Boston University

MET CS 699 A2

Data Mining

Spring 2023

Project Report

Future Asteroids for Material Mining

By: -

Shivesh Raj Sahu and Shriansh Nauriyal

BU ID: -

U28571764 & U35276962

Table of Contents-

Project Background

Asteroid mining is the concept of extracting natural resources from asteroids, which are small, rocky, or metallic bodies that orbit the sun. The idea of asteroid mining has been around for several decades, but it has gained increased attention and interest in recent years due to advances in technology and the growing demand for rare resources.

Asteroids are believed to contain a wealth of valuable resources, including metals such as iron, nickel, and platinum, as well as water and other volatile compounds that can be used for fuel and life support systems. These resources could potentially be used to support long-term space exploration, as well as to meet the growing demand for resources on Earth.

There are several reasons why asteroid mining is seen as a promising option for resource extraction. First, asteroids are relatively close to Earth and can be accessed with current space technology. Second, they contain valuable resources that are becoming increasingly scarce on Earth, such as rare earth elements that are critical for electronics and high-tech industries. Third, mining asteroids could potentially reduce the environmental impact of resource extraction on Earth, as well as mitigate the risk of resource shortages and geopolitical conflicts. However, asteroid mining also poses several challenges and risks.

One of the biggest challenges in this field right now is the identification and categorization of these asteroids into several classes called Tholen classes which tend to have the same general mineral composition. If we are able to identify these classes accurately for nearby asteroids, we can plan out the optimal manner in which we can prioritize which asteroid to target for mining in the future according to our resources and needs.

There are some other challenges such as developing the technology and infrastructure needed to extract resources from asteroids, which would require significant investments and advances in space technology. Identifying the asteroids themselves again will help a lot here by optimizing the infrastructure needed and focussing the technology for that specific class of asteroids.

Despite these challenges, many companies and organizations are actively exploring the possibilities of asteroid mining and investing in research and development in this area. It remains to be seen how feasible and sustainable asteroid mining will be in the long term, but it has the potential to be a breakthrough in space exploration and resource management.

Data Mining Goal

The goal of this project is to accurately predict the class of an asteroid using size, density, orbit, mass distribution, orbit features and many more such features. For the purposes of the project, we will be focussing on a data-focussed approach to predict the classes.

While there are several other classes of asteroids that are present out there, the data that we have today is sufficient in size only for certain classes of asteroids. Nevertheless, the prediction of the classes of these asteroids is a challenging task.

For the purposes of the project, a comprehensive dataset on asteroids have been taken from the NASA databases hosted by CalTech that includes information on their physical and chemical properties, as well as their orbital characteristics.

We will need to apply a series of data pre-processing techniques so that we are able to filter out the set of asteroids that we can feasibly work with. We will then be using several Machine Learning algorithms to predict the classes of these asteroids based on their properties.

It is important to note that accurately predicting the class of an asteroid for mining purposes would require validation through actual mining operations. However, with the increasing interest and investment in asteroid mining, accurate predictions based on comprehensive data analysis could greatly aid in the planning and feasibility assessment of potential mining operations.

Dataset Description

The dataset contains scientific and economic information about asteroids such as their mass, composition, dimensions and other information which also have been classified into specific categories called Tholen classes. Our data mining goal for this project is to build a classification model which can accurately predict this class for unknown asteroids based on this data.

Dataset Source: The primary data source for this classification project has been obtained from the Small-Body Database Query webpage (https://ssd.jpl.nasa.gov/tools/sbdb_query.html) from the Jet Propulsion Laboratory CalTech NASA website.
The git repository for Asterank (https://github.com/typpo/asterank) and the Asterank website (https://www.asterank.com/) was also referred for the dataset.

Dataset Size: The dataset that we are using has a total of **1058** tuples and there are **34** features describing these asteroids.

Class Distribution: The class distribution for the dataset in descending order is as follows:

Classes	Class Counts
S	406

C	261
X	213
B	47
L	34
K	29
V	23
T	13
A	13
D	7
Q	5
R	4
O	3

The attributes in the dataset are as follows:

Name of the attribute	Attribute description
Albedo	Geometric Albedo
PHA	Potentially Hazardous Asteroid Flag
neo	Whether the asteroid is near the earth
Rot_per	Rotation Period
H	Absolute Magnitude Parameter
Diameter	Object Diameter (in km)
Epoch	Osculation
E	Eccentricity
A	Semi-Major Axis
Q	Perihelion Distance
Rot_per	Rotation Period
Diameter_sigma	1-Sigama Uncertainty in Object Diameter
I	Inclination, angle with Respect to X-Y plane
OM	Mean Anomaly

MA	Magnitude
AD	Non Gravitational Parameter
N	Mean Motion
TP	Time of Perihelion Passage
PER	Argument of Perihelion
MOID	Earth Minimum Orbit Intersection Distance
MOID_JUP	Jupiter Minimum Orbit Intersection Distance
T_JUP	Jupiter Tisserand Invariant
Sigma_e	Eccentricity
Sigma_a	Semi-Major Axis
Sigma_q	Perihelion Distance
Sigma_i	Inclination, 1-Sigama Uncertainty
Sigma_om	Mean Anomaly, 1-Sigama Uncertainty
Sigma_ma	Magnitude, 1-Sigama Uncertainty
Sigma_ad	Non Gravitational Parameter, 1-Sigama Uncertainty
Sigma_n	Mean Motion, 1-Sigama Uncertainty
Sigma_tp	Time of Perihelion Passage, 1-Sigama Uncertainty
Sigma_per	Argument of Perihelion, 1-Sigama Uncertainty
RMS	Normalized RMS of Orbit Fit
Neo_flag	Near Earth Object
PHA_flag	Potentially Hazardous Asteroid
Class_var	Class Variable (Type of Asteroid)

Data Mining Tools

The two main data mining tools used in this project are as follows:

1. **R** - The primary programming language that we have used in this project for data preprocessing, model building, visualization and model evaluation is R. It is an open source programming language mainly used for statistical analysis and visualization. It hosts a variety of statistical and graphical techniques including linear modeling, clustering and classification. It also has a very active community of users who contribute to its development and create packages that extend its capabilities.

2. **Microsoft Excel** - We have used Excel to view the dataset and get an overview of the various features that we have to work with for the purposes of model building. Excel is a popular spreadsheet program developed by Microsoft. It allows us to organize, analyze and manipulate data in a tabular format.

Classification algorithms

We are using several classification algorithms as part of this project for predicting the correct class variables for all of the asteroids. These are as follows:

1. **Decision Tree (RPart) -** RPart is a decision tree algorithm present in the 'rpart' package. It is a recursive partitioning algorithm used for classification, regression and survival trees. It works by building a tree structure by recursively splitting the data into smaller subsets based on the values of the input variables. The RPart algorithm is popular because of its simplicity, interpretability and ability to handle both categorical and continuous data.

2. **Random Forest -** Random Forest is an ensemble learning method that is used for classification, regression, and other tasks. It is part of the 'randomForest' package. It combines multiple decision trees to make more accurate predictions than any individual tree.
   The algorithm works by constructing a set of decision trees, each of which is trained on a random subset of the input variables and a random subset of the training data. Each tree in the forest independently predicts the class or value of the target variable, and the final prediction is made by aggregating the predictions of all the trees, usually by taking the majority vote or the average.

3. **Radial SVM** - SVM or Support Vector Machine is a machine learning algorithm that find the optimal hyperplane to separate classes in a high-dimensional feature space. One of the popular hyperplanes is the radial basis function (RBF) kernel which we have used in this project. It is part of the 'kernlab' package. It is widely used in applications such as text and image classification, bioinformatics where the data is often high-dimensional and non-linearly separable.
   Since our dataset is also high-dimensional, this kernel is suited for it.

4. **Neural Net NNet** - A Neural network is a ML algorithm inspired by the structure and function of a human brain. NNet is a type of neural network algorithm used to make feed-forward neural networks with a single hidden layer, and for multinomial log-linear models. It is part of the 'nnet' package in R. It is composed of layers of interconnected nodes, or neurons, that process information through a series of mathematical operations. Neural networks have been successfully used for several ML tasks such as image classification, NLP and speech recognition.

5. **Bagged AdaBoost** - Bagged AdaBoost is an ensemble machine learning algorithm that combines the concepts of bagging and boosting with the AdaBoost algorithm. This is part of the 'adabag' and the 'plyr' package. Bagging (Bootstrap Aggregating) involves randomly sampling the training data with replacement to create multiple subsets, or bags, of data. AdaBoost (Adaptive Boosting) is a boosting algorithm that trains a series of weak classifiers, such as decision trees or stumps, in a sequential manner. Each classifier is trained on a weighted version of the data, where the weights are adjusted to focus on the misclassified samples in the previous iteration. The final prediction is obtained by combining the predictions of these weak classifiers using a weighted sum. Bagged

AdaBoost combines these two techniques by applying bagging to AdaBoost. It creates multiple bags of data and trains AdaBoost on each of them to produce multiple sets of weak classifiers. The final prediction is obtained by combining the predictions of these weak classifiers using a weighted sum. This algorithm has been used in object detection, face recognition and medical diagnosis.

Attribute Selection methods

We have used five different attribute selection methods as part of this project. Attribute selection methods aim to reduce the dimensionality of the feature space by selecting the most relevant and informative attributes for a given task.

For the purposes of our dataset and problem statement, we have decided to use the top 10 attributes according to their importance for the classification task calculated according to the following attribute selection methods.

The attribute selection methods that we have used are as follows:

1. **Fisher's score** - Fisher's score is a statistical measure used for feature selection in machine learning. It evaluates the discriminatory power of a feature by computing the ratio of the between-class variance to the within-class variance. Features with a high Fisher score indicate that they have high discriminatory power and can effectively separate the classes, making them good candidates for feature selection.
   Fisher's score has been widely used in various applications such as image processing, bioinformatics, and natural language processing.

2. **Mean Decreased Accuracy (Random Forest) -** Mean Decreased Accuracy is a measure used in Random Forest feature selection, which evaluates the importance of a feature by computing the decrease in the accuracy of the model when the feature is removed. It works by randomly permuting the values of a feature and measuring the decrease in accuracy of the Random Forest model on the permuted data.
   A large decrease in accuracy indicates that the feature is important for the model's performance, while a small decrease indicates that the feature is less important.
   The method has been applied in various fields such as genetics, image analysis, and bioinformatics.

3. **Mean Decreasing GINI Index (Random Forest) -** Mean Decrease Gini Index is a measure used in Random Forest feature selection that evaluates the importance of a feature by computing the decrease in Gini impurity when the feature is included in the model.
   Gini impurity is a measure of the node purity in decision trees and Random Forests, where a lower impurity indicates a better split between the classes.
   Mean Decrease Gini Index works by randomly permuting the values of a feature and measuring the decrease in Gini impurity of the Random Forest model on the permuted

data. A large decrease in Gini impurity indicates that the feature is important for the model's performance, while a small decrease indicates that the feature is less important.
It is widely used in various applications such as image processing, bioinformatics, and natural language processing.

4. **GBM Variable Importance -** Gradient Boosting Machine (GBM) is a machine learning algorithm that uses an ensemble of weak prediction models, such as decision trees, to build a strong predictive model by iteratively correcting the errors of the previous models in the ensemble.
   GBM Variable Importance is a measure used in the GBM algorithm that evaluates the importance of a feature by computing the total reduction in the loss function when the feature is included in the model. The loss function measures the error between the predicted output and the actual output, and reducing the loss function is the objective of the GBM algorithm.
   Variable Importance works by computing the average reduction in the loss function across all the trees in the GBM model that use the feature. Features with a high average reduction indicate that they have a high impact on the model's performance, while features with a low average reduction are less important.
   The method has been applied in various fields such as finance, marketing, and bioinformatics.

5. **SVM Variable Importance -** As we discussed before, SVM or Support Vector Machine is a machine learning algorithm that finds the optimal hyperplane to separate classes in a high-dimensional feature space.
   Just like GBM, it involves permutation testing involving randomly permuting the values of a feature and measuring the change in the SVM model's performance. Features that result in a large decrease in performance when permuted are considered important.

Attributes selected in the Dataset

The attributes selected by the attribute selection methods listed above are as follows:

Fisher's score	Mean Decreased Accuracy (RF)	Mean Decreasing GINI Index (RF)	GBM Variable Importance	SVM Variable Importance
albedo	diameter	H	albedo	albedo
per	albedo	diameter	moid_jup	diameter
a	a	albedo	a	rms
ad	q	a	sigma_ma	ad
moid_jup	n	ad	n	moid_jup

diameter	per	n	sigma_tp	sigma_ma
moid	moid	per	i	a
sigma_per	t_jup	moid_jup	rms	n
q	sigma_a	t_jup	diameter	per
sigma_a	sigma_per	sigma_per	t_jup	t_jup

Note: Please note that some of these attributes may change for some of the attribute selection methods such as the Random Forest Mean Decrease in Accuracy in consecutive runs even after keeping the same seed in R. The above set of attributes are the ones that we are setting for the model building.

Data Mining Procedure

The data mining procedure involved in our project consists of the following steps:

1. **Data Pre-processing including Data exploration and cleaning**

- The Data pre-processing process starts first with data exploration where we first used Excel to go through the dataset and check if there are any issues with the dataset such as missing data or NA values.
- For our dataset, there weren't any issues such as missing values but there were some issues with the class distributions themselves. After checking the class distribution (also described above in the dataset description section), we recognized that there were some classes which had a very low number of tuples.
- The next step was Data cleaning which we did with the help of R. We first read the csv file as a dataframe and then ran the 'table' function to view the class distribution in descending order.
- We then proceeded to trim the dataset so that we didn't have any class which had a frequency of less than 10. This resulted in a decrease in the overall tuples of our dataset from 1058 to 1039.
- After this, we then proceeded to save the dataframe as a seperate csv file.

2. **Feature Selection**

- For feature selection, we chose the following feature selection methods:
    - Fisher's score
    - Mean decrease in accuracy (RF)
    - Mean decrease in Gini index (RF)
    - GBM Variable importance

    - SVM Variable importance
- These methods were possible with the help of additional libraries such as caret, gbm, randomForest and Rdimtools.
- Fisher's score used the 'do.fscore' function from the Rdimtools library to extract the top 10 columns.
- For the Mean decrease in accuracy and Gini index, we ran a Random Forest algorithm (with the help of the randomForest library) on the dataset and extracted the top 10 most important columns.
- For the GBM and SVM Variable importance methods, we used the inbuilt summary method for the gbm model from the gbm library, and the varImp method from the caret library on the SVM model respectively to extract the top 10 most important columns.
- Please note that the columns that we extracted were done on a seed of 31. However, some of these feature selection methods such as the Mean decrease in Accuracy method still gives slightly different results in consecutive runs.

3. Data Preparation

- In this process, we ran some basic commands to prepare the dataset for model training.
- This involved first changing the class variable into a factor datatype.
- After this, we proceeded to split our dataset into the train and test set. Please note that the seed has already been set at 31 so that we get the same values in the training and test sets everytime we run the code.
- After splitting the dataset, we also save the training and test datasets into 2 separate csv files.

4. Model Training on full dataset

- For the purposes of the project, we first started with training all five models on the full datasets as a comparison against subsets of the datasets.
- For each machine learning model, we have used 10-fold cross-validation.

5. Model Training on the five subsets of the full dataset created with the help of the feature selection methods

- After all the models have been trained on the full datasets, we proceeded to train models on the subsets of the datasets created using the feature selection methods.
- For every feature selection method, we first put the selected features into a vector and then created the subset train and test dataframes using those columns.
- The training of all the ML models from that point onwards followed the same procedure as the model training process on the full dataset.

6. Model Evaluation

- For model evaluation, since our dataset consists of multiple classes (total of 9 after trimming out the 4 classes containing less than 10 tuples each), our confusion matrix is a 9 x 9 matrix.

- The primary model evaluation method, we have used is confusion matrix and the main metric we are focussing on is the accuracy.
- Recall is not overly important in our usecase as predicting the “positive” value over the False “negative” values isn’t important as it is in a medical use-case. Our model at the end of the day is going to be used in a business use-case and from a financial standpoint, accurately predicting the type of asteroid so that we can accordingly prepare for Asteroid mining is most important. Hence, Accuracy is the most important metric.
- For model evaluation metrics, we have recorded the following list of metrics:
    - Confusion matrix
    - TP rate or Sensitivity
    - FP rate or Specificity
    - Precision (most important)
    - Recall
    - F-measure
    - MCC
- We have also recorded the weighted averages of the above metrics
- According to the above metrics, specifically precision, we have chosen the best model from the feature selection methods and compared it to the same model from the dataset with all attributes.

Data Mining Results and Evaluation

Firstly, we will go through the data mining results for the whole dataset:

1. **Decision Tree - RPart**

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	8	63	2	3	1	4	0	38
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	3	7	8	3	136	1	9	19
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	2	12	6	4	3	1	0	16

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.768293	0.794872	0.529412	0.768293	0.626866	0.5027986
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.971429	0.739535	0.708333	0.971429	0.819277	0.6972461

Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.219178	0.900709	0.363636	0.219178	0.273504	0.1470458

Overall Accuracy: 0.6056

Weighted Averages:

TP Rate - 0.6056338
FP Rate - 0.8294821
Precision - NA
Recall - NA
F_Measure - NA
MCC - 0.4408468

2. **Random Forest**

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	Specificity	Precision	Recall	F1	MCC
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

3. **NNet**

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0

B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	1
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	72

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	0.997135	0.857143	1	0.923077	0.9244928
Class: V	1	1	1	1	1	1
Class: X	0.986301	1	1	0.986301	0.993103	0.9913709

Overall Accuracy: 0.9972

Weighted Averages:

TP Rate - 0.9971831
FP Rate - 0.9999516
Precision - 0.9975855
Recall - 0.9971831
F_Measure - 0.9972817
MCC - 0.9962246

4. **Bagged Adaboost**

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	9	69	4	4	1	5	0	40
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	1	4	4	3	125	1	8	13
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	3	9	8	3	14	0	1	20

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.841463	0.769231	0.522727	0.841463	0.64486	0.3941501
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.892857	0.813953	0.757576	0.892857	0.819672	0.5783971
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.273973	0.865248	0.344828	0.273973	0.305344	0.2195964

Overall Accuracy: 0.6028

Weighted Averages:

TP Rate - 0.6028169
FP Rate - 0.8456156
Precision - NA
Recall - 0.6028169
F_Measure - NA
MCC - 0.4422757

5. **SVM - Radial**

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	2	0	0	0	0	0	0	0	0
B	0	5	0	0	0	0	0	0	0
C	0	6	76	5	5	1	4	0	17
K	0	0	0	1	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	4	0	2	3	5	139	1	7	6
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	2	0
X	0	2	4	7	0	0	1	0	50

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0.333333	1	1	0.333333	0.5	0.5740698
Class: B	0.384615	1	1	0.384615	0.555556	0.6130450

Class: C	0.926829	0.860806	0.666667	0.926829	0.77551	0.7109718
Class: K	0.0625	1	1	0.0625	0.117647	0.2446461
Class: L	0	1	NA	0	NA	NA
Class: S	0.992857	0.869767	0.832335	0.992857	0.905537	0.8446328
Class: T	0	1	NA	0	NA	NA
Class: V	0.222222	1	1	0.222222	0.363636	0.4667071
Class: X	0.684932	0.950355	0.78125	0.684932	0.729927	0.6679114

Overall Accuracy: 0.7746

Weighted Averages:

TP Rate - 0.7746479
FP Rate - 0.90628
Precision - NA
Recall - 0.7746479
F_Measure - NA
MCC - 0.6900021

Now, we will go through metrics for feature selected Models:

- **Fisher's Score**

1. Decision Tree - RPart

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	8	63	2	3	1	4	0	35
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	2	4	8	3	119	1	9	10
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	3	15	6	4	20	1	0	28

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.7683	0.8059	0.5431	0.7683	0.6364	0.5159271
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.8500	0.8000	0.7346	0.8500	0.7881	0.6377646
Class: T	0	1	NA	0	NA	NA

Class: V	0	1	NA	0	NA	NA
Class: X	0.38356	0.82624	0.36364	0.38356	0.37333	0.2057451

Overall Accuracy: 0.5915

Weighted Averages:

TP Rate - 0.5915493
FP Rate - 0.8405526
Precision - NA
Recall - 0.5915493
F_Measure - NA
MCC - 0.4232828

2. Random Forest

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**

Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

3. NNet

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0

S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

4. Bagged Adaboost

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	9	69	4	4	1	5	0	40
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	3	4	7	3	128	1	8	14
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	1	9	5	3	11	0	1	19

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.8415	0.7692	0.5227	0.8415	0.6449	0.5325650
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.9143	0.7860	0.7356	0.9143	0.8153	0.6846576
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.26027	0.89362	0.38776	0.26027	0.31148	0.1803186

Overall Accuracy: 0.6085

Weighted Averages:

TP Rate - 0.6084507
FP Rate - 0.8404436
Precision - NA
Recall - 0.6084507
F_Measure - NA
MCC - 0.4495846

5. SVM - Radial

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	10	72	3	4	3	5	0	44
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	3	5	5	5	125	1	9	18
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	0	5	8	1	12	0	0	11

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.8780	0.7473	0.5106	0.8780	0.6457	0.5325650
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA

Class: S	0.8929	0.7581	0.7062	0.8929	0.7886	0.6846576
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.15068	0.90780	0.29730	0.15068	0.20000	0.1803186

Overall Accuracy: 0.5859

Weighted Averages:

TP Rate - 0.5859155
FP Rate - 0.8272782
Precision - NA
Recall - 0.5859155
F_Measure - NA
MCC - 0.4495846

- **Mean Decrease - Accuracy**

1. Decision Tree - RPart

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	10	77	8	7	6	5	0	57
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	3	5	8	3	134	1	9	16
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0

X	0	0	0	0	0	0	0	0	0

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.939024	0.659341	0.452941	0.939024	0.611111	0.5048288
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.957143	0.762791	0.724324	0.957143	0.824615	0.7043124
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0	1	NA	0	NA	NA

Overall Accuracy: 0.5944

Weighted Averages:

TP Rate - 0.5943662
FP Rate - 0.8277652
Precision - NA
Recall - 0.5943662
F_Measure - NA
MCC - 0.4567648

2. Random Forest

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0

B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

3. NNet

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1

Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

4. Bagged Adaboost

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	9	70	4	4	1	5	0	40
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	1	4	4	3	128	1	8	13
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	3	8	8	3	11	0	1	20

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA

Class: B	0	1	NA	0	NA	NA
Class: C	0.853659	0.769231	0.526316	0.853659	0.651163	0.5423714
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.914286	0.813953	0.761905	0.914286	0.831169	0.7128233
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.273973	0.879433	0.37037	0.273973	0.314961	0.1726418

Overall Accuracy: 0.6141

Weighted Averages:

TP Rate - 0.6140845
FP Rate - 0.8485324
Precision - NA
Recall - 0.6140845
F_Measure - NA
MCC - 0.4588347

5. SVM - Radial

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	10	70	4	4	2	5	0	45
K	0	0	0	0	0	0	0	0	0

L	0	0	0	0	0	0	0	0	0
S	6	2	6	4	5	127	1	8	17
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	1	6	8	1	11	0	1	11

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.853659	0.74359	0.5	0.853659	0.630631	0.51506247
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.907143	0.772093	0.721591	0.907143	0.803797	0.66392805
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.150685	0.900709	0.282051	0.150685	0.196429	0.06642382

Overall Accuracy: 0.5859

Weighted Averages:

TP Rate - 0.5859155
FP Rate - 0.8304765
Precision - NA
Recall - 0.5859155
F_Measure - NA
MCC - 0.4190144

- **Mean Decrease - Gini Index**

1. Decision Tree - RPart

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	8	63	2	3	1	4	0	35
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	1	4	4	2	119	1	8	11
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	4	15	10	5	20	1	1	27

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.768293	0.805861	0.543103	0.768293	0.636364	0.5159271
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.85	0.827907	0.762821	0.85	0.804054	0.6675204
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA

Class: X	0.369863	0.801418	0.325301	0.369863	0.346154	0.1635584

Overall Accuracy: 0.5887

Weighted Averages:

TP Rate - 0.5887324
FP Rate - 0.8464538
Precision - NA
Recall - 0.5887324
F_Measure - NA
MCC - 0.4213474

2. Random Forest

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	1	1	1	1	1	1

Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

3. NNet

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0

T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

4. Bagged Adaboost

Confusion matrix:

Reference									

Prediction	A	B	C	K	L	S	T	V	X
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	9	70	4	4	1	5	0	40
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	3	5	6	4	129	1	8	13
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	1	7	6	2	10	0	1	20

Metrics by Class:

	Sensitivity	Specificity	Precision	Recall	F1	MCC
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.853659	0.769231	0.526316	0.853659	0.651163	0.5423714
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.921429	0.786047	0.737143	0.921429	0.819048	0.6915747
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.273973	0.904255	0.425532	0.273973	0.333333	0.2125384

Overall Accuracy: 0.6169

Weighted Averages:

TP Rate - 0.6169014
FP Rate - 0.8426312

Precision - NA
Recall - 0.6169014
F_Measure - NA
MCC - 0.462598

5. SVM - Radial

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	10	73	4	5	2	5	0	45
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	2	6	6	4	129	1	9	17
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	1	3	6	1	9	0	0	11

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.890244	0.739927	0.506944	0.890244	0.646018	0.5409087
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.921429	0.762791	0.716667	0.921429	0.80625	0.6688416

Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.150685	0.929078	0.354839	0.150685	0.211538	0.1141915

Overall Accuracy: 0.6

Weighted Averages:

TP Rate - 0.6
FP Rate - 0.8317955
Precision - NA
Recall - 0.6
F_Measure - NA
MCC - 0.4427187

- **GBM Variable Importance**

1. Decision Tree - RPart

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	10	77	8	7	6	5	0	57
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	3	5	8	3	134	1	9	16
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	0	0	0	0	0	0	0	0

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.939024	0.659341	0.452941	0.939024	0.611111	0.5048288
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.957143	0.762791	0.724324	0.957143	0.824615	0.7043124
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0	1	NA	0	NA	NA

Overall Accuracy: 0.5944

Weighted Averages:

TP Rate - 0.5943662
FP Rate - 0.8277652
Precision - NA
Recall - 0.5943662
F_Measure - NA
MCC - 0.4567648

2. Random Forest

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0

C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

3. NNet

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

4. Bagged Adaboost

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	9	70	2	4	1	5	0	40
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	1	4	4	3	125	1	8	12
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	3	8	10	3	14	0	1	21

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA

Class: C	0.853659	0.776557	0.534351	0.853659	0.657277	0.5504496
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.892857	0.818605	0.762195	0.892857	0.822368	0.6974229
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.287671	0.861702	0.35	0.287671	0.315789	0.1610915

Overall Accuracy: 0.6085

Weighted Averages:

TP Rate - 0.6084507
FP Rate - 0.8484129
Precision - NA
Recall - 0.6084507
F_Measure - NA
MCC - 0.4504427

5. SVM - Radial

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	3	0	0	0	0	0	0	0
C	0	10	65	4	4	3	5	0	45
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	0	9	6	5	128	1	9	16
T	0	0	0	0	0	0	0	0	0

V	0	0	0	0	0	0	0	0	0
X	0	0	8	6	1	9	0	0	12

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0.230769	1	1	0.230769	0.375	0.4735117
Class: C	0.792683	0.739927	0.477941	0.792683	0.59633	0.4617488
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.914286	0.75814	0.711111	0.914286	0.8	0.6573126
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.164384	0.914894	0.333333	0.164384	0.220183	0.1061421

Overall Accuracy: 0.5859

Weighted Averages:

TP Rate - 0.5859155
FP Rate - 0.8270444
Precision - NA
Recall - 0.5859155
F_Measure - NA
MCC - 0.4186842

- **SVM Variable Importance**

1. Decision Tree - RPart

Confusion matrix:

Reference									

Prediction	A	B	C	K	L	S	T	V	X
A	0	0	0	0	0	0	0	0	0
B	0	2	5	0	0	0	0	0	5
C	0	8	58	3	3	1	5	0	35
K	0	0	1	0	0	0	0	0	1
L	0	0	0	0	0	0	0	0	0
S	6	1	4	4	2	119	1	8	11
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	2	14	9	5	20	0	1	21

Metrics by Class:

	Sensitivity	Specificity	Precision	Recall	F1	MCC
Class: A	0	1	NA	0	NA	NA
Class: B	0.153846	0.97076	0.166667	0.153846	0.16	0.12950524
Class: C	0.707317	0.798535	0.513274	0.707317	0.594872	0.45768292
Class: K	0	0.9941	0	0	NA	-0.0163526 4
Class: L	0	1	NA	0	NA	NA
Class: S	0.85	0.827907	0.762821	0.85	0.804054	0.66752036
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.287671	0.819149	0.291667	0.287671	0.289655	0.10736922

Overall Accuracy: 0.5634

Weighted Averages:

TP Rate - 0.5633803

FP Rate - 0.847071
Precision - NA
Recall - 0.5633803
F_Measure - NA
MCC - 0.3892794

2. Random Forest

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1

Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

3. NNet

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	6	0	0	0	0	0	0	0	0
B	0	13	0	0	0	0	0	0	0
C	0	0	82	0	0	0	0	0	0
K	0	0	0	16	0	0	0	0	0
L	0	0	0	0	10	0	0	0	0
S	0	0	0	0	0	140	0	0	0
T	0	0	0	0	0	0	6	0	0
V	0	0	0	0	0	0	0	9	0
X	0	0	0	0	0	0	0	0	73

Metrics by Class:

	Sensitivity	Specificity	Precision	Recall	F1	MCC
Class: A	1	1	1	1	1	1
Class: B	1	1	1	1	1	1
Class: C	1	1	1	1	1	1
Class: K	1	1	1	1	1	1
Class: L	1	1	1	1	1	1
Class: S	1	1	1	1	1	1
Class: T	1	1	1	1	1	1
Class: V	1	1	1	1	1	1
Class: X	1	1	1	1	1	1

Overall Accuracy: 1

Weighted Averages:

TP Rate - 1
FP Rate - 1
Precision - 1
Recall - 1
F_Measure - 1
MCC - 1

4. Bagged Adaboost

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	9	69	4	4	1	5	0	40
K	0	0	0	0	0	0	0	0	0

L	0	0	0	0	0	0	0	0	0
S	6	1	4	4	3	126	1	8	14
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	3	9	8	3	13	0	1	19

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.841463	0.769231	0.522727	0.841463	0.64486	0.5325650
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.9	0.809302	0.754491	0.9	0.820847	0.6945083
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.260274	0.868794	0.339286	0.260274	0.294574	0.1431120

Overall Accuracy: 0.6028

Weighted Averages:

TP Rate - 0.6028169
FP Rate - 0.8445105
Precision - NA
Recall - 0.6028169
F_Measure - NA
MCC - 0.441945

5. SVM - Radial

Confusion matrix:

Reference									
Prediction	**A**	**B**	**C**	**K**	**L**	**S**	**T**	**V**	**X**
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	10	70	4	4	2	5	0	46
K	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0
S	6	2	7	5	5	127	1	9	16
T	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0
X	0	1	5	7	1	11	0	0	11

Metrics by Class:

	Sensitivity	**Specificity**	**Precision**	**Recall**	**F1**	**MCC**
Class: A	0	1	NA	0	NA	NA
Class: B	0	1	NA	0	NA	NA
Class: C	0.853659	0.739927	0.496454	0.853659	0.627803	0.51127543
Class: K	0	1	NA	0	NA	NA
Class: L	0	1	NA	0	NA	NA
Class: S	0.907143	0.762791	0.713483	0.907143	0.798742	0.65481459
Class: T	0	1	NA	0	NA	NA
Class: V	0	1	NA	0	NA	NA
Class: X	0.150685	0.911348	0.305556	0.150685	0.201835	0.08305357

Overall Accuracy: 0.5859

Weighted Averages:

TP Rate - 0.5859155
FP Rate - 0.8281495
Precision - NA
Recall - 0.5859155
F_Measure - NA
MCC - 0.4194693

<u>To conclude:</u>

- As discussed before, the main metric we are focussing on is the accuracy of a model.
- Recall is not overly important in our usecase as predicting the "positive" value over the False "negative" values isn't important as it is in a medical use-case. Our model at the end of the day is going to be used in a business use-case and from a financial standpoint, accurately predicting the type of asteroid so that we can accordingly prepare for Asteroid mining is most important. Hence, Accuracy is the most important metric.

Comparing all the subset dataset models, we can conclude that the RF Model under the Mean Decrease Accuracy is the best model as it along with every other RF model has a 100% accuracy for our usecase. Mean Decrease in accuracy is inherently the best metric for attribute selection method for our business usecase as well.

Comparing the performance of the model with the RF model built on the full dataset, even though the accuracy and other parameters are the same for all metrics on both of the ML models, the fact that the subset model is able to give the same results with almost **1/4th the features**, thus dramatically decreasing the time to train the model, makes the **subset RF model** much more superior.

Discussion and conclusion

In conclusion, the idea of asteroid mining has gained increased attention and interest in recent years due to advances in technology and the growing demand for rare resources. The potential benefits of asteroid mining are vast, with the extraction of a wide range of valuable resources such as common earth metals, rare earth materials, and materials for industrial and business purposes. However, asteroid mining also poses various economic, environmental, and legal challenges that need to be carefully addressed.

In this project, we learned about the importance of comprehensive data analysis and machine learning techniques in accurately predicting the class of an asteroid based on its physical and chemical properties.

We went through several different classification algorithms and concluded that Random Forest was the best choice for our usecase as even with only about 1/4th the features of the entire dataset, we were able to achieve a perfect accuracy with this algorithm, regardless of the attribute selection method that we used with it. Neural Net was a close second algorithm which also had a perfect accuracy with the attribute selection subset of datasets.

With the model that we have now, we will be able to accurately predict any new asteroid which may belong to one of the Tholen classes that we have identified in the dataset above which can help us to accurately estimate the mineral composition of the asteroid for future mining purposes.

What each team member did for this project:

1. Project Background: Done by Shivesh Raj Sahu
2. Data Mining Goal: Done by Shivesh Raj Sahu
3. Data Mining tools: Done by Shriansh Nauriyal
4. Classification algorithms: Done by Shriansh Nauriyal and Shivesh Raj Sahu
5. Attribute Selection methods and Data Mining Procedure: Done by Shriansh Nauriyal
6. Data Mining Results and Evaluation: Done by Shriansh Nauriyal and Shivesh Raj Sahu
7. Discussion and conclusion: Done by Shivesh Raj Sahu

References

Source Links: -

1. Git repo for Asterank data - hLps://github.com/typpo/asterank
2. https://data.world/markmarkoh/future-asteroids
3. NASA Data by Subject hLps://www.nasa.gov/open/data.html
4. hLp://www.ianww.com/latest_fulldb.csv
5. https://en.wikipedia.org/wiki/Asteroid_spectral_types
6. https://sbn.psi.edu/pds/archive/asteroids.html
7. https://sites.google.com/view/sources-asteroidmining/
8. https://data.world/markmarkoh/future-asteroids
9. https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/16-psyche/in-depth/
10. https://www.kaggle.com/datasets/basu369victor/prediction-of-asteroid-diameter

Research Papers and Literature: -

1. How Many Ore-Bearing Asteroids, 2013

   https://arxiv.org/ftp/arxiv/papers/1312/1312.4450.pdf

2. Hayabusa 2, 2019

   In Depth | Hayabusa 2 – NASA Solar System Exploration

3. Asteroid Retrieval Feasibility Study, 2012

   https://www.kiss.caltech.edu/final_reports/Asteroid_final_report.pdf

4. Near-Earth Asteroid Mining, 2001

   https://pdfs.semanticscholar.org/e444/0ba004c28f88a698aa8f08635d5f39187f62.pdf

5. NASA: Asteroids, 2019

   https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/in-depth/

6. Masses of asteroids and total mass of the main asteroid belt, 2016

   https://www.cambridge.org/core/services/aop-cambridge-core/content/view/S1743921315008388

7. Measurement of Gold and Other Metals in Electronic and Automotive Waste Using Gamma Activation Analysis, 2016

   https://link.springer.com/article/10.1007/s40831-016-0051-y

8. REE - Rare Earth Elements and their Uses, retrieved 2019

   https://geology.com/articles/rare-earth-elements/

9. What Is The Environmental Impact Of The Mining Industry?, retrieved 2019

   https://www.worldatlas.com/articles/what-is-the-environmental-impact-of-the-mining-industry.html

10. How can metal mining impact the environment?, retrieved 2019

    https://www.americangeosciences.org/critical-issues/faq/how-can-metal-mining-impact-environment