Prediction of detonation velocity of explosives based on their molecular and crystalline properties

Aleksandra Olshanova
master student

Final project for “Python – data processing, analysis and visualization” course (84-642)

Bar-Ilan University, Israel

2022 – 2023

Contents

Introduction

Explosives are chemical compounds that undergo rapid decomposition, releasing large amounts of energy in the form of heat, light, and gas [1].

The safety of explosives mainly includes sensitivity parameters [2] and detonation parameters [3], which is accurately assessed that will reduce personal injury and economic losses, and provided new ideas and new methods for the performance prediction and design of new explosives. Among them the detonation parameters of explosives mainly include detonation velocity, detonation pressure, detonation heat.

The detonation velocity of an explosive is a measure of the speed at which the detonation wave travels through the explosive material. It is an important characteristic of an explosive because it determines how quickly the energy is released during the detonation process. The detonation velocity of an explosive is influenced by several factors, including the chemical composition and physical properties of the explosive, as well as the conditions under which it is detonated [4].

The purpose of this project is to establish a method for predicting the detonation velocity performance of nitrogen-containing compounds. Also, understanding the factors, which are important for the safe handling and use of explosives, as well as for the development of new explosives with desired detonation characteristics.

Materials & Methods

Missing values

All missing values were filled with the help of the Tanimoto similarity method, which allows to screen the dataset for similar compounds. The use of fingerprints and the Tanimoto coefficient represents one of the most popular methods for quantifying molecular similarity. Fingerprints encode structural features of a molecule in a binary vector format and the Tanimoto coefficient quantifies the overlap of features of two molecules as the ratio of the number of common features to the total number of features in each fingerprint. The Tanimoto coefficient has the value range 0 to 1 and can be interpreted as the percentage of features shared by two molecules [5].

SMILES cleaning and descriptors calculation

The RDKit (Rapid Development Kit) library and mols2grid [6], which is an interactive molecule viewer for 2D structures, based on RDKit, were used to analyze and visualize chemical structures. Rdkit. Chem.Descriptors module, a popular open-source cheminformatics

toolkit written in C++ and Python, was applied to generate molecular descriptors that can be used to predict new properties. ChemDraw generated SMILES strings for structures with errors.

Data visualization

Seaborn graphic visualization library was used to find relationships between different features and represent them. It is built on the primary configurations of Matplotlib, and provides accessibility to the users with some of the most commonly provides data visualizations processes with certain data visualizations necessities such as mapping colour to a variable or using faceting requirements across the globe [7].

Dimensionality reduction

The most frequently used approach for reduction of dimensionality PCA (principal component analysis) was applied to reduce the high dimensions in the data set to fewer dimensions. The aim of PCA is to find an optimal position for the best information variance and vector dimensional features reduction. The low dimensional data representation of the initial data can be easily analyzed, processed, and visualized [8].

Clustering of molecules

In order to put similar data items in a same group clustering k-mean technique was used. According to the basic k-mean clustering algorithm, clusters are fully dependent on the selection of the initial clusters centroids. K data elements are selected as initial centers. Then distances of all data elements are calculated by Euclidean distance formula. Data elements having less distance to centroids are moved to the appropriate cluster. The process is continued until no more changes occur in clusters [9].

Regression models

Several regression models were used in this project to compare their performance on given data set: linear regression model, random forest regression, decision tree regression.

The linear regression is one of the simplest and most common machine learning algorithms. It is commonly used in mathematical research methods, where it is possible to measure the predicted effects and model them against multiple input variables [10].

The regression decision tree is an iterative process that splits the data into partitions. Initially, all the training samples are used to determine the structure of the tree. The algorithm then breaks the data using every possible binary split and selects the split that partitions the data into two parts such that it minimizes the sum of the squared deviations from the mean in the separate parts. The splitting process is then applied to each of the new branches. The process

continues until each node reaches a user-specified minimum node size (i.e., the number of training samples at the node) and becomes a terminal node [11].

Random forest is an ensemble of decision trees, and it trains many decision trees in parallel. Each decision tree is trained on only a random subset of observations, and the predictions are combined into one decision tree [12].

Model evaluation

To evaluate the performance of regression models metrics below were used [13]:

Mean Absolute Error (MAE) measures the average absolute difference between predicted and actual values. It gives an idea of how close the predictions are to the true values, but doesn't penalize large errors as much as other metrics.

Mean Squared Error (MSE) measures the average squared difference between predicted and actual values. It penalizes larger errors more heavily than MAE and can be useful when large errors are particularly undesirable.

Root Mean Squared Error (RMSE) is the square root of the MSE and gives an idea of how much the predictions deviate from the true values on average. It has the same unit as the target variable and can be more easily interpreted than the MSE.

Results & Discussion

The data was imported into a dataframe by using the “pandas” library in Python. The following changes were made to give appropriate names to all columns for convenient work later.

Exploratory data analysis and visualization

First of all, it was vital to analyse the data and identify null values, outliers and missing values. By using *info()* and *isnull()* functions all missing values were defined and visualized with heatmap (**Figure 1**), which is a graphical representation of data in which values are depicted as colors within a two-dimensional matrix. Overall, there was one missing value in the ‘Packing coefficient’ column, and two missing values in the ‘Hydrogen bond area’ column.

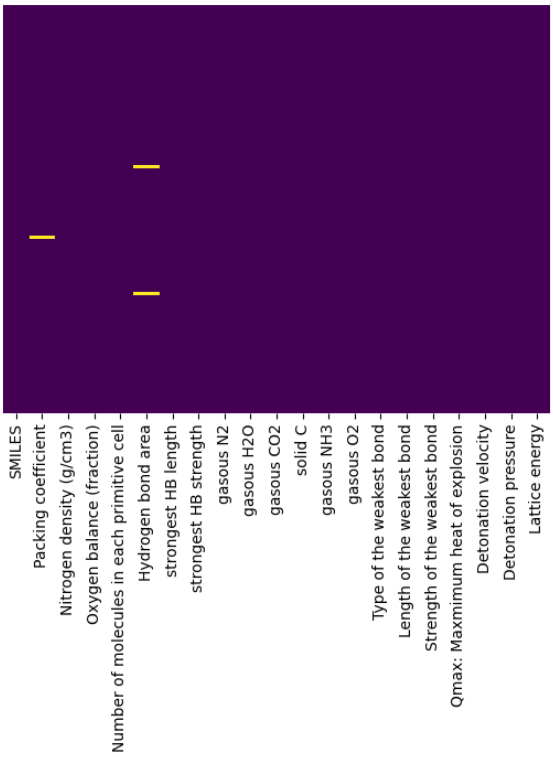


Figure 1. Graphical representation of missing values of the data.

It is important to handle missing values appropriately as they can affect the accuracy of statistical analyses and machine learning models.

By reviewing info analysis it was also found that the ‘Lattice energy’ column has an object dtype, which is not appropriate for numerical values. With the help of *pd.to_numeric()* function dtype was changed to float64, and at the same time the weird value “72.63/72.61” in one row was substituted by mean value “72.62”.

The ‘Type of the weakest bond’ column has similar type of bonds, but Python considers them as different unique values, because of space in some strings: 'N-C', 'C-C', 'N-C ', 'O-C', 'C-C ', 'N-N', 'O-N', ' O-C'. Eventually, unnecessary spaces were deleted.

Outliers in a dataframe refer to observations that are significantly different from the other observations in the same dataset. These can be due to measurement errors, data entry errors, or genuine extreme values. There are several ways to identify outliers in a dataframe, including plotting the data using box plots, histograms, or scatter plots. In this project box plot method and *describe()* function were used to find out outliers. First, a box plot was used to visualize the unscaled data (**Figure 2**), and after removing outliers in the ‘Packing coefficient’ column the same technique was implemented to observe additional outliers (**Figure 3**).

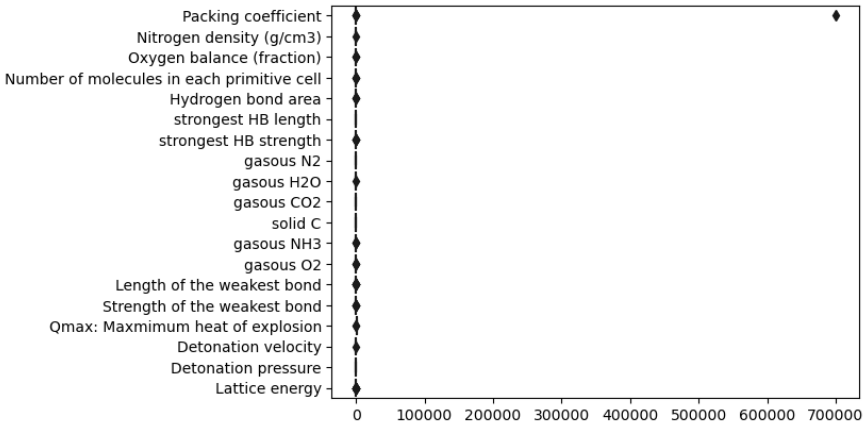


Figure 2. Graphical representation of unscaled data.

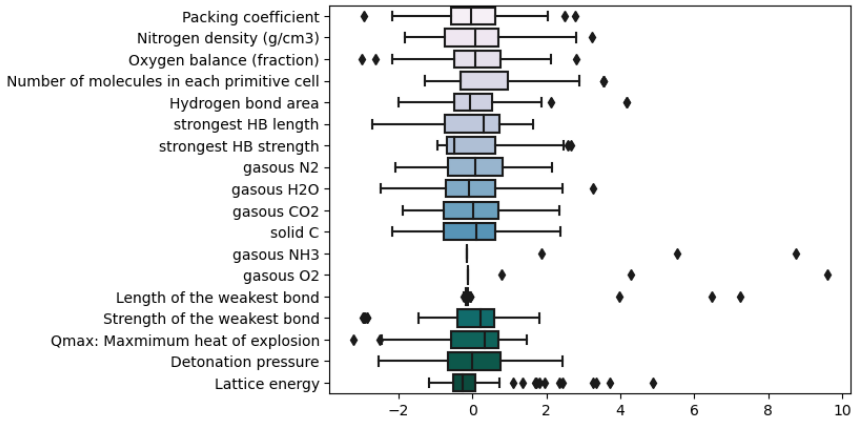


Figure 3. Graphical representation of scaled data after removing outliers in 'Packing coefficient' column.

Some outliers were found in the 'Length of the weakest bond' column. Particularly, some of the values for bond length were higher than 10 angstrem, which is not possible for this property. By checking the type of the bond, it was found that outliers exist only for 'N-N' type of bond. All rows with 'N-N' type bonds were reviewed and the mean value for their length was calculated. Then, all outliers were substitutes with this mean value.

SMILES representation of molecular structures

Some errors in SMILES strings were found during visualization of all molecular structures with *Draw.MolsToGridImage()* function. Particularly, two SMILES strings were not correct, and new relevant SMILES representations were generated for these strings by using ChemDraw. Moreover, some strings represented mixes and two similar compound structures in one unit of cell. Additionally, SMILES of some compounds were coded together with solvents. All these errores are illustrated in **Figure 4**.

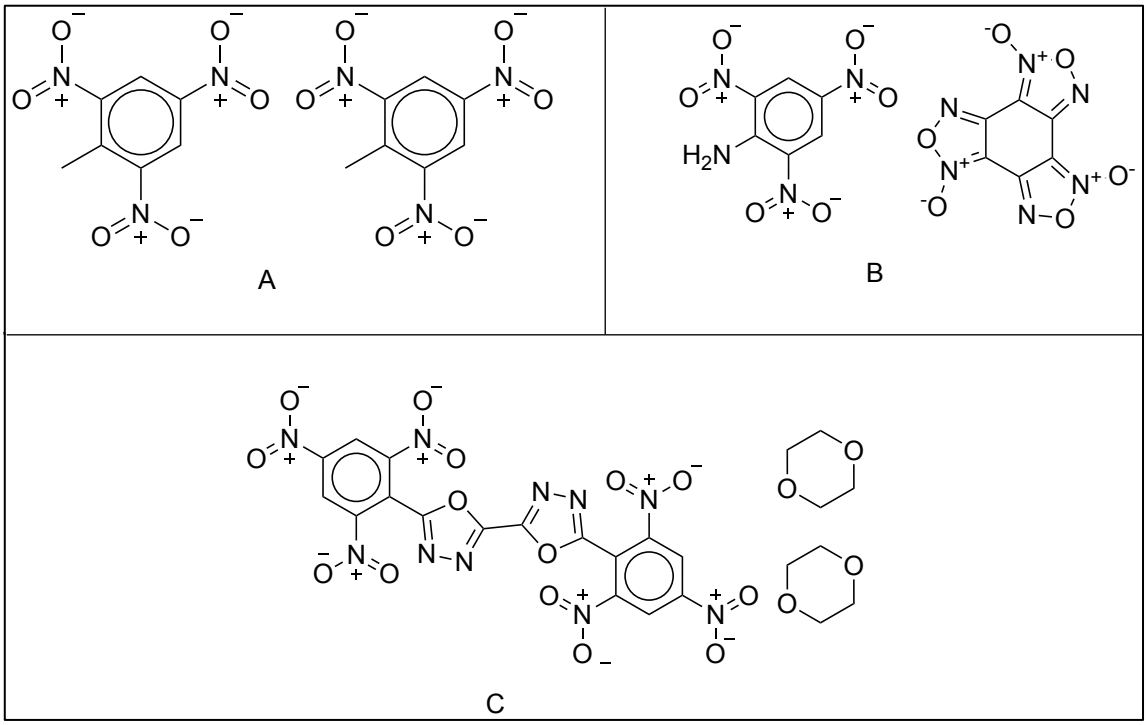


Figure 4. Graphical representation of errors in SMILES strings. **A**: similar compound structures in one string; **B**: mixes; **C**: molecules with solvent.

First of all, strings with solvents were identified, and illustrated solvents in **Figure 5** were deleted out of the strings. All strings which present two the same compounds were defined, and the SMILES string for one compound was left. Finally, all mixes were deleted.

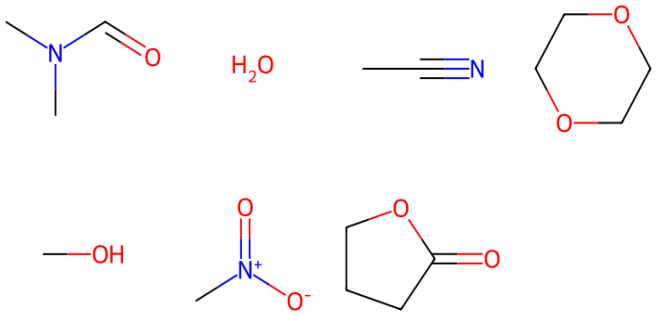


Figure 5. Solvents found in SMILES strings.

Later, the whole DataFrame was checked on duplicates in the SMILES column, and 8 duplicates were identified. The *drop_duplicates* function modified the original DataFrame by removing the duplicate rows based on the SMILES column and updated it with only unique rows (inplace=True parameter was set).

So, after cleaning the SMILES strings and deleting some outliers, 92 unique rows were left.

Values for molecules missing data

All missing values in the ‘Packing coefficient’ column and in the ‘Hydrogen bond area’ were calculated with the Tanimoto similarity method. Indeed, the Tanimoto similarity is often used in cheminformatics to compare the similarity of chemical compounds. It is based on the presence or absence of certain molecular features or fingerprints that are characteristic of the compounds. By using *AllChem.GetMorganFingerprintAsBitVect()* function, fingerprints for all compounds were calculated, and later were used for Tanimoto coefficient calculation with the help of *DataStructs.TanimotoSimilarity()* function.

There is one missing value in the ‘Packing coefficient’ column. With the help of the method described above, the structure with the Tanimoto similarity coefficient 0.65 was found, and the ‘Packing coefficient’ value 75.8 was added. The structure with missing value, and similar compounds are illustrated in **Figure 6**.

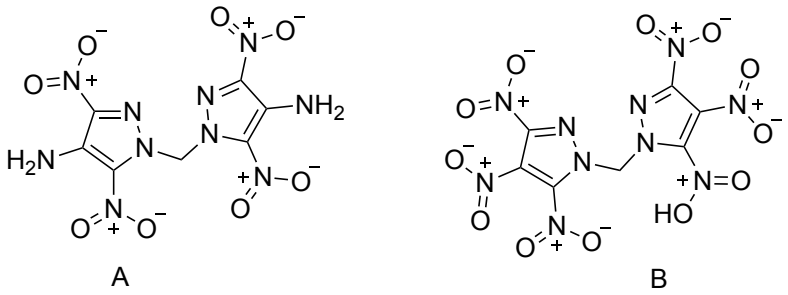


Figure 6. A: molecule with a missing value in the ‘Packing coefficient’ column; **B**: found molecule with the Tanimoto similarity method.

There are two missing values in the ‘Hydrogen bond area’ column. With the help of the same method described above, the structures with the Tanimoto similarity coefficient 0.69 and 0.8 were found, and corresponding ‘Hydrogen bond area’ values 134.63 and 90.43 were added. The structures with missing values, and similar compounds are illustrated in **Figure 7**.

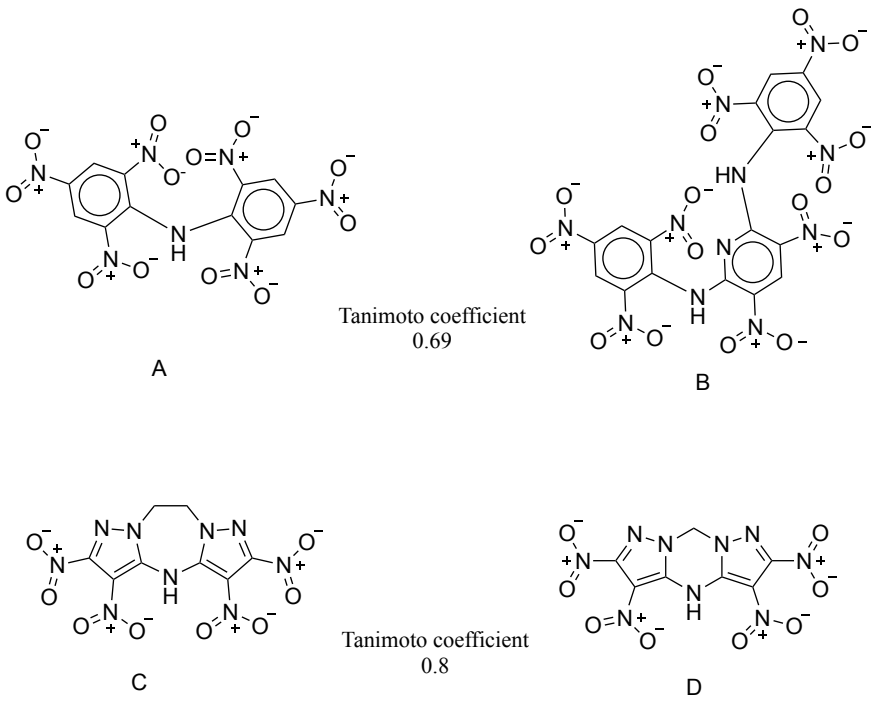


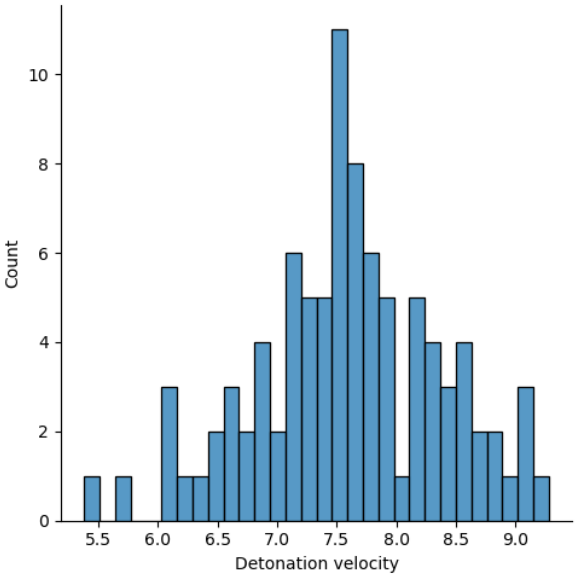
Figure 7. A, C: molecule with a missing value in the 'Hydrogen bond area' column; **B, D**: found molecules with the Tanimoto similarity method.

Descriptors

Some basis descriptors for molecules were calculated with the *rdkit.Chem.Descriptors* module, and then added to the DataFrame. First descriptor is MolLogP, or molecular octanol-water partition coefficient, which is a measure of the hydrophobicity of a molecule. It is defined as the logarithm of the ratio of the concentrations of the molecule in octanol and water at equilibrium. Also, molecular weight of a molecule (MolWt) was calculated to better analyze and compare molecules based on their physicochemical properties. However, some descriptors decrease the quality of the final model, for instance, the Topological Polar Surface Area (TPSA). TPSA is used to estimate the polarity and water solubility of a molecule. It represents the surface area of a molecule that is occupied by polar atoms or polar groups, such as hydroxyl (-OH), amine (-NH2), and carbonyl (-CO) groups. The TPSA is measured in square angstroms ($Å^2$). Finally, this descriptor was deleted so as not to impact on the model performance.

Data visualization

The "target property" in this project is the 'Detonation velocity', which refers to the variable that the machine learning model is trying to predict or estimate based on the input features or variables. Some visualization techniques were implemented to see the features'



influence on the target property. First of all, *sns.displot()* function was used to understand the distribution of the 'Detonation velocity' values (**Figure 8**). All values are in the range 5.38 - 9.28 (km/s), but most of them are around 7.5 km/s. As 'Detonation velocity' values are continuous numerical values, the linear regression machine learning algorithm was used for their prediction. To do this, the relationship between the feature and the target variable were estimated with *sns.heatmap()* method and calculated correlation coefficients.

Figure 8. 'Detonation velocity' values distribution.

There are some features which correlate more with the target property and their corresponding correlation coefficients: 'Oxygen balance' (0.84), 'gasous CO_2' (0.72), 'solid C' (-0.81), 'gasous N_2' (0.58). The *sns.regplot()* method was used to illustrate the relationships between these features (**Figure 9**).

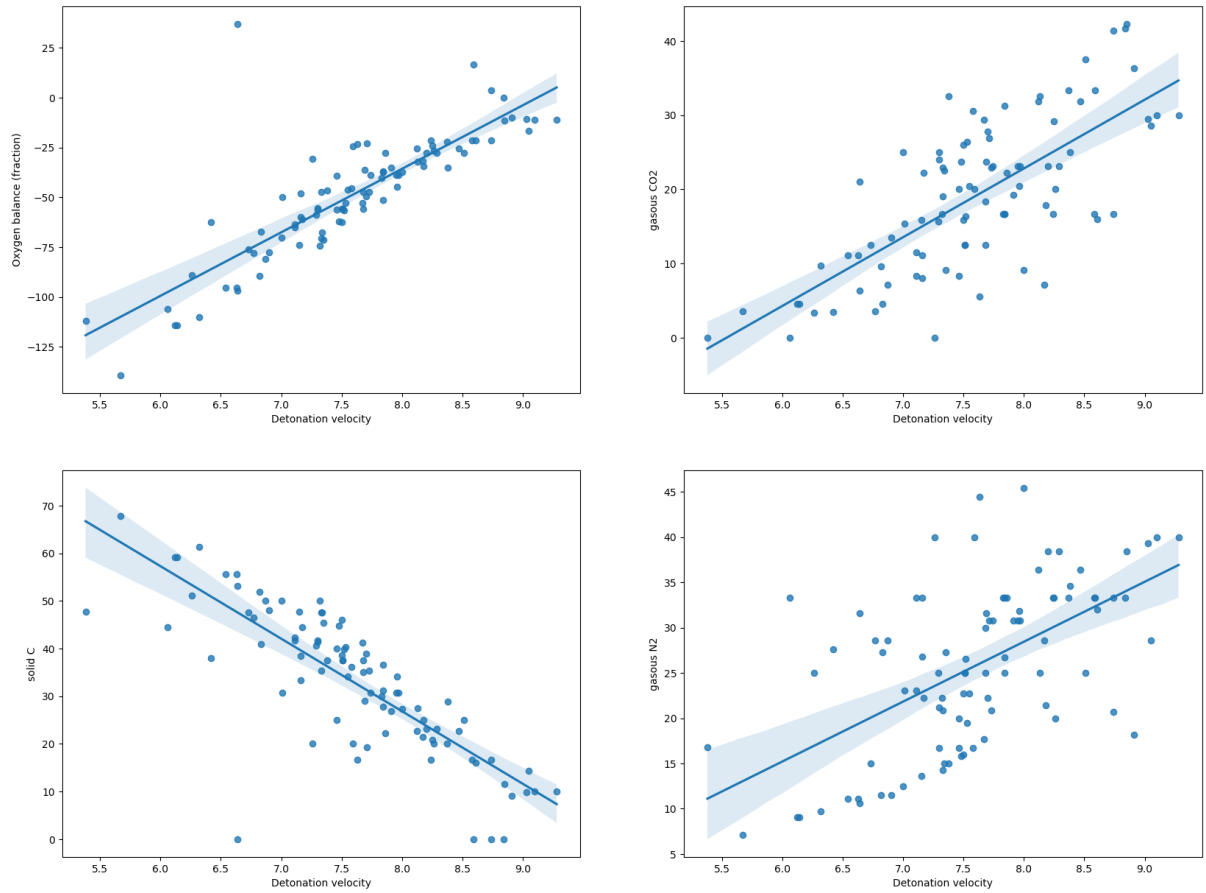


Figure 9. The relationships between features that correlate with the 'Detonation velocity'.

Data standardization

In order to make the data more comparable and easier to analyze, especially when there are features with different units or scales, a data standardization step was performed to transform the data to have a consistent scale and distribution. Some columns with object dtype of the values and target property were excluded from the standardization. The *boxplot()* method was used to illustrate the scaled data in **Figure 10**.

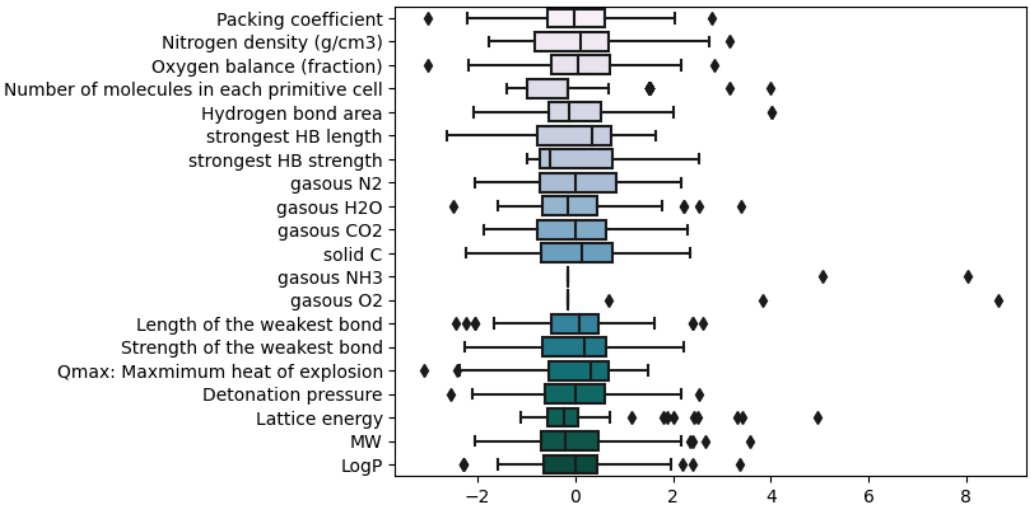


Figure 10. Graphical representation of scaled data.

Dimensionality reduction

The initial dataset contains many variables, and to identify the most important features in a dataset and transform the data into a lower-dimensional space while retaining as much information as possible, the PCA (Principal Component Analysis) technique was used.

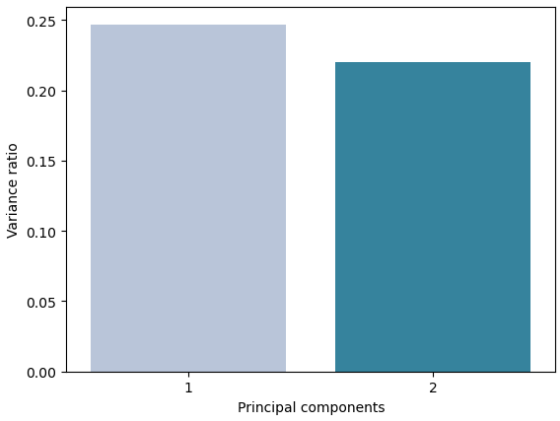


Figure 11. The percentage of variance explained by each principal component.

The first step in analyzing a PCA plot was to look at the percentage of variance explained by each principal component. This helped to understand how much of the variability in the data is captured by each component. In particular, the first principal component captured almost 25 % of the data, and the second principal component a little bit less (22%). The variance ratio of each component is illustrated in **Figure 11**.

The result of a PCA analysis is typically represented as a plot, with each point on the plot representing a data point in the original dataset (**Figure 12**). Each principal component represents a combination of the original variables in the dataset. The axes of the PCA plot can therefore be interpreted as a representation of these variables. By analyzing the plot, it is possible to look for groups of points that are clustered together and indicate some patterns and relationships in the data. All points on the plot were colored based on their "Detonation velocity" value (target property). As a result, the first principal component describes the data more sufficiently, and there is a pattern in clustering if we go along the "Component 1" axe.

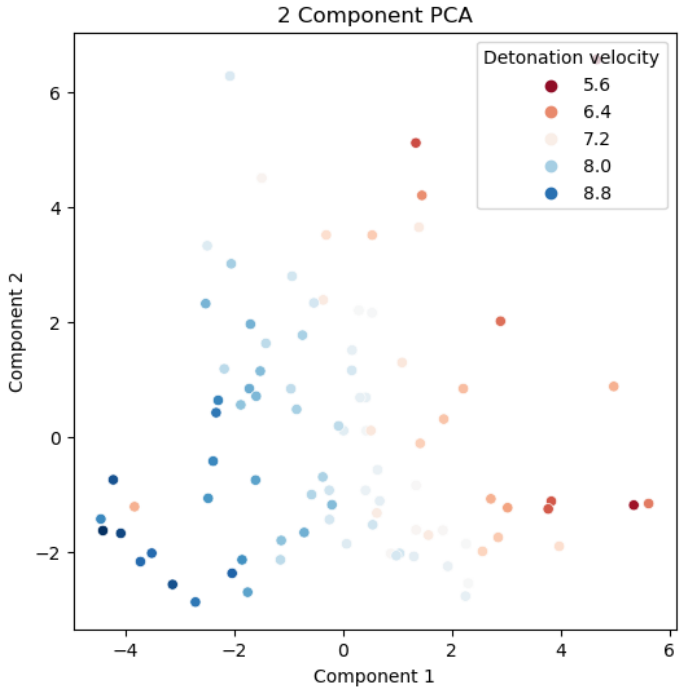


Figure 12. PCA plot.

Clustering of molecules

The k-means method was implemented for grouping similar molecules based on their molecular and crystalline properties. Different numbers of clusters (2 and 3) were experienced to define best groups with scaled data described above. Also, the elbow method was used to determine the optimal number of clusters in a k-means clustering algorithm (**Figure 13**). The method works by plotting the within-cluster sum of squares (WCSS) against the number of clusters used in the algorithm.

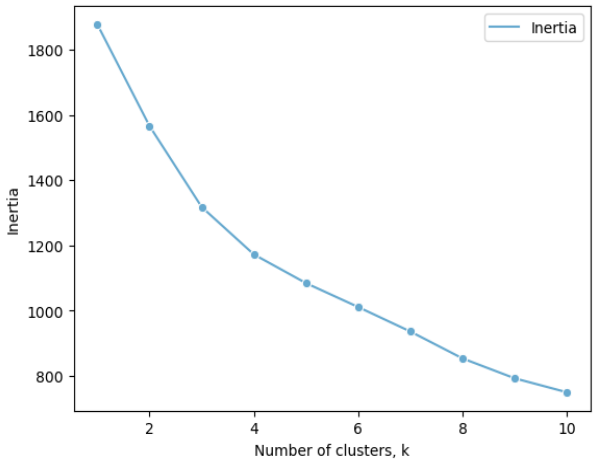


Figure 13. Elbow method results.

In order to select the optimal number of clusters, I looked for the "elbow" point in the plot, where the rate of decrease in WCSS slows down. The possible and optimal number of clusters were chosen as 4. After applying the k-means algorithm to the standardized data, the scatter plot of the data with points colored according to the assigned cluster was created (**Figure 14**). The second cluster represents only two points, but other clusters contain around 30% each of the data.

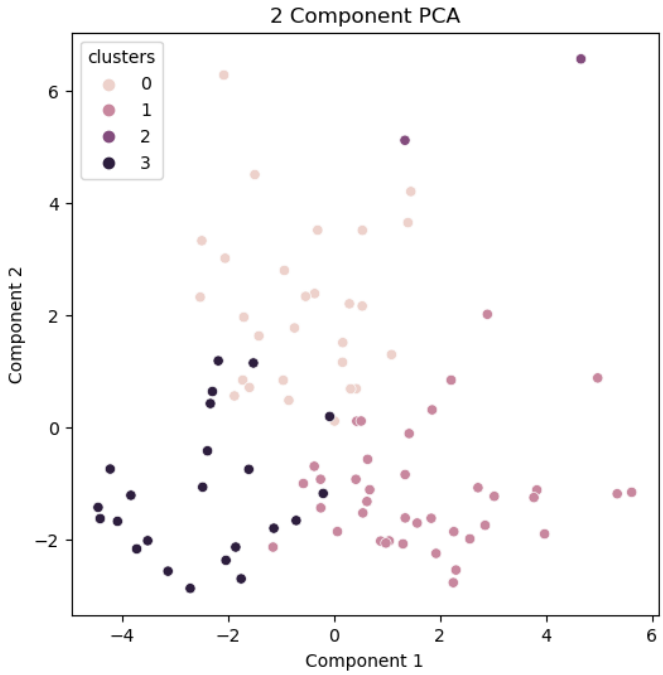


Figure 14. The scatter plot of the data with points colored according to the assigned cluster.

The k-means clustering provides insights and understanding of the structure of the data by identifying patterns, relationships, and groups of similar data points. By analyzing obtained clusters I didn't find features that are extremely different from other clusters, and are capable of describing this unique cluster.

Building predictive model

First of all, the data was split into training and test sets. This is an essential step in building a machine learning model. The data was splitted into training and test sets using the *train_test_split()* function with a test size of 0.3, which means 30% of the data is reserved for testing. 'X_train' and 'X_test' represent the feature scaled matrix (without 'clusters', 'Detonation velocity' features) of the training and test sets. 'y_train' and 'y_test' represent the target unscaled variable ('Detonation velocity') of the training and test sets.

Regression models

Linear regression model, random forest regression, decision tree regression methods were implemented for building prediction models. A scatter plot with a regression line was used for visualizing the relationship between predicted and actual target variable (**Figure 15**).

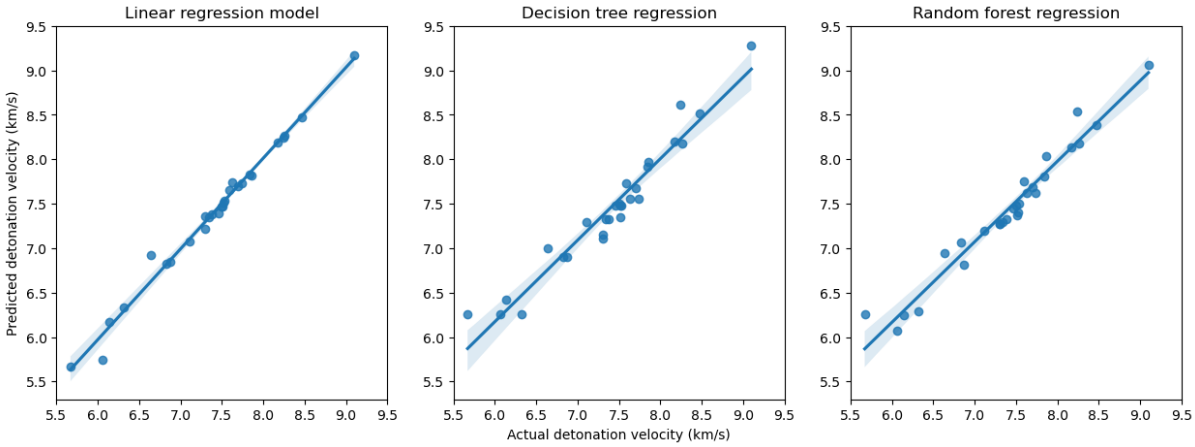


Figure 15. Scatter plots with a regression line for predictive models.

Some common metrics were used to evaluate the performance of regression models, which are mentioned in the **Table 1**.

Table 1. Model validation.

Metric	Linear regression model	Random forest regression	Decision tree regression
Mean Absolute Error	0.04759	0.10450	0.13429
Mean Squared Error	0.00795	0.02604	0.03526
Root Mean Squared Error	0.00199	0.00361	0.00420

Linear regression models show better prediction results based on the above metrics.

Conclusions

First of all, a method for predicting the detonation velocity performance of nitrogen-containing compounds based on their properties was established. Studies have shown that 'Oxygen balance', 'gasous CO_2', 'solid C', 'gasous N_2' features have a strong influence on the regression models. Compared with other regression methods (random forest regression, decision tree regression), linear regression method has the advantages of accurate prediction results, and better metrics for model validation (MAE, MSE, RMSE).

References

[1] He T. et al. The detonation heat prediction of nitrogen-containing compounds based on quantitative structure-activity relationship (QSAR) combined with random forest (RF). // Chemometrics and Intelligent Laboratory Systems. **2021**.V. 213.

[2] Sikder A. K., Sikder N. A review of advanced high performance, insensitive and thermally stable energetic materials emerging for military and space applications. // Journal of Hazardous Materials. **2004**. V. 112. №1–2. P. 1–15.

[3] Zhang C. Origins of the energy and safety of energetic materials and of the energy & safety contradiction. // Propellants, explosives, pyrotechnics. **2018**. V. 43. №9. P. 855–856.

[4] Bohanek V. et al. Effect of confinement on detonation velocity and plate dent test results for ANFO explosive. // Energies (Basel). **2022**. V. 15. №12.

[5] Vogt M., Bajorath J., Ccbmlib - A python package for modeling tanimoto similarity value distributions. // F1000Research. **2020**. V. 9.

[6] RDKit: Open-source cheminformatics. https://www.rdkit.org

[7] Ali Hassan Sial et al. Comparative analysis of data visualization libraries matplotlib and seaborn in Python. // International Journal of Advanced Trends in Computer Science and Engineering. **2021**. V. 10. №1. P. 277–281.

[8] Salih Hasan B. M., Abdulazeez A. M. A review of principal component analysis algorithm for dimensionality reduction. // Journal of Soft Computing and Data Mining. **2021**. V. 2. №1. P. 20-30.

[9] Rauf A. et al. Enhanced k-mean clustering algorithm to reduce number of iterations and time complexity. // Middle-East Journal of Scientific Research. **2012**. V. 12. №7. P. 959–963.

[10] Maulud D., Abdulazeez A. M. A Review on linear regression comprehensive in machine learning. // Journal of Applied Science and Technology Trends. **2020.** V. 1. №4. P. 140–147.

[11] Xu M. et al. Decision tree regression for soft classification of remote sensing data. // Remote Sensing of Environment. **2005**. V. 97. №3. P. 322–336.

[12] Murugan S. et al. Classification and prediction of breast cancer using linear regression, decision tree and random forest. // International Conference on Current Trends in Computer, Electrical, Electronics and Communication (ICCTCEEC-**2017**).

[13] Chicco D. et al. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. // PeerJ Computer Science. **2021**. V. 7. P. 1–24.