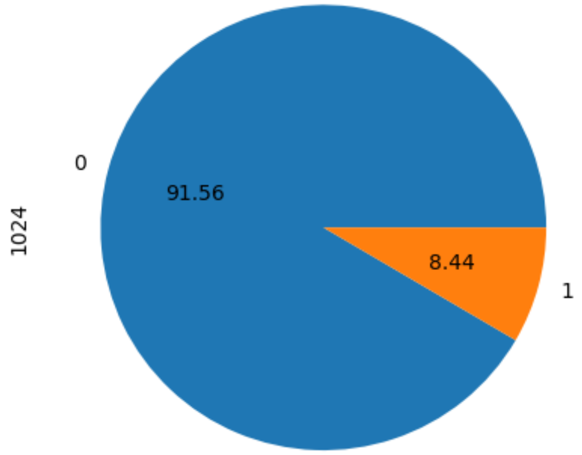
4.3 Result and Finding

The following table is the F2 score of different models analyzing oral toxicity under different sampling conditions.

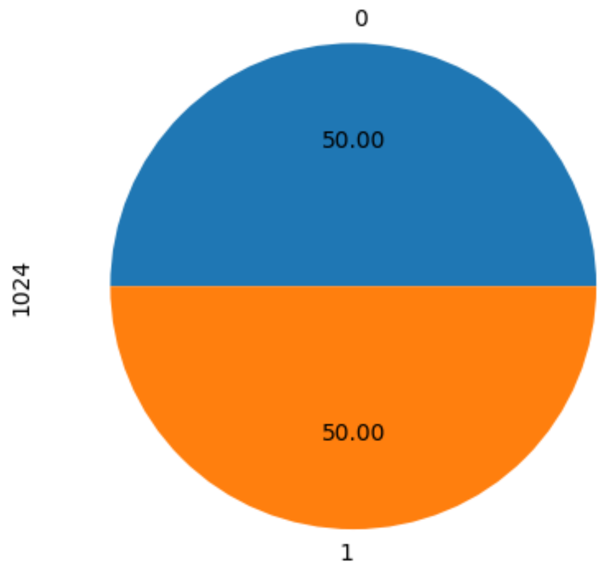
Type of the Classifier	**Original dataset**	**Undersampling**	**Oversampling**
Decision Tree Classifier	0.4877	0.6293	0.5543
K-Neighbors Classifier (KNN)	0.5506	0.7196	0.6584
Support Vector Classifier (SVC)	0.3284	0.7301	0.7176
Gradient Boosting Classifier (GB)	0.4961	0.6982	0.6166
Gaussian NB (NB)	0.6260	0.6421	0.6297
Logistic Regression	0.4418	0.705	0.4516
Linear Discriminant Analysis (LDA)	0.4912	0.4611	0.6763
AdaBoost Classifier	0.1661	0.607	0.6404
XGB Classifier (XGB)	0.5025	0.7422	0.6004
Linear Regression	0.3269	0.4611	0.6763
Multi-layer perceptron (MLP)	0.5537	0.6894	0.5028

It is found that undersampling performs better with shorter training time, while oversampling can also improve performance but is not as effective as undersampling and takes longer training time. After resampling, both undersampling and oversampling have 50% positive and negative records, but the training dataset size is 1102 for undersampling and 12522 for oversampling.

Distribution of the training data in original dataset



Distribution of the training data in undersampling dataset and oversampling dataset



Gaussian NB is the best model when the original data set is used, with f2 score 0.63. After preforming undersampling, XGB Classifier is the best model with the f2 score 0.74. And the best model in oversampling is SVC with f2 score 0.72. The F2 scores of different models are basically improved, which means that resampling can effectively improve the recognition performance of the model.

It is recommended to choose to use undersampling because of the great performance and the lower training cost compare with oversampling.

5. Dimensionality reduction

Dimensionality reduction is mainly used in data sets with large dimensions, which can help reduce training time and computing resources, while improving the overall performance of machine learning algorithms. In this part, four dimensionality reduction techniques including K-selection from sklearn, LDA, ICA and PCA will be used to reduce the dimensionality of the undersampling dataset.

5.1. Feature Selection by using SelectKBest

SelectKBest is used to remove all but the k highest-scoring attributes. By using chi square to calculate the molecular fingerprints of all chemicals, when the value of chi square is larger, the weight of this molecular fingerprint is greater. Then set the value of k to 300, which means that the 300 molecular fingerprints with the highest chi square will be retained, and the rest will be deleted. This method can filter out the attributes that are less important to the recognition of oral drugs, so as to increase the accuracy of recognition while reducing the training cost.

5.2. Linear Discriminant Analysis (LDA)

While LDA can be used for multi-class classification, it can also be used as a dimensionality reduction technique. LDA, which seeks to find the linear combination of input variables that achieves the maximum separation of samples between classes and the minimum separation of samples within each class, can be achieved by using Scikit-learn's LinearDiscriminantAnalysis function. After fitting the model using fit(X, Y), the transform(X) method is used to reduce the dimensionality of the dataset.

5.3. Independent Component Analysis (ICA)

ICA is a linear dimensionality reduction method used to transform a dataset into columns of independent components. Dimensionality reduction by the fast ICA method reduces the time cost of training overall without significantly affecting accuracy, and this method can be achieved by using Scikit-learn's FastICA function. After splitting the undersampled data set into training data and test data, use FastICA() to fit the model and set the number of components to 300, which represents the number of molecular fingerprints that need to be used. Then, the transform(X) method is used to reduce the dimensionality of the dataset.

5.4. Principal Component Analysis (PCA)

PCA is a linear dimensionality reduction technique that can be used to transform a dataset of correlated variables (p) into a smaller k (k<p) number of uncorrelated variables, called principal components. This method can effectively reduce the dimensionality of the data set while the overall variation does not change much. By calculating the cumulative percentage of variation from the most important principal component to the least important principal component, which attributes are more important in the data set can be known. Same to the method above, PCA can be implemented using the PCA function in Scikit-learn. After splitting the undersampled data set, use PCA for model fitting, and set the number of components to 300, then use the transform(X) method to reduce the dimensionality of the data set.

5.5 Result and Finding

The following table is the F2 score of different models analyzing oral toxicity under different dimensionality reduction techniques.

Type of the Classifier	**SelectKBest**	**LDA**	**Fast ICA**	**PCA**	**Undersampling**
Decision Tree Classifier	0.6485	0.4208	0.5476	0.5293	0.6293
K-Neighbors Classifier (KNN)	0.6785	0.4438	0.6460	0.6853	0.7196
Support Vector Classifier (SVC)	0.7498	0.4438	0.7573	0.7466	0.7301
Gradient Boosting Classifier (GB)	0.7198	0.4208	0.6240	0.6881	0.6982
Gaussian NB (NB)	0.5982	0.4464	0.6513	0.5921	0.6421
Logistic Regression	0.6542	0.4436	0.6808	0.6834	0.705
Linear Discriminant Analysis (LDA)	0.6509	0.4436	0.6602	0.6590	0.4611
AdaBoost Classifier	0.6038	0.4208	0.5131	0.5782	0.607
XGB Classifier (XGB)	0.7123	0.4439	0.5917	0.7021	0.7422

Linear Regression	0.6509	0.4436	0.6602	0.6590	0.4611
Multi-layer perceptron (MLP)	0.6856	0.4681	0.6909	0.6903	0.6894

By comparing the performance of different dimensionality reduction techniques, it is found that LDA is not effective because it can only reduce the dataset to one dimension, and it cause for information loss. As can be seen from the above table, after dimensionality reduction using LDA, the F2 scores of all models decrease to around 0.4. On the other hand, K-Selection, ICA, and PCA are similar in performance to undersampling. For the SVC, it even increases the f2 score to 0.7498, which is better than the performance without preform any dimensionality reduction technique. Those methods except LDA reduce the dimensionality of the data by more than half, resulting in a further reduction of training cost and time.

6. Conclusion

To find the effectiveness of the model of classifying toxic and non-toxic chemicals. It was compared with different machine learning models and selected Support Vector Classifier as the best model based on the highest F2 score. After the resampling is performed, the undersampling method is used for the following part. After that dimensionality reduction was performed to reduce the dimensionality of the data and improve the model's performance. For the dimensionality reduction part, except the LDA, PCA, ICA and select K best are performed similar to the undersampling. And the performance of the SVC model slightly increased, while the other models' performance slightly decreased. The combination of model selection and dimensionality reduction resulted in a more efficient and accurate classification of toxic and non-toxic chemicals.

7. Reference

DeepAI. (2020). Gradient Boosting. DeepAI. https://deepai.org/machine-learning-glossary-and-terms/gradient-boosting

Kampakis, S. (2023). Performance Measures: Cohen's Kappa statistic. The Data Scientist. https://thedatascientist.com/performance-measures-cohens-kappa-statistic/

Kundu, R. (2023, February 2). F1 Score in Machine Learning: Intro & Calculation. V7. https://www.v7labs.com/blog/f1-score-guide

Martins, C. (2022, March 26). Gaussian Naive Bayes Explained and Hands-On with Scikit-Learn. Medium. https://pub.towardsai.net/gaussian-naive-bayes-explained-and-hands-on-with-scikit-learn-4183b8cb0e4c

Raj, A. (2021, December 16). Perfect Recipe for Classification Using Logistic Regression. Medium. https://towardsdatascience.com/the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592

Sharma, U. (n.d.). Why do we need data splitting? www.linkedin.com. https://www.linkedin.com/pulse/why-do-we-need-data-splitting-utkarsh-sharma

Techopedia. (2016, September 14). What is a Support Vector Machine (SVM)? - Definition from Techopedia. https://www.techopedia.com/definition/30364/support-vector-machine-svm

The Best Metric to Measure Accuracy of Classification Models - KDnuggets. (n.d.). KDnuggets. https://www.kdnuggets.com/2016/12/best-metric-measure-accuracy-classification-models.html/2

What is the k-nearest neighbors algorithm? | IBM. (n.d.). https://www.ibm.com/topics/knn

Yıldırım, S. (2021, December 13). Gradient Boosted Decision Trees-Explained - Towards Data Science. Medium. https://towardsdatascience.com/gradient-boosted-decision-trees-explained-9259bd8205af