Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

CSE4108: Artificial Intelligence Lab

Fall 2020

Project Report

Glass Types Classification prediction based on Reactive Index and oxides is present the glass.

Lab Section: A1

Submitted To

Md. Siam Ansary	Mr. Ashek Seum
Department of CSE, AUST	Department of CSE, AUST

Submitted By

K.M. Azizullah Tanim

Student ID: 170204001

September 9, 2021

1 Introduction

From USA Forensic Science Service; 6 types of glass; defined in terms of their oxide content. The study of classification of types of glass was motivated by criminology investigation. At the scene of the crime, the glass left can be used as evidence, if it is correctly identified.

In this classification problem we predicted different types of glass based on their refractive index and the oxide content that is present in the glass.

2 A Brief Description of the Dataset

At a Glance Overview

Name of the Dataset	Glass Identification Database
File Format of the Dataset	.csv
Dimension of the Dataset	214 x 10
Number of Total Columns	10
Number of Total Rows	214
Number of Feature Columns	9
Name of Feature Columns	RI, Na, Mg, Al, Si, K, Ca, Ba, Fe
Number of Target Column(s)	1
Name of Target Columns	Type

Description

The dataset has 10 columns and 214 rows. Of 10 columns, 9 columns are feature columns and they are RI(refractive index), and rest of the features column are the unit measurement of weight percent in corresponding oxide; Na(Sodium), Mg(Magnesium), Al(Aluminium), Si(Silicon), K(Potassium), Ca(Calcium), Ba (Barium), and Fe(Iron). The last column is the target column which we are going to predict the value of and the name of that column is Type; which contains the type of the glass. Below is a brief description of each columns:

Name of the Feature : RI(refractive index)
Unit : Float
Description : The refractive index of glass determines how much the path of light is bent, or refracted, while entering the glass. This column contains the refractive index of the glasses which is always greater than one and float value.

Name of the Feature : Na(Sodium)
Unit : Float
Description : This feature column contains the unit measurement of weight percent in corresponding oxide of Na (Sodium) element in the glass. This is always a float value.

Name of the Feature : Mg(Magnesium)
Unit : Float
Description : This feature column contains the unit measurement of weight percent in corresponding oxide of Mg(Magnesium) element in the glass. This is always a float value.

Name of the Feature : Al(Aluminium)
Unit : Float
Description : This feature column contains the unit measurement of weight percent in corresponding oxide of Al(Aluminium) element in the glass. This is always a float value.

Name of the Feature : Si(Silicon)
Unit : Float
Description : This feature column contains the unit measurement of weight percent in corresponding oxide of Si(Silicon) element in the glass. This is always a float value.

Name of the Feature : K(Potassium)
Unit : Float
Description : This feature column contains the unit measurement of weight percent in corresponding oxide of K(Potassium) element in the glass. This is always a float value.

Name of the Feature : Ca(Calcium)
Unit : Float
Description : This feature column contains the unit measurement of weight percent in corresponding oxide of Ca(Calcium) element in the glass. This is always a float value.

Name of the Feature : Ba(Barium)
Unit : Float
Description : This feature column contains the unit measurement of weight per-

cent in corresponding oxide of Ba(Barium) element in the glass. This is always a float value.

Name of the Feature : Fe(Iron)
Unit : Float
Description : This feature column contains the unit measurement of weight percent in corresponding oxide of Fe(Iron) element in the glass. This is always a float value.

3 Description of the Models used in this Project

There are total 6 classification models used in this project. They are:

1. **Logistic Regression**
   First model used in this project is Logistic Regression Model. 80% data was used for training and 20% data was used for testing in this model.

   Since we have more than two categories we used Multinomial logistic regression. Multinomial logistic regression is a classification method that generalizes logistic regression to multiclass problems, i.e. with more than two possible discrete outcomes. That is, it is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables (which may be real-valued, binary-valued, categorical-valued, etc.).

   The Accuracy that we got using this model was 81.4%

2. **KNN (k-nearest neighbors)**
   Second model used in this project is KNN (k-nearest neighbors) Classification Model. 80% data was used for training and 20% data was used for testing in this model.

   The k-nearest neighbors model (k-NN) is a non-parametric classification model. In this classification model, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.k-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this model relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically.

The Accuracy that we got using this model was 76.7%

3. **Support Vector Classifier**
   Third model used in this project is Support Vector Classifier Model. 80% data was used for training and 20% data was used for testing in this model.

   Support-vector classifier methods are supervised learning models with associated learning algorithms that analyze data for classification analysis. SVMs are one of the most robust prediction methods, being based on statistical learning frameworks. More formally, a support-vector classifier method constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.

   The Accuracy that we got using this model was 83.7%

4. **Naive Byes Classifier**
   Fourth model used in this project is Naive Byes Classifier Model. 80% data was used for training and 20% data was used for testing in this model.

   Naive Bayes classifier model is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression; which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

   The Accuracy that we got using this model was 48.8%

5. **Decision Tree Classifier**
   Fifth model used in this project is Decision Tree Classifier Model. 80% data was used for training and 20% data was used for testing in this model.

Decision tree classifier model is a very commonly used classifier model. The goal of this model is to create a model that predicts the value of a target variable based on several input variables. It is a simple representation for classifying examples. Assume that, all of the input features have finite discrete domains, and there is a single target feature called the "classification". Each element of the domain of the classification is called a class. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with an input feature are labeled with each of the possible values of the target feature or the arc leads to a subordinate decision node on a different input feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes, signifying that the data set has been classified by the tree into either a specific class, or into a particular probability distribution, if the decision tree is well-constructed, is skewed towards certain subsets of classes.

A tree is built by splitting the source set, constituting the root node of the tree, into subsets—which constitute the successor children. The splitting is based on a set of splitting rules based on classification features. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has all the same values of the target variable, or when splitting no longer adds value to the predictions. This process of top-down induction of decision trees (TDIDT) is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data.

The Accuracy that we got using this model was 79.07%

6. **Random Forest Classifier**

Sixth model used in this project is Random Forest Classifier Model. 80% data was used for training and 20% data was used for testing in this model.

Random Forest Classifier model is a classification model that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. Random forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

The Accuracy that we got using this model was 90.7%

4 Performance Scores of Each Model

Classification Model	Accuracy	Mean Squared Error	F1 Score	Precision
Logistic Regression	81.4%	1.1628	0.6593	0.7039
KNN (k-nearest neighbors) Classifier	76.7%	1.3953	0.6179	0.6512
SVC Support Vector Classifier	83.7%	0.7907	0.7307	0.7130
Naive Byes Classifier	48.8%	1.3488	0.4368	0.5109
Decision Tree Classifier	79.07%	1.9535	0.5085	0.6886
Random Forest Classifier	90.7%	0.8837	0.7829	0.8528

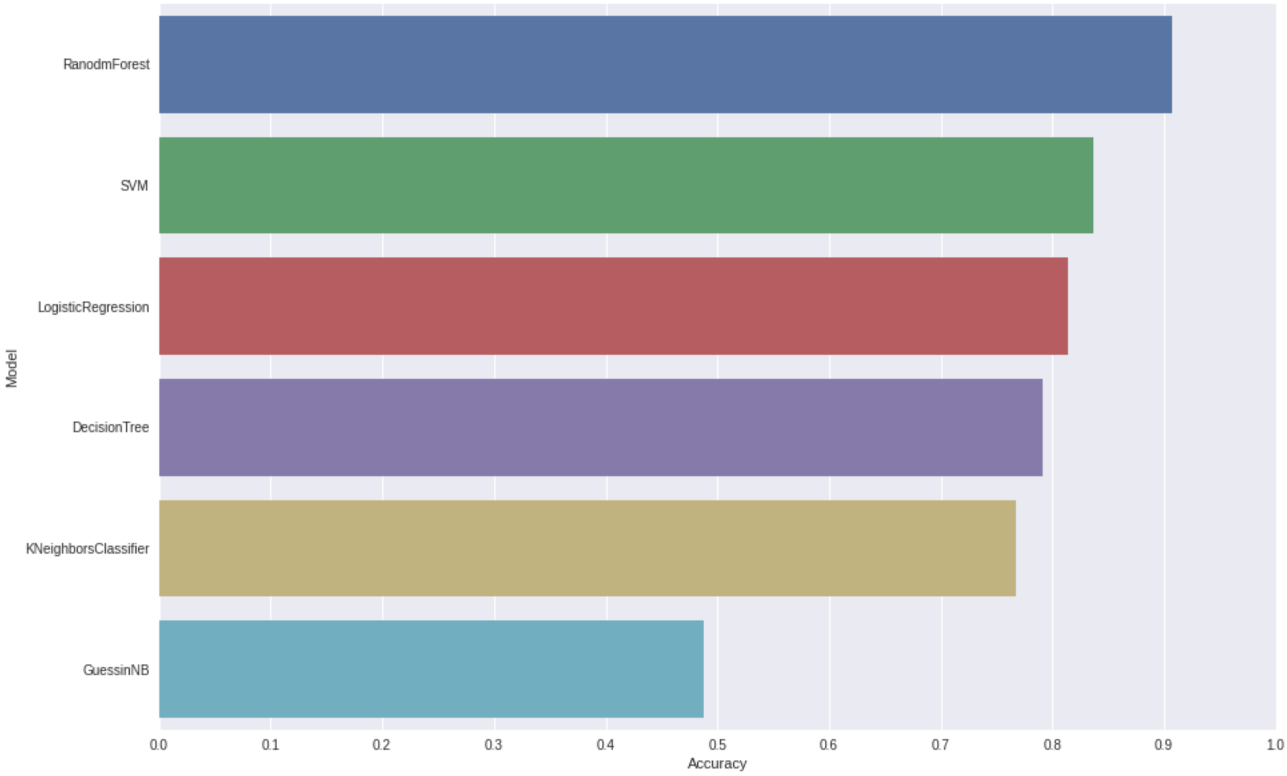

Figure 1: Visual Comparison of Every Model's Performance

5 Conclusion

For Accuracy: From the above figure it clearly shows the Random Forest Classifier model gives higher accuracy compared to other model. And Naive Byes Classifier model gives the lowest accuracy score.

For MSE value: From the above table We can see that Decision Tree Classifier model has the highest MSE value and the Support Vector Classifier model has the lowest MSE value.

For F1 score: From the above table we can see that the Random Forest Classifier model have the highest F1 score compared to other model. And Naive Byes Classifier model gives the lowest F1 score.

For Precision: From the above table we can see that the Random Forest Classifier model have the highest Precision score compared to other model. And Naive Byes Classifier model gives the lowest Precision score.

Now, we can come to the conclusion that, with the given performance scores for the six classification models, we can say both the Random Forest Classifier model and the Support Vector Classifier model are most suitable for this dataset since, Support Vector Classifier model has the least MSE score and the Random Forest Classifier model has the highest accuracy, F1 Score and precision score. and Naive

Byes Classifier model is the least suitable since it has the lowest accuracy, F1 Score and precision score.