Composit 2024

Excavate

Team : Data_Riders

CONTENTS

Overview

Scenario

Predicting how well a metallic alloy will form a glass can be a slow, trial-and-error process. AI and data science are changing that. By analysing datasets of existing metallic glasses, machine learning models can predict the glass-forming ability of new alloys, reducing the need for extensive experimentation. This not only saves time and resources, but can also lead to the discovery of entirely new glass compositions with superior properties.

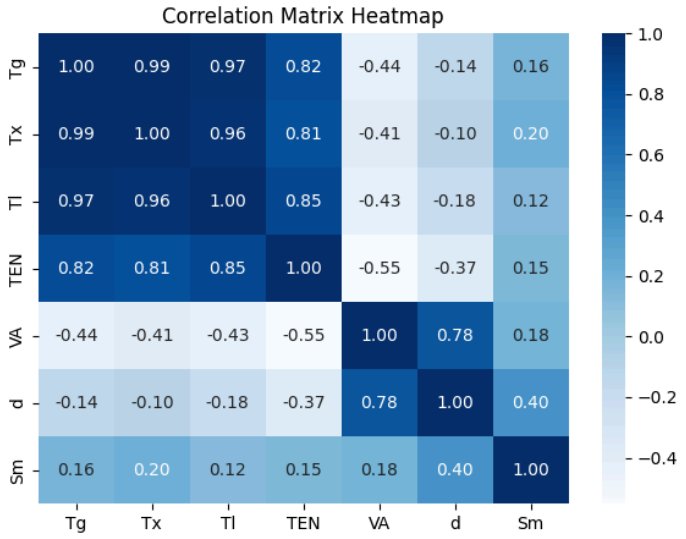
Problem

Design and implement a machine learning model to predict the glass forming ability (GFA) of metallic glass, expressed as Dmax (in mm), the target variable. It is theorised that GFA depends on several intrinsic parameters like total electronegativity (TEN), atomic size difference (d), average atomic volume (VA), mixing entropy (Sm), glass-transition temperature (Tg), onset crystallisation-temperature (Tx), and liquidus temperature (Tl). Develop an robust and accurate predictive model that can predict the Dmax of a metallic glass sample, given the input features

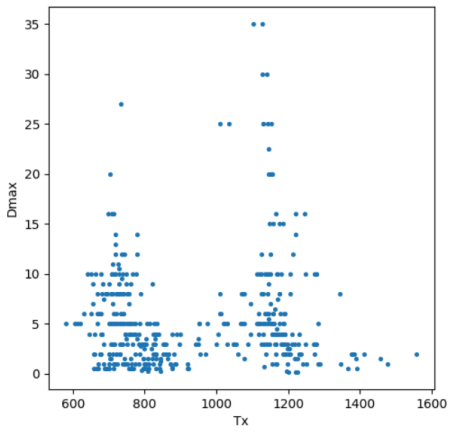
Approach

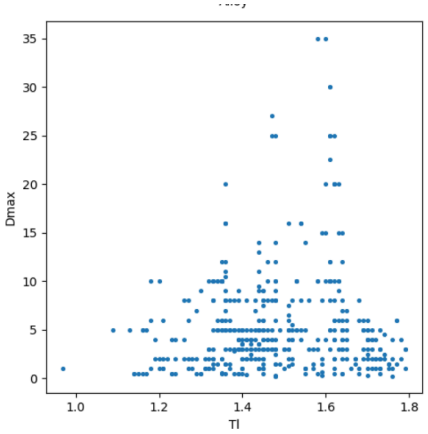
In order to address the issue of a small dataset and having only a single input for some Dmax values, we have employed various data augmentation methodologies, so as to have sufficient data to train our model and to accurately give the required result. Further, we employ Exploratory Data Analysis(EDA) to obtain an understanding of the data, and Principal Component Analysis(PCA), to plot the data into graphs, to realise the potential of the dataset. A comprehensive approach was adopted, after trying multiple regression models, the XGBoost model was used to implement regression on the data, and obtain a precise estimation. The mean squared error(MSE) has been used as a yardstick to determine the effectiveness of different models. We have implemented regression through six models, determining training and testing MSEs for each one of them. Using this selection criteria, we have arrived at the conclusion that XGBoost is the best model among them all.

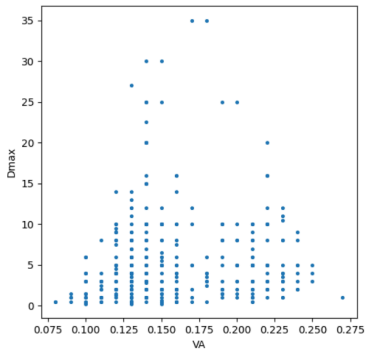
Exploratory Data Analysis



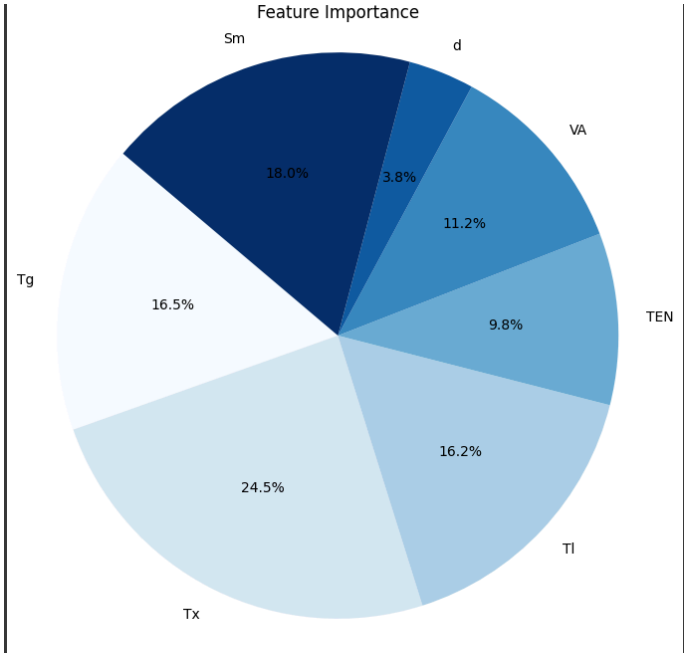
The adjoining plot is a correlation matrix heatmap, showing the relation between the seven input features. This heatmap gives us an idea that if two features are highly correlated, one of the features can be dropped. We can see that the variables glass-transition temperature (Tg), onset crystallisation-temperature (Tx) and liquidus temperature (TI) are highly correlated with each other, thereby any two of these features can be excluded from the dataset. This leads to the formation of simpler models that are preferred because they are easier to interpret and less prone to overfitting. Correlated features can lead to multicollinearity in regression models. Multicollinearity makes it challenging to estimate the individual effect of each feature on the target variable. It can also cause unstable coefficient estimates and inflated standard errors.



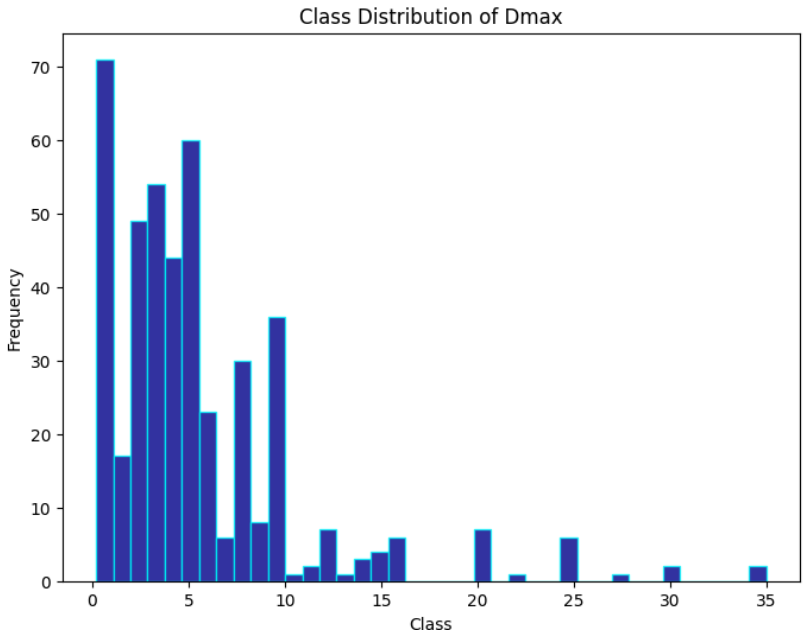




The following series of plots shows the relation of Dmax, the target variable, with different features such as Tx, TI, and VA. Such plots are quite useful while performing data analysis. Positive correlations (when the target increases with the input feature) and negative correlations (when the target decreases with the input feature) become evident. Outliers or nonlinear relationships may also be visible. Examining feature-target plots helps identify which input features have a significant impact on the target variable. Features with strong correlations or clear patterns are likely more important for prediction.

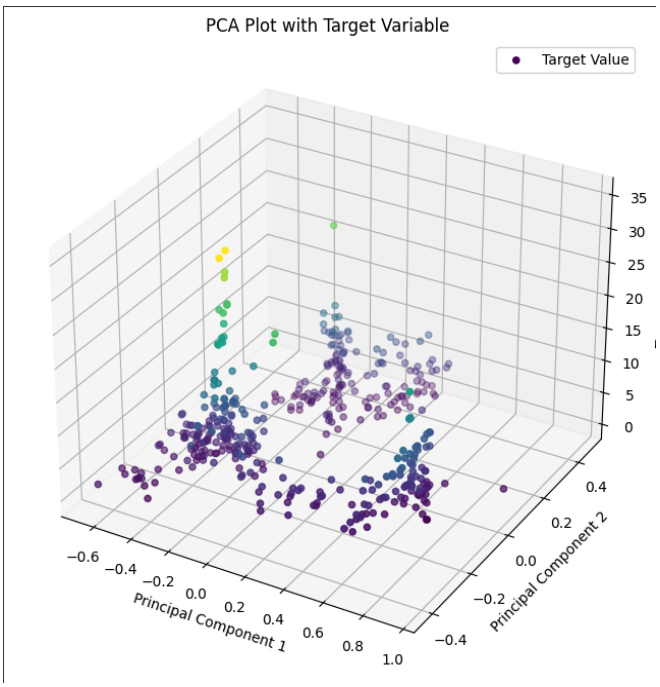


This is a pie graph showing the importance of each input feature towards the target variable, Dmax. Although pie charts are commonly used for categorical data, we can adapt them to represent the relative importance of input features in a predictive model. Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable. These scores help us understand which features significantly contribute to the model's predictions. A pie chart provides a clear and intuitive representation of how much each feature contributes to the overall prediction. By dividing a circle into slices, we visually grasp the relative importance of different features.



The given plot is the multiclass distribution of target variable Dmax, which is a discrete variable, with respect to around 39 distinct classes. Histograms are commonly used for continuous data; we can adapt them to represent the distribution of different classes within a categorical target variable. A multiclass distribution histogram shows the frequency or count of each class within the target variable. Each class corresponds to a bar in the histogram. Since the number of classes is large, we have used a grouped histogram chart

Principal Component Analysis (PCA)

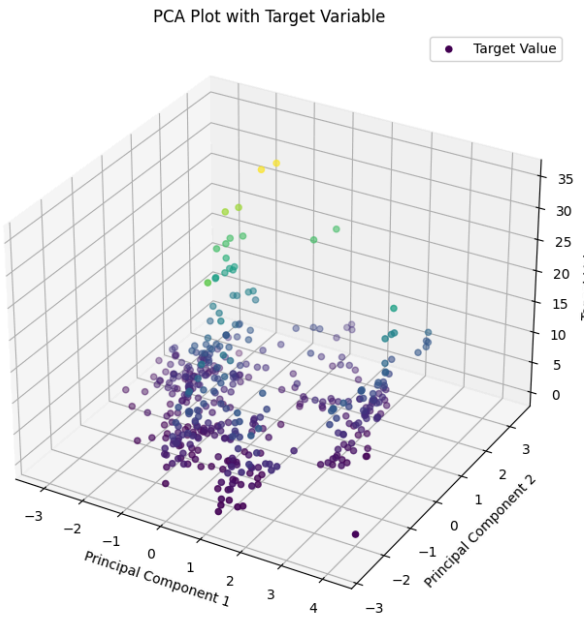


This method combines the seven features into two, and shows its relation with the target variable. As the number of features (dimensions) in a dataset increases, the amount of data required for meaningful results grows exponentially. This phenomenon is known as the curse of dimensionality. PCA helps address this issue by transforming correlated variables into a set of uncorrelated principal components. By retaining most of the original information, PCA reduces the dimensionality while preserving essential patterns. In predictive modelling, PCA simplifies the dataset without sacrificing predictive power. It improves model efficiency, reduces overfitting, and speeds up training. PCA is especially useful when dealing with high-dimensional data patterns. PCA is an unsupervised learning algorithm. It doesn't require prior knowledge of target variables. By focusing on variance, it captures essential information without bias. PCA aids in feature engineering by creating new features (principal components) that are linear combinations of the original ones. These components are orthogonal, making them easier to interpret.

Data Preprocessing

Data preprocessing is a critical step in preparing raw data for analysis and machine learning. Real-world data often arrives in inconsistent formats with varying structures. Human errors during data collection can introduce inaccuracies. Data preprocessing resolves these issues, making the dataset more consistent and reliable. Noisy data includes outliers, mislabels, and meaningless information. These outliers can distort model training. Data preprocessing removes noise, improving model robustness. Clean data leads to better model performance. Techniques like cleaning, normalisation, handling missing values, and outlier treatment improve data integrity. Accurate data facilitates better decision-making.

In our code, we have used **min-max scaling** normalisation. Data normalisation is a vital preprocessing technique used to transform features in a dataset to a common scale. The main goal of normalisation is to eliminate biases and distortions caused by different scales of features. By bringing all values to a similar range, normalisation ensures that each feature contributes equally during model training.

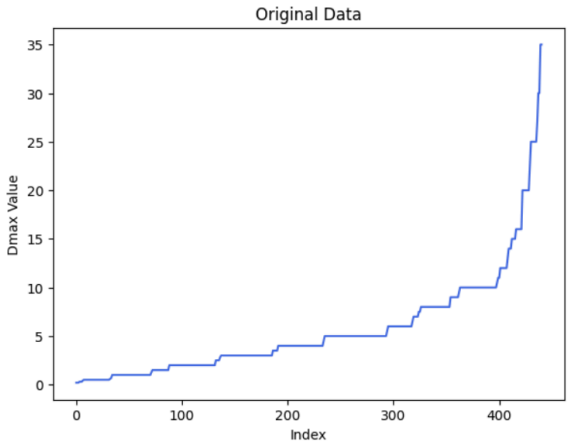


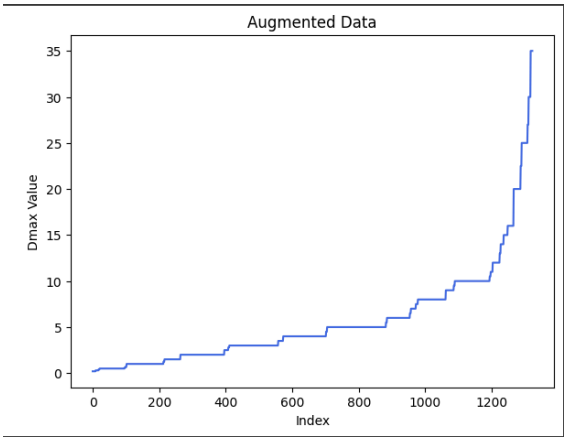
The plot alongside shows a 3-dimensional idea of the target variable with respect to 2 principal components. This plot has been obtained after normalisation of the input variables. If compared with the above PCA plot, which is obtained before normalisation, the effect of normalisation can easily be seen in the dataset

It may seem that this data contains outliers, but in our project, we have not removed our outlier, because of the following reasons:

- Certain values of Dmax have only single values, so upon outlier removal, those data may get removed from dataset, skewing our analysis
- The dataset does not have any experimental or measurement errors, so outlier removal does not have any effect in this dataset

Data Augmentation



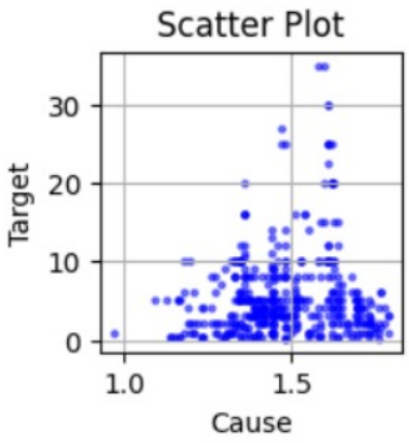


The following two plots given above show the before and after of applying data augmentation on the small given dataset. The data augmentation is a major step in this project. It was observed using various models that the MSE was coming up to be quite high. This could be attributed mainly to the overfitting of these models on the insufficient dataset. Augmenting the data introduces variability, making the model more robust and better at generalising. By artificially creating new training examples, data augmentation effectively increases the dataset size. More data allows the model to learn better representations and capture underlying patterns. We achieved an efficient augmented dataset using a few steps, keeping in mind the type of data, and the models that can be used.

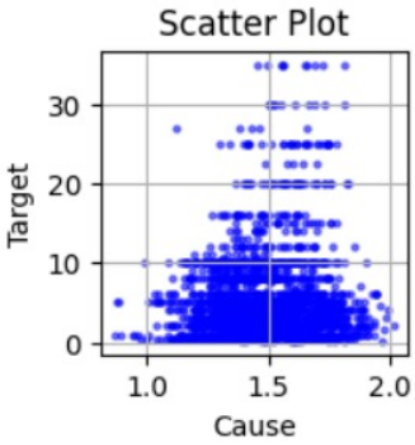
Adding noise to data is one approach to data augmentation. We add Gaussian noise with mean 0 and standard deviation 0.1 to each feature of the original sample to create new augmented samples. The target values remain unchanged.

Another method after noise augmentation is Smote. **SMOTE** stands for Synthetic Minority Over-sampling Technique. It is an oversampling technique used to balance the class distribution of a dataset by creating synthetic minority class samples. In imbalanced classification, there are too few examples of the minority class for a model to effectively learn the decision boundary. Like in the dataset given, values for higher Dmax value is very low, which causes inaccuracy while training the model. Simply duplicating examples from the minority class in the training dataset can balance the class distribution but doesn't provide additional information to the model. **SMOTE** improves upon duplication by synthesising new examples from the minority class. It is a type of data augmentation for tabular data and can be very effective. **SMOTE** creates a more balanced dataset, and allows the model to learn from synthetic examples, leading to more accuracy.

Plot of a feature with respect to target variable before & after data augmentation:



Before Augmentation



After Augmentation

Why Data Augmentation?

Small Dataset: Original dataset is very small so it becomes very difficult for any model to grasp the complex relationship between features and target. We are getting a very high MSE which is very undesirable. So we artificially increase the dataset to get better results.
Discontinuous Values: For higher values of Dmax, the values are very discontinuous. They may seem as outliers but by manual observation we can conclude they are not outliers.
Skewed Data: There are very few data points for higher values of Dmax which makes uniform distribution of data. This makes it difficult for the model to give efficient results. So we resample the data using SMOTE.

```
print(f"Mean Squared Error on Train : {mse_train} ")
print(f"Mean Squared Error on Test : {mse_test} ")
```

```
Mean Squared Error on Train : 5.4583263476
Mean Squared Error on Test : 26.1548621468
```

Overfitting: Due to the small dataset, we can see that model is overfitting, that is, it is not recognizing patterns of the data but memorising the data.

Model : Regression

Regression analysis is a widely used statistical technique that aims to model the relationship between one or more predictor variables and a response variable. The regression model matches closely with the problem statement, where the input variables like total electronegativity (TEN), atomic size difference (d), average atomic volume (VA), mixing entropy (Sm), glass-transition temperature (Tg), onset crystallisation-temperature (Tx), and liquidus temperature (Tl) influence the glass-forming ability (GFA) of metallic glass,which is specified in the material's critical casting diameter (Dmax).

Let's explore different types of regression models and their use cases:

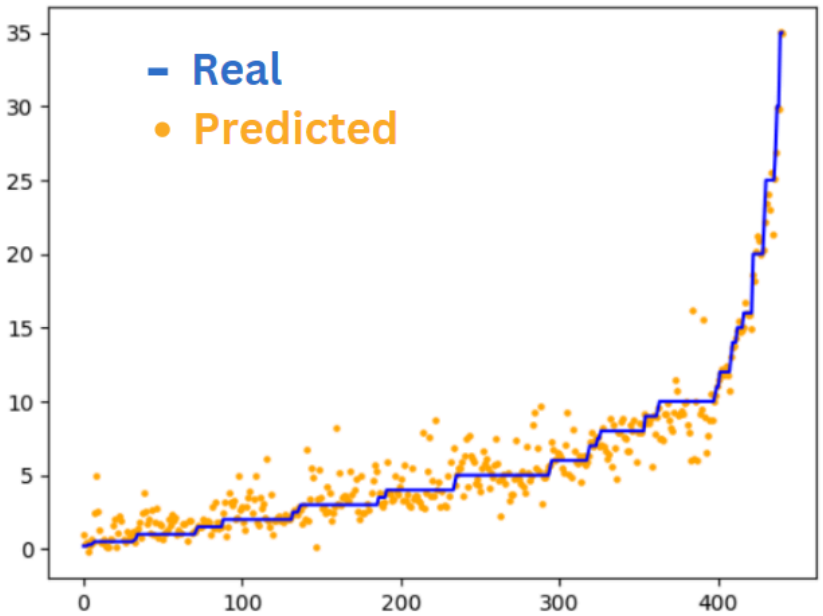
1. Linear Regression:
    - Use Case: When the relationship between the predictor variable(s) and the response variable is reasonably linear and the response variable is continuous (numeric).
2. Polynomial Regression:
    - Use Case: When the relationship between predictor variable(s) and the response variable is non-linear.
3. Ridge Regression:
    - Use Case: When predictor variables are highly correlated, and multicollinearity becomes a problem.
4. SVR (Support Vector Regression):
    - SVR is a type of machine learning algorithm used for regression analysis.
    - It aims to find a function that approximates the relationship between input variables and a continuous target variable while minimising prediction error.
5. ANN (Artificial Neural Network):
    - ANN is a computational network inspired by the human brain.
    - It consists of interconnected nodes (neurons) organised in layers and is used for tasks like image classification, natural language processing, and pattern recognition

Finally let us delve into the most important model, which has been used in our project, which is **XGBoost**. **XGBoost** (Extreme Gradient Boosting) is a powerful machine learning algorithm that has gained immense popularity due to its exceptional performance in various domains. Here's a detailed description of XGBoost:

1. Introduction:
    - XGBoost stands for Extreme Gradient Boosting.
    - It is an implementation of gradient boosted decision trees designed for speed, performance, and accuracy.
2. Key Features:
    - Parallel Tree Building:
        - Unlike traditional gradient boosting, XGBoost builds trees in parallel rather than sequentially.
        - This significantly speeds up the training process.
    - Regularisation:
        - XGBoost introduces L1 (Lasso) and L2 (Ridge) regularisation terms to prevent overfitting.
        - Regularisation helps control the complexity of the model.
    - Handling Missing Values:

        - XGBoost can handle missing values internally during tree construction.
        - It splits nodes based on missingness, ensuring robustness.
    - Custom Loss Functions:
        - You can define custom loss functions tailored to specific problems.
        - This flexibility allows optimization for various objectives (e.g., ranking, regression, classification).
    - Tree Pruning:
        - XGBoost uses pruning to remove branches that do not contribute significantly to improving the model.
        - This prevents overfitting and enhances generalisation.
    - Cross-Validation:
        - Built-in cross-validation helps tune hyperparameters effectively.
        - It selects the best model based on validation performance.
3. Use Cases:
    - Structured Data:
        - XGBoost excels in tabular data scenarios (e.g., financial predictions, customer churn, fraud detection).
        - It handles both regression and classification tasks.
    - Ranking and Recommendation Systems:
        - XGBoost can be used for personalised recommendations and search engine ranking.
    - Time Series Forecasting:
        - XGBoost handles temporal dependencies well.
4. Performance and Popularity:
    - XGBoost consistently ranks high in Kaggle competitions.
    - It is widely adopted in industry and research due to its robustness and scalability.

In summary, XGBoost is a versatile and efficient algorithm that combines the power of gradient boosting with regularisation, parallelization, and advanced features, making it a valuable tool for machine learning practitioners.



This plot shows the correspondence of the real and predicted values of the target variable Dmax.

Testing

Testing has been done keeping 20% of the dataset for testing.

Mean Squared Error for 10 times training		
Serial no.	Training MSE	Testing MSE
1	0.262300	0.979264
2	0.263162	0.848903
3	0.334851	0.864587
4	0.261840	0.837816
5	0.229610	1.070218
6	0.297928	0.822306
7	0.261201	0.892058
8	0.261796	1.268137
9	0.281595	0.659459
10	0.363880	1.153578

Predictive performance = Average MSE of the ten performances = 0.978274
Fluctuation in performance = Standard deviation of ten performances = 0.180935

For one of the training set:

1. Training MSE = 0.260214
2. Testing MSE = 0.761704
3. R^2 Score = 0.989941
4. Root Mean Square Error = 0.872757
5. Mean Absolute Error = 0.398819
6. Mean Absolute Percentage Error = 14.499320

	Train set					Test set				
	MSE	RMSE	MAE	MAPE	R2 Score	MSE	RMSE	MAE	MAPE	R2 Score
0	0.325242	0.570300	0.281052	12.363061	0.995860	0.971014	0.985400	0.452339	18.807370	0.987435
1	0.325914	0.570889	0.280248	12.247030	0.995769	0.782781	0.884749	0.423567	19.584313	0.990601
2	0.277191	0.526489	0.258191	10.632884	0.996419	1.151015	1.072854	0.429620	15.586003	0.985924
3	0.283964	0.532883	0.262205	10.848112	0.996327	0.863412	0.929200	0.403254	17.013611	0.989499
4	0.252485	0.502479	0.245185	10.655187	0.996774	1.265778	1.125068	0.446711	18.715560	0.983877
5	0.260214	0.510111	0.256306	11.174609	0.996704	0.761704	0.872757	0.398819	14.499320	0.989941
6	0.254259	0.504241	0.247580	10.604155	0.996756	1.006739	1.003364	0.432181	16.762689	0.987086
7	0.286041	0.534828	0.262496	10.472476	0.996361	0.794834	0.891534	0.392265	15.548265	0.989691
8	0.320173	0.565839	0.285391	12.441978	0.995947	0.985231	0.992588	0.436168	18.754417	0.986928
9	0.362345	0.601951	0.301905	12.915238	0.995367	1.200229	1.095550	0.443859	17.199800	0.984748

	Train set				Test set			
	MSE	RMSE	MAE	R2 Score	MSE	RMSE	MAE	R2 Score
count	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000
mean	0.259447	0.509084	0.249912	0.996690	0.845414	0.915892	0.395320	0.989176
std	0.017863	0.017642	0.007821	0.000218	0.158720	0.085342	0.016837	0.001960
min	0.232546	0.482230	0.238569	0.996413	0.633381	0.795853	0.375678	0.985236
25%	0.244026	0.493972	0.244505	0.996509	0.713311	0.844575	0.382834	0.988485
50%	0.263225	0.513054	0.250400	0.996650	0.855016	0.924667	0.392552	0.989286
75%	0.274138	0.523581	0.256132	0.996901	0.908497	0.953142	0.404396	0.990850
max	0.280820	0.529924	0.259822	0.996987	1.142738	1.068989	0.425347	0.991285

Metrics used

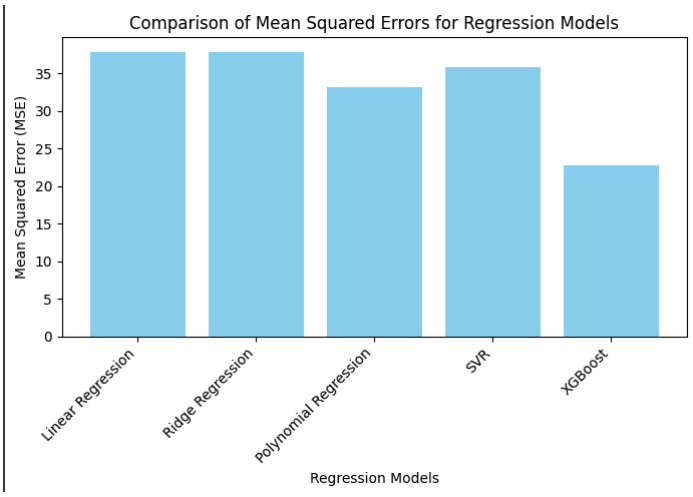
R^2 score : The coefficient of determination, often denoted as R^2, measures how well a statistical model predicts an outcome. R^2 is a number between 0 and 1. It quantifies how well a model explains the variance in the dependent variable (outcome). The higher the R^2, the better the model's predictions align with the actual data.

Mean Squared Error (MSE) is a machine learning metric commonly used for evaluating regression models. MSE quantifies the average squared difference between the predicted values and the actual values in a dataset. It measures how well the model's predictions align with the ground truth.

We have used MSE as the major metric due to various reasons:

1. In MSE, the errors are squared. Squaring the errors highlights large deviations, making it useful when minimising occasional large errors is crucial.
2. MSE is straightforward to compute in Python or any other programming language.
3. It provides a single numerical value that represents the overall error of the predictions.

MSE is commonly used for evaluating regression models.The closer the MSE value is to 0, the more accurate the model.In summary, MSE serves as a valuable metric for assessing regression model performance, but its interpretation and comparison require careful consideration



The adjoining plot is a comparison of the MSE values for all of the relevant regression models that were ideal for the given dataset. From this plot, it is clearly evident that XGBoost emerges as the go-to model for the given dataset.

Mean Absolute Percentage Error (MAPE), a commonly used metric for evaluating prediction accuracy in regression and forecasting models. MAPE quantifies the average percentage deviation between predicted values and actual (observed) values. It measures the average percentage difference between predictions and actuals. A lower MAPE indicates better model performance.

1. Advantages of MAPE:
    - Relative Metric: MAPE is independent of the scale of the data.
    - Easy Interpretation: Stakeholders can understand percentage errors.
    - Robust to Outliers: Large errors are not disproportionately penalised.
2. Limitations of MAPE:
    - Division by Zero: If actual values ((O_i)) are zero, MAPE becomes undefined.
    - Sensitivity to Extreme Values: Outliers can significantly impact MAPE.
    - Asymmetry: MAPE treats overestimation and underestimation equally.

Root Mean Square Error (RMSE), a commonly used metric for evaluating prediction accuracy in regression and forecasting models. The Root Mean Square Error (RMSE) is a statistical measure that quantifies the average deviation between predicted values and actual (observed) values. A lower RMSE indicates better model performance.

1. Advantages of RMSE:
    - Sensitive to Large Errors: RMSE penalises large errors more significantly due to the squaring operation. This is useful when outliers or extreme values need attention.
2. Limitations of RMSE:
    - Scale Dependency: RMSE is sensitive to the scale of the data. Comparing RMSE across different datasets may be misleading.
    - Outliers Impact RMSE: Large errors disproportionately affect RMSE. We must be cautious when dealing with extreme values.

Sustainability @ Metallic Glass

The use and manufacturing of metallic glass offer several avenues for sustainability measures:

1. Energy Efficiency in Manufacturing:
   - Metallic glass manufacturing processes, such as melt spinning or splat quenching, often require less energy compared to traditional methods used for crystalline materials like casting and forging. This reduced energy consumption contributes to lower carbon emissions and overall environmental impact.

2. Material Efficiency:
   - Metallic glass fabrication typically involves minimal material waste compared to conventional manufacturing processes. The ability to form intricate shapes through processes like injection moulding or additive manufacturing further enhances material efficiency, reducing the need for excess material and minimising scrap generation.

3. Lightweight Design:
   - Metallic glasses possess exceptional strength and hardness while being significantly lighter than traditional metallic alloys. By incorporating metallic glasses into various applications, such as aerospace components or lightweight structural elements in automotive vehicles, manufacturers can achieve weight reduction, leading to improved fuel efficiency and reduced greenhouse gas emissions.

4. Durability and Longevity:
   - The unique properties of metallic glasses, including high strength, corrosion resistance, and fatigue endurance, contribute to the longevity and durability of products made from these materials. Enhanced durability reduces the frequency of replacements and repairs, thereby lowering resource consumption and waste generation over the product's lifecycle.

5. Recyclability and Circular Economy:
   - Metallic glasses are inherently recyclable due to their homogeneous and amorphous structure. At the end of a product's life, metallic glass components can be easily reclaimed and reintegrated into the manufacturing process through recycling techniques like melting and reprocessing. Embracing a circular economy model promotes resource conservation and minimises the reliance on virgin materials.

6. Improved Performance and Efficiency:
   - Utilising metallic glasses in various applications can lead to improved performance and efficiency, such as in electrical transformers, where reduced energy losses due to magnetic properties of metallic glasses translate to energy savings. Higher efficiency means reduced resource consumption and environmental impact per unit of output.

7. Alternative to Critical Materials:
   - Some metallic glasses can serve as alternatives to conventional materials that rely on scarce or environmentally harmful elements. By substituting critical materials with more abundant and sustainable alternatives, such as metallic glasses, manufacturers can mitigate supply chain risks and promote resource diversification.

By integrating sustainability considerations into the use and manufacturing of metallic glasses, industries can not only benefit from their exceptional properties but also contribute to environmental conservation and resource stewardship. Collaboration among researchers, engineers, policymakers, and industry stakeholders is essential to maximise the sustainability potential of metallic glass technologies.

Appendix

XGBoost

XGBoost stands for Extreme Gradient Boosting. It's a powerful open-source machine learning library known for its efficiency and accuracy, particularly in tasks like:

- **Regression:** Predicting continuous values, like house prices or stock prices.
- **Classification:** Categorizing data points, like spam detection or image recognition.
- **Ranking:** Ordering items based on preference, such as product recommendations or search results.

Here's a breakdown of how XGBoost works:

1. **Ensemble Learning:** It builds a model by combining multiple weaker models, often decision trees, into a single stronger model.
2. **Gradient Boosting:** It iteratively trains these decision trees, with each one focusing on correcting the errors made by the previous ones.
3. **Regularisation:** It incorporates techniques to prevent the model from overfitting to the training data.

XGBoost's advantages include:

- **Scalability:** It can handle large datasets efficiently.
- **Performance:** It often achieves state-of-the-art results in machine learning competitions.
- **Flexibility:** It can be fine-tuned with various parameters for different tasks.
- **Interpretability:** You can understand how the model makes predictions to some extent.

This makes XGBoost a popular tool for data scientists working on various applications.

Principal Component Analysis (PCA)

PCA stands for Principal Component Analysis. It's a dimensionality reduction technique used in machine learning and data analysis. Here's the gist of it:

- **Simplifying Complex Data:** Imagine a dataset with many variables. PCA helps condense this data into a smaller set of variables, capturing the most important information.
- **Focus on Variance:** PCA identifies new variables, called principal components, that represent the directions of greatest variance in the data.
- **Less is More (sometimes):** By keeping the principal components with the most variance, we can often achieve good results with fewer variables, making data analysis and visualisation easier.

Here are some common applications of PCA:

- **Data Visualization:** Reducing dimensions allows for easier visualisation of complex datasets in lower dimensions (e.g., from 3D to 2D).

- **Feature Engineering:** PCA can be used to create new, more informative features for machine learning models.
- **Noise Reduction:** Sometimes PCA can help remove irrelevant noise from data, improving model performance.

However, it's important to remember that PCA discards some information in the process of dimensionality reduction. So, it's crucial to choose the number of components that retains the most important aspects of the data for your specific task.

Exploratory Data Analysis (EDA)

EDA stands for Exploratory Data Analysis. It's a crucial initial step in data science projects that involves investigating and understanding a dataset before diving into more complex modelling or analysis.

Here's a breakdown of what EDA entails:

- **Unveiling Patterns:** EDA uses statistical and visualisation techniques to identify patterns, trends, and relationships within the data. This can involve techniques like calculating summary statistics, creating histograms, scatter plots, and boxplots.
- **Looking for Outliers:** Exploratory analysis helps spot outliers, data points that fall far outside the expected range. These can be genuine insights or indicate errors requiring further investigation.
- **Understanding Relationships:** EDA focuses on how different variables in the data connect with each other. This can help refine hypotheses and inform the direction of further analysis.
- **Data Cleaning Check:** Often, EDA uncovers issues with the data, like missing values or inconsistencies. This paves the way for data cleaning tasks to ensure the quality of the data before using it in models.

Overall, EDA is like an initial conversation with your data. It helps you get acquainted with its characteristics, strengths, and weaknesses before attempting to draw conclusions or make predictions.

Data Augmentation

Data augmentation is a technique used specifically to improve the performance of machine learning models, particularly those dealing with limited data. It essentially involves artificially creating new data from existing data in your dataset.

There are two main reasons why data augmentation is useful:

1. **Increased Dataset Size:** Machine learning models often require a lot of data to train effectively. Data augmentation helps address situations where collecting real-world data is expensive, time-consuming, or impractical. By creating new variations of your existing data, you can effectively increase the size and diversity of your training dataset.

2. **Improved Model Generalizability:** Real-world data often comes with variations. For instance, an image of a cat might be taken from different angles, in varying lighting

conditions, or with slight occlusions. Data augmentation helps introduce these variations artificially. By training a model on data that incorporates this variability, the model becomes better at generalising to unseen examples, improving its performance on real-world data.

Here are some common data augmentation techniques:

- **For Images:** Flipping, rotating, cropping, scaling, and adjusting brightness and contrast are all common ways to augment image data.
- **For Text:** Techniques like synonym replacement, word shuffling, and random deletion can be used to create variations of text data.

The effectiveness of data augmentation depends on the specific task and data type. It's important to experiment and find the right balance of techniques to improve your model's performance without introducing noise or unrealistic variations.

Regression

Regression analysis is a powerful statistical technique used to model and understand the relationships between variables. Let's explore the key aspects of regression:

1. What Is Regression Analysis?
    - Definition: Regression analysis estimates the relationships between a dependent variable (also called the response or outcome variable) and one or more independent variables (also known as predictors, covariates, or features).
    - Purpose:
        - Predict future outcomes based on historical data.
        - Understand how changes in independent variables impact the dependent variable.
        - Infer causal relationships (in some cases).

2. Types of Regression Models:
    - Linear Regression:
        - Simplest form of regression.
        - Assumes a linear relationship between the dependent and independent variables.
            - (Y): Dependent variable
            - (X): Independent variable
            - (a): Intercept
            - (b): Slope
            - (\epsilon): Residual (error)
    - Multiple Linear Regression:
        - Extends linear regression to multiple independent variables.
    - Nonlinear Regression:
        - Handles more complex relationships (e.g., exponential, logarithmic).
        - Useful when linear models don't fit the data well.
3. Assumptions of Linear Regression:

    - Linearity: Dependent and independent variables have a linear relationship.
    - Independence: Residuals are independent of each other.
    - Homoscedasticity: Residuals have constant variance.
    - Normality: Residuals follow a normal distribution.

4. Interpreting Regression Coefficients:
    - Intercept ((a)): Predicted value of the dependent variable when all independent variables are zero.
    - Slope ((b)): Change in the dependent variable for a one-unit change in the independent variable.
5. Model Evaluation:
    - Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values.
    - R-squared ((R^2)): Proportion of variance in the dependent variable explained by the model.
    - Adjusted R-squared: Penalises adding unnecessary variables.

In summary, regression analysis is a versatile tool used across various fields to model and predict outcomes based on data patterns. Whether you're analysing financial markets or studying social phenomena, understanding regression is essential

Dataset

A **dataset** is a structured collection of data points grouped together for analysis, modelling, or processing. Let's explore the key aspects of datasets:

1. Definition:
    - A dataset consists of multiple data points organised into a single table or matrix.
    - Each row represents an individual data observation, and each column corresponds to a specific feature or variable.
    - Datasets are used across various fields, including machine learning, business, and government.
2. Features of a Dataset:
    - Rows (Data Points):
        - Represent individual instances or observations.
        - Examples: Customers, patients, images, transactions.
    - Columns (Features):
        - Represent attributes or characteristics of the data.
        - Examples: Age, temperature, income, product category.
    - Data Types:
        - Datasets can contain numerical, categorical, or time-related data.
        - Numerical data includes measurements (e.g., temperature, weight).
        - Categorical data includes labels or categories (e.g., colours, gender).
        - Time-related data includes timestamps or dates.
3. Creating a Dataset:
    - Datasets can be stored in various formats:
        - CSV (Comma-Separated Values)
        - Excel

        - JSON
        - ZIP files (for large datasets like images)

In summary, datasets serve as the foundation for various operations, techniques, and models. They allow developers to gain insights, make informed decisions, and train algorithms

Heatmap

A heatmap is a 2-dimensional data visualisation technique that represents the magnitude of individual values within a dataset using colours. Let's explore the key aspects of heatmaps:

1. Representation:
    - A heatmap displays data as a grid of coloured cells, where each cell corresponds to a specific data point.
    - The variation in colour intensity or hue reflects the value associated with that data point.
2. Colour Mapping:
    - Heatmaps use a colour scale to represent the magnitude of values.
    - Common colour scales:
        - Sequential: Shades of a single colour (e.g., from light to dark blue).
        - Diverging: Two contrasting colours (e.g., blue to red) for positive and negative values.
        - Categorical: Different colours for discrete categories.
3. Interpretation:
    - Darker colours indicate higher values.
    - Lighter colours represent lower values.
    - Heatmaps help identify trends, clusters, and anomalies.

In summary, heatmaps provide an effective way to visualise patterns, correlations, and distributions within datasets using colour gradients.

Normalisation

Normalisation is the process of organising a database to reduce redundancy and improve data integrity. It is commonly used in relational database design to achieve an optimal structure composed of atomic elements (i.e., elements that cannot be broken down into smaller parts). Let's explore the key aspects of normalisation:

1. Goals of Normalisation:
    - Minimise Data Redundancy:
        - Redundant data wastes storage space and complicates maintenance.
    - Avoid Data Anomalies:
        - Anomalies include insertion, update, and deletion issues.
    - Simplify Queries and Updates:
        - Normalised data allows efficient querying and modification.
2. Benefits of Normalisation:
    - Minimises Data Redundancy:
        - Avoids storing the same information in multiple places.
    - Minimises Null Values:
        - Each table contains only relevant data.

- Simplifies Data Maintenance:
    - Fewer anomalies during insertions, updates, and deletions.
- Improves Query Performance:
    - Well-structured tables lead to efficient queries.

In summary, normalisation ensures efficient, well-organised databases by reducing redundancy and maintaining data integrity