1. Introduction

We are doing a classification task by trying to predict which mathematical symbol is drawn on an image using the HASYv2 dataset. We will use three algorithms for this task: one clustering algorithm: K-means, two linear algorithms: Logistic Regression and Support Vector Machines.

For classification metrics, we use accuracy, F1-Score and runtime analysis. Indeed, our 10 classes are a bit imbalanced. This is why we want to carefully handle the class imbalance using the F1-Score metric. The accuracy metric, on the other hand, may be biased towards the majority class with more instances.

2. Method:

- 2.1 Data preparation:

We used 2 types of data preparation:

1) Data normalization: Z-score normalization is used to reduce the effect of differences in feature scales on the outcome. This can lead to better model convergence and faster training times.

2) Bias term: adding a bias term is essential to improve the accuracy and effectiveness of Logistic Regression and SVM models. It allows the linear equation to fit the data better and prevents overfitting by adding flexibility to the model.

- 2.2 Cross Validation

We are using a 1-Fold Cross Validation method in which the raw data is separated as follows: 80% is used for training and 20% for testing. From these 80%, we have dedicated 20% to validation and the rest is for fitting the model.

- 2.3 K-means

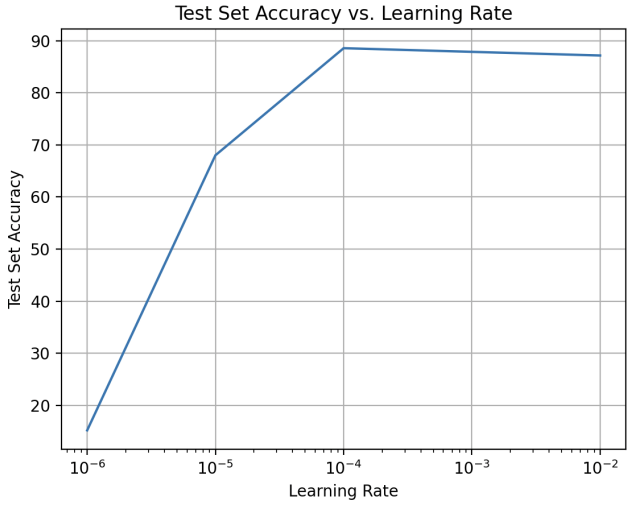
The K-means algorithm utilizes the hyperparameter "k" to determine the desired number of clusters, which impacts both the model's complexity and variability. Choosing a smaller value of "k" leads to a simpler and more stable model, whereas selecting a larger "k" produces a more complex but less biased model. To achieve a balance between these factors, cross-validation is employed to select a suitable value of "k." Through experimentation with hyperparameters ranging from 1 to 60, we determined that the optimal number of clusters was **k = 35**, resulting in an 82.486% validation accuracy.

Regarding the *max_iters* parameter, optimal solutions may not be achieved if a small value is used since the fitting process may not have converged. It is preferable to use a large value instead because the fitting process stops either when the maximum number of iterations is reached or when the stop criterion is met.

- 2.4 Logistic Regression

In logistic regression, we are able to optimize the learning rate and the number of iterations. We opted to show the optimization of the gradient descent learning rate, which is more meaningful than the number of iterations. It is sufficient to say that the accuracy compared to the number of iterations stabilizes at 50 iterations.

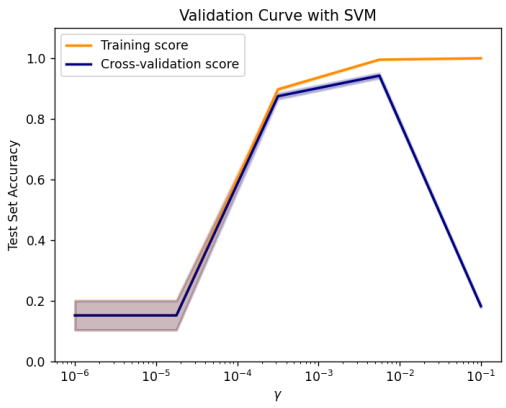
We clearly show in the graph below that the accuracy on the test set is the best when the learning rate is close to **lr = 1e-4**.



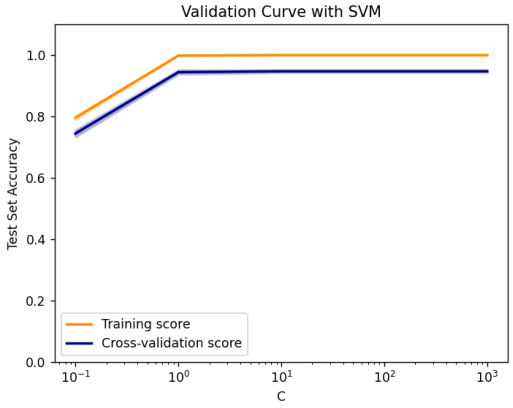
- 2.5 SVM:

In SVM when data is not linearly separable we are using the kernel trick which is a transformation of data to higher dimensional space.

We were testing three kernels: linear, rbf and polynomial. All of them give us the accuracy above 90%, so we can assume that the results are satisfactory, however the best we got was for the **rbf kernel**, with values of parameters: **C = 9.** and **γ = 0.006**. The accuracy was equal to **96.42%**



We can conclude that the best value is γ = **0.006** because at this point the cross-validation score reaches its maximum and then decreases. On the other hand, training accuracy keeps increasing because of overfitting.



C metaparameter affects the size of margin, because they are correlated negatively. For C values bigger than 1, the accuracy stops increasing. That is the result of increasing the C while the smallest possible margin is already achieved.

3. Experiment/Results:

After performing all the models with their optimal hyperparameters on the test dataset, the resulting outcomes were obtained:

Model	K-means	Logistic Regression	SVM
Accuracy	82.48%	90.19%	96.42%
F1-score	0.8235	0.8990	0.9623
Runtime Analysis (Training/ Testing)	2.85s/ 0.02s	0.779s/ 0.001s	2.763s/ 0.495s

K-means has the maximum runtime for training due to its iterative process of assigning data points to centroids.
SVM gives us the best accuracy, but the prediction time is the biggest among all of the classifiers. The time needed for SVM to train is arbitrary, because it depends on the kernel we use (linear needs less time to perform, when rbf is the most complex from those we were using) as well as values of parameters, greater C make our computations slower.

4. Discussion and Conclusion:

During our experiments we analyzed runtime execution. However, the actual runtime of these algorithms can vary depending on several factors, including the size and complexity of the dataset, the implementation of the algorithm, and the hyperparameters used.

SVM:
It is of much importance to tune hyperparameters as the classification of SVM is highly based on chosen values. Kernels like polynomial and rbf are useful in terms of non-linear hyperplanes, which are in higher dimensions. Penalty terms could help us achieve an accurate tradeoff between margin and accuracy. Also different values of gamma make a big difference whereas low gamma's use only nearby points to calculate the separation line, bigger gamma's take into account more samples.
SVM works well on data which can be clearly separated, also in high dimension, but its training is time costly. Because of that we should not apply it to large datasets. Also choosing a properly working kernel can be problematic and makes this algorithm harder to acquire for novices.

K-means:
We observed that during the validation, the larger K the greater the accuracy on the test set. In other words, overfitting did not really happen with large K. Regarding that, we considered the complexity of the model to choose K because a higher K means more clusters and for each test point we need to compute more distances (to each cluster center). It showed that increasing K after 35 does not bring a lot of improvement, so we settled for a smaller one to have a simpler model. There is a trade-off between accuracy and computation complexity.

Logistic Regression:
We added a slight modification to the algorithm when doing the softmax function. In fact, we opted to subtract the maximum score for numerical stability. This has helped us to create more accurate data.

We conclude that even if SVM appears to be the most accurate, Logistic Regression has the best tradeoff between accuracy and runtime. K-mean is far behind, which suggests that the data is more easily represented in higher-order dimensionality as linearly separable.