CSC2062 AIDA – Assignment 1

[Riain Walsh]

[13/03/2025]

Introduction

This report focuses on the development of a machine learning system to classify handwritten lowercase letters (a-j) and three non-letter symbols: a smiley face, a sad face, and an exclamation mark. The goal is to build a robust system capable of accurately identifying these characters through dataset creation, feature engineering, statistical analysis, and machine learning techniques.

The process begins with the creation of a dataset, where handwritten symbols are generated, resized to 18×18 pixels, binarized, and stored in .csv format. Next, 16 numerical features are extracted. Statistical analysis is then conducted to identify key discriminatory features using hypothesis testing and correlation analysis. Finally, machine learning models, such as logistic regression, are implemented and evaluated to classify the symbols effectively. This comprehensive approach ensures the development of a reliable system for classifying handwritten letters and symbols, with insights gained from each stage guiding further optimization and refinement.

Section 1

When creating the dataset I used the computer mouse and the GIMP application, the symbols were drawn on an 18x18 point space, using 1 pixel pencil tool. The dataset consists of 140 images, including 8 images each for the letters {a-j}, 20 for a happy face, 20 for a sad face, and 20 for an exclamation mark. These images were exported as PGM(Portable Gray Map), type ASCII. Then I wrote the code to convert these PGM files to csv files, The code converts **PGM (Portable Gray Map) image files** containing **18x18 grayscale images** into **CSV (Comma-Separated Values) files** containing binary matrices of **0s (white)** and **1s (black)** based on a **threshold value of 128 (**any value less than 128 is a black pixel and any value greater then 128 is a white pixel) It first reads the PGM file, extracts the pixel values (ignoring the header), reshapes them into an **18x18 NumPy array**, and applies a threshold to convert the grayscale image into a binary image. The binary matrix is then saved as a CSV file without headers. The script also automatically renames the output CSV files using a provided **student number, label, and index** extracted from the filenames.

Section 2

In this section, I developed Python code to extract 16 numerical features from each 18x18 binary image matrix generated in Section 1. These features were designed to capture structural and spatial characteristics of the symbols, providing measurable differences to aid in distinguishing between letters and non-letters during later analysis. The extracted features were saved in a CSV file named STUDENTN_features.csv, where STUDENTN corresponds to my student number. The file contains 141 rows: one header row with feature names and 140 rows of feature values, corresponding to each image, sorted alphabetically by label and numerically by index.

Feature Calculation

The following outlines the logic and implementation for each feature:

- Number of Black Pixels (nr_pix): Sums all black pixels using np.sum(), capturing ink density. Higher values indicate thicker or more filled-in symbols.
- Rows with Exactly One Black Pixel (rows_with_1): Counts rows with one black pixel using np.sum(), identifying narrow horizontal strokes or isolated marks.
- Columns with Exactly One Black Pixel (cols_with_1): Counts columns with one black pixel, capturing isolated vertical strokes.
- Rows with Three or More Black Pixels (rows_with_3p): Counts rows with three or more black pixels, identifying thick horizontal segments.
- Columns with Three or More Black Pixels (cols_with_3p): Counts columns with three or more black pixels, detecting thick vertical segments.
- Aspect Ratio (aspect_ratio): Calculates the width-to-height ratio of the bounding box around black pixels using np.argwhere(), indicating whether symbols are tall/narrow or wide/short.
- Black Pixels with One Neighbor (neigh_1): Counts black pixels with exactly one black neighbor in an 8-connected neighborhood, capturing endpoints or spikes.
- No Black Neighbors Above (no_neigh_above): Counts black pixels with no neighbors above, identifying floating components or upper boundaries.
- No Black Neighbors Below (no_neigh_below): Counts black pixels with no neighbors below, capturing lower boundaries or hanging elements.
- No Black Neighbors to the Left (no_neigh_left): Counts black pixels with no left neighbors, identifying left-side boundaries or isolated left strokes.
- No Black Neighbors to the Right (no_neigh_right): Counts black pixels with no right neighbors, capturing right-side boundaries or rightmost strokes.
- No Horizontal Neighbors (no_neigh_horiz): Counts pixels with no left or right neighbors, identifying isolated vertical strokes or thin lines.
- No Vertical Neighbors (no_neigh_vert): Counts pixels with no top or bottom neighbors, capturing thin horizontal lines or floating segments.
- Connected Areas (connected_areas): Uses scipy.ndimage.label() to count connected black pixel components, identifying unified shapes or disjointed parts.
- Eyes (eyes): Counts enclosed white spaces by inverting the image and applying connected component labeling, identifying closed loops.
- Vertical Symmetry (custom): Measures vertical symmetry by comparing left and right halves of the image using NumPy. Higher similarity indicates symmetrical symbols.

The resulting CSV file contains 140 rows of feature data, where each row begins with the image label and index, followed by the 16 numerical feature values.

One key challenge I encountered was accurately counting connected components and enclosed white spaces (eyes). To resolve this, I utilized scipy's ndimage.label() function, which efficiently identifies clusters of connected pixels. Additionally, handling image boundaries while counting neighbours required careful indexing, which I addressed by padding the image with white pixels to simplify neighbour counting.

Ensuring consistent sorting in the output CSV was another consideration. I resolved this by explicitly sorting the rows by label and index prior to saving the file, ensuring the data was well-structured for the subsequent statistical analysis.

Section 3

In this section, we conduct statistical analyses to explore which extracted features are most effective in distinguishing between different types of handwritten symbols. By applying descriptive statistics, hypothesis testing, and visualizations, we aim to uncover patterns and relationships within the feature data that contribute to classification.

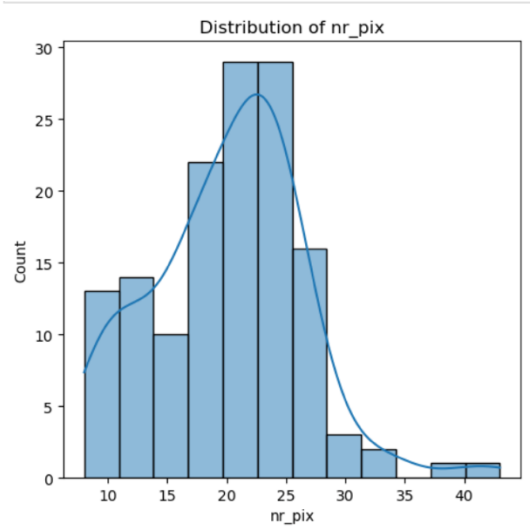
First, we analyze the distribution of the first six features using histograms, identifying key patterns and potential outliers. Next, we compute summary statistics for all features, comparing their distributions across letters and non-letters. This provides an initial insight into which features may hold the most discriminative power.

To quantify these differences, we employ statistical hypothesis testing to assess whether certain features significantly separate letters and non-letters. We also investigate linear associations between features, using correlation measures to determine whether some features provide redundant information or are strongly related.

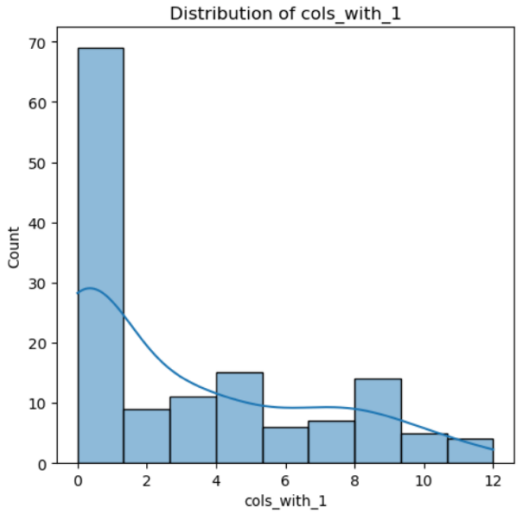
Throughout this section, appropriate statistical methods will be chosen and justified based on the data characteristics. The findings will be supported with tables, graphs, and discussions to ensure clarity.

Section 3.1

The histograms below show the distribution of the first six features across the full dataset:

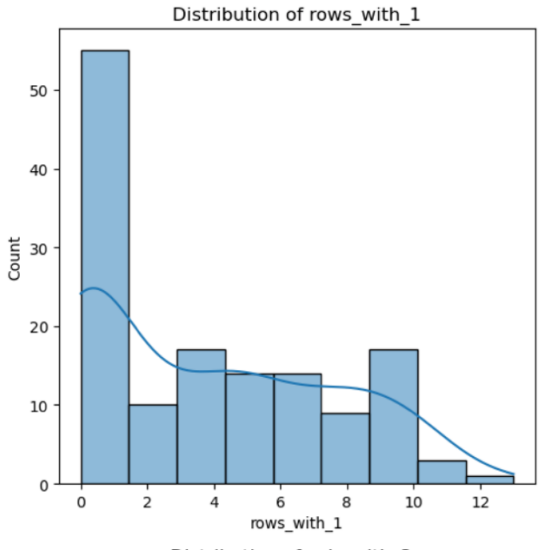


The histogram of pixel count is unimodal and slightly skewed to the right. The majority of symbols have pixel counts in the 20-25 range. There is a long tail to the right, indicating that some symbols have a much higher pixel count. The distribution is centered around 20-25 pixels, which is the peak of the distribution.

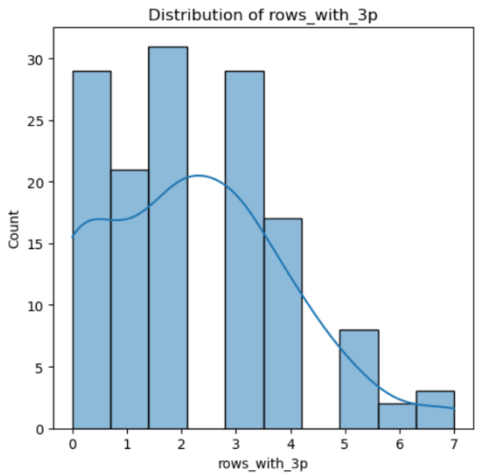


The histogram of the distribution of the number of columns with pixels is multimodal, meaning that it has multiple peaks. This indicates that there are several distinct groups of symbols in the dataset, each with a

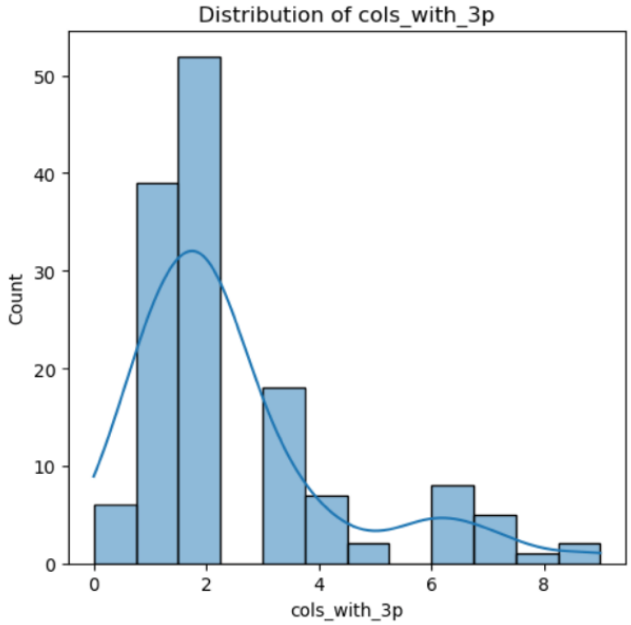
different number of columns with pixels. The tallest peak is at zero, indicating that the most common number of columns with pixels is zero. However, there are several other smaller peaks at higher values, indicating that there are other groups of symbols with more columns with pixels. The graph is spread out, indicating that there is a wide range of values for the number of columns with pixels in the dataset.



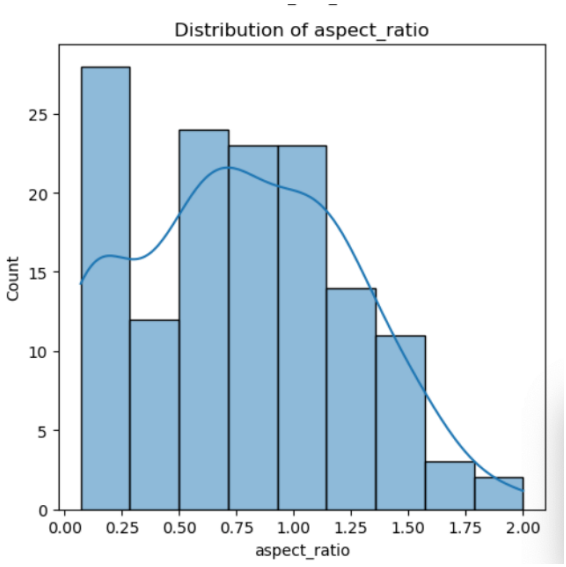
The histogram illustrates a multimodal distribution of the number of rows containing pixels. A prominent peak appears near zero, indicating a significant portion of symbols have very few rows with pixels. Smaller peaks are observed at higher values, suggesting other groups of symbols with more rows containing pixels. This distribution implies distinct categories of symbol heights within the dataset. The spread of the data suggests variability in the vertical dimension of the handwritten symbols. Overall, the graph indicates that the number of rows with pixels is a distinguishing feature, with the majority of symbols having very few rows with pixels.



The histogram displays a multimodal distribution for the number of rows with at least three pixels. There are multiple distinct peaks, suggesting different categories of symbols based on their vertical pixel density. The distribution is fairly spread out, indicating a range of values for this feature. The peaks are relatively even, suggesting that there isn't one dominant category. This feature, "rows_with_3p", appears to be useful for distinguishing between different types of handwritten symbols based on their vertical pixel density. Overall, the graph indicates that the number of rows with at least three pixels is a distinguishing feature, with various groups of symbols exhibiting different densities.



The histogram illustrates a multimodal distribution for the number of columns with at least three pixels. A dominant peak is present around 2, suggesting a significant portion of symbols have a specific width characteristic. Smaller peaks at higher values indicate other symbol categories with wider pixel densities. The spread of the data suggests variability in the horizontal pixel concentration. The feature "cols_with_3p" is useful for distinguishing between symbols based on their horizontal pixel density. Overall, the graph indicates that the number of columns with at least three pixels is a distinguishing feature, with various groups of symbols exhibiting different horizontal densities.



The histogram shows a multimodal distribution of the aspect ratio (width/height) of the handwritten symbols. A significant peak appears near 0.5, indicating many symbols have a width roughly half their height. Smaller peaks at higher values suggest other groups of symbols with wider widths relative to their heights. The data spans a range of aspect ratios, indicating variability in symbol shapes. Overall, the aspect ratio is a distinguishing feature, with various groups of symbols exhibiting different width-to-height proportions.

Some key observations are:

- The distributions for nr_pix, cols_with_1, and aspect_ratio show notable variation, suggesting they may capture meaningful differences across instances.
- Features like rows_with_1 and cols_with_1 are highly skewed, indicating that many instances have a low pixel spread in either direction, while a few have more extended shapes.

- The varying density observed in rows_with_3p and cols_with_3p indicates that compact and dense regions of pixels are less common, potentially making these features useful for distinguishing different types of patterns in the data.

Section 3.2

In this section, I performed statistical analysis on the extracted features to identify differences between letters and non-letters. I calculated key summary statistics, including mean, median, and standard deviation, for each feature across both groups. The goal of this analysis was to identify features that may provide useful discrimination between letters and non-letters. The summary statistics for each group are presented below:

Summary Statistics for Letters:

```
Summary statistics for letters:
           nr_pix  rows_with_1  cols_with_1  rows_with_3p  cols_with_3p  \
mean    21.762500     5.112500     1.625000      2.687500      2.975000
median  23.000000     5.000000     1.000000      3.000000      2.000000
std      5.726341     2.972421     1.823719      1.665691      2.068296

        aspect_ratio   neigh_1  no_neigh_above  no_neigh_below  no_neigh_left  \
mean        0.697890  1.800000        6.900000        7.200000       8.562500
median      0.700000  1.500000        6.000000        7.000000       9.000000
std         0.326133  1.036059        3.612916        3.688504       3.426789

        no_neigh_right  no_neigh_horiz  no_neigh_vert  connected_areas  \
mean          8.700000       10.187500       9.275000         1.200000
median        9.000000       10.000000       9.500000         1.000000
std           3.227257        3.479028       5.312929         0.402524

           eyes    custom
mean    0.02500  0.899460
median  0.00000  0.895062
std     0.15711  0.030624
```

The summary statistics for the letter dataset provide a quantitative overview of the extracted features. The mean and median values offer insights into the central tendencies of each feature, while the standard deviation indicates the spread or variability within the data. For instance, the 'nr_pix' (number of pixels) has a mean of approximately 21.76 and a median of 23, suggesting a slightly skewed distribution. Features like 'cols_with_1' and 'rows_with_1' have lower medians compared to their means, indicating potential right skewness. The 'aspect_ratio' shows a mean and median close to 0.7, suggesting a common shape characteristic among the letters. The 'neigh' features, representing neighboring pixel counts, display varying means and medians, reflecting the structural diversity of the letters. Overall, these statistics provide a foundation for understanding the distribution and characteristics of the features used for classifying the handwritten letters.

Summary Statistics for Non-Letters:

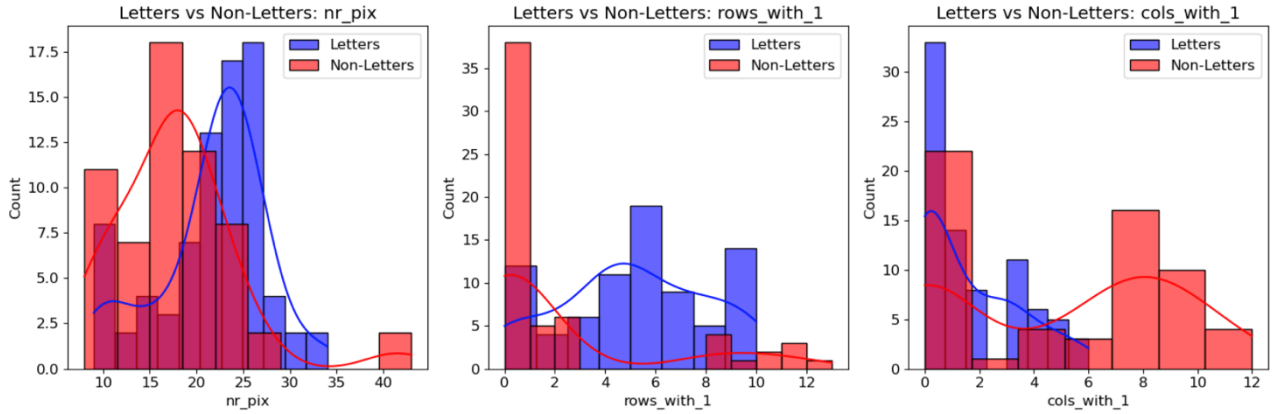
```
Summary statistics for non-letters:
           nr_pix  rows_with_1  cols_with_1  rows_with_3p  cols_with_3p  \
mean    17.733333     2.066667     4.983333      1.600000      1.766667
median  17.000000     0.000000     6.500000      1.500000      2.000000
std      6.573874     3.777124     4.216560      1.575232      1.280448

        aspect_ratio   neigh_1  no_neigh_above  no_neigh_below  no_neigh_left  \
mean        0.872029  1.833333        7.166667        7.133333       7.666667
median      1.095455  2.000000        8.000000        7.000000       7.000000
std         0.597099  1.497644        3.692258        3.784297       2.659595

        no_neigh_right  no_neigh_horiz  no_neigh_vert  connected_areas  eyes  \
mean          7.716667        6.783333       6.900000         2.666667   0.0
median        7.000000        8.000000       9.000000         3.000000   0.0
std           2.643134        3.719357       5.414043         0.475383   0.0

          custom
mean    0.920576
median  0.919753
std     0.034890
```

The summary statistics for the non-letter symbols (smiley faces, sad faces, and exclamation marks) reveal distinct differences compared to the letter dataset. The 'nr_pix' (number of pixels) shows a lower mean and median, indicating that non-letters generally have fewer pixels than letters. The 'cols_with_1' feature exhibits a higher mean and median, suggesting that non-letters tend to have more columns with pixels. The 'aspect_ratio' has a higher mean and median, indicating that non-letters are generally wider relative to their height. The 'neigh' features, representing neighboring pixel counts, also show variations, reflecting the structural differences between letters and non-letters. Notably, the 'eyes' feature has a mean and median of 0.0, suggesting that this feature is not applicable to the non-letter symbols. Overall, these statistics highlight the unique characteristics of the non-letter symbols and provide a basis for distinguishing them from letters.



The three histograms compare the distributions of 'nr_pix' (number of pixels), 'rows_with_1' (number of rows with at least one pixel), and 'cols_with_1' (number of columns with at least one pixel) between letters and non-letters.

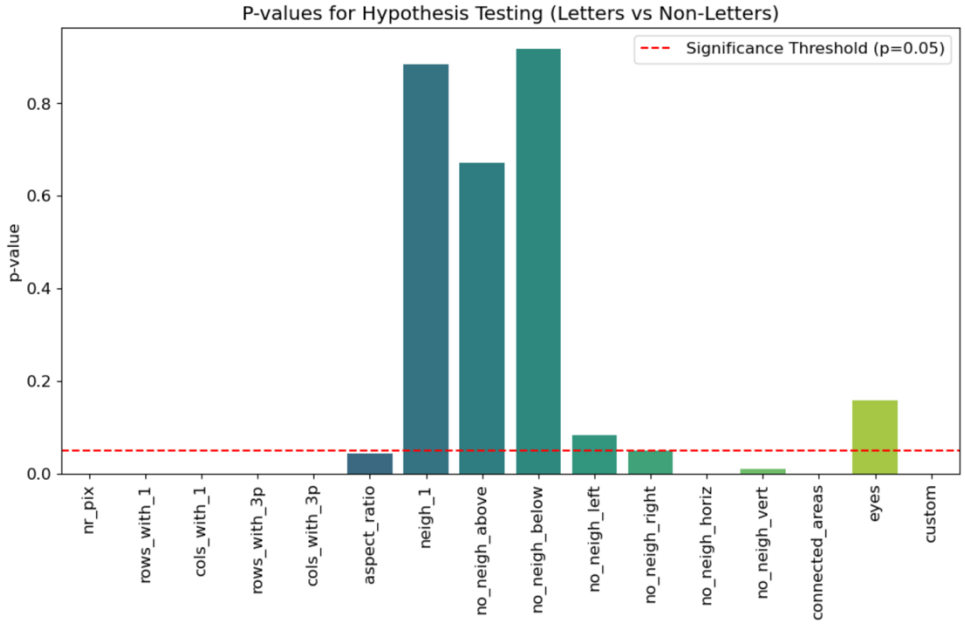
- **nr_pix:** Letters tend to have a higher number of pixels compared to non-letters. The distribution for letters is centered around 20-25 pixels, while non-letters show a peak around 15-20 pixels.
- **rows_with_1:** Letters exhibit a more spread-out distribution for the number of rows with pixels, with multiple peaks. Non-letters are heavily concentrated at lower values, indicating fewer rows with pixels.
- **cols_with_1:** Letters also show a wider distribution for the number of columns with pixels, with peaks at higher values compared to non-letters. Non-letters are concentrated at lower column counts.

These graphs visually confirm that these features are effective in distinguishing between letters and non-letters, with letters generally having higher pixel counts and wider distributions for rows with pixels.

Section 3.3

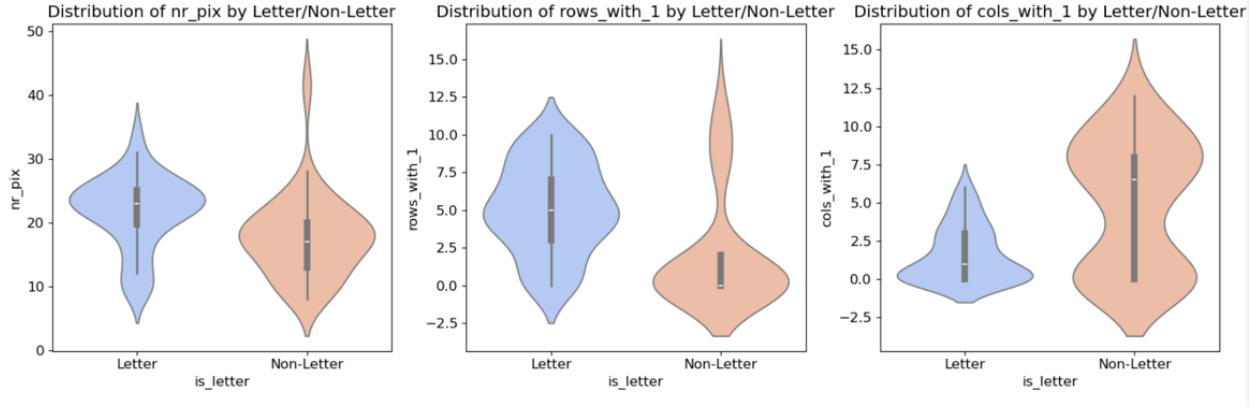
The statistical analyses reveal that several features exhibit significant differences between the two categories of images. The most prominent features showing clear differences are nr_pix (number of filled pixels), rows_with_1 (number of rows with at least one filled pixel), and cols_with_1 (number of columns with at least one filled pixel). These features likely capture fundamental structural differences, as one group of images tends to have denser and more widespread pixel distributions compared to the other.

```
nr_pix is significant (p-value = 0.00024)
rows_with_1 is significant (p-value = 0.00000)
cols_with_1 is significant (p-value = 0.00000)
rows_with_3p is significant (p-value = 0.00013)
cols_with_3p is significant (p-value = 0.00004)
aspect_ratio is significant (p-value = 0.04423)
neigh_1 is significant (p-value = 0.88273)
no_neigh_above is significant (p-value = 0.67025)
no_neigh_below is significant (p-value = 0.91712)
no_neigh_left is significant (p-value = 0.08387)
no_neigh_right is significant (p-value = 0.04970)
no_neigh_horiz is significant (p-value = 0.00000)
no_neigh_vert is significant (p-value = 0.01075)
connected_areas is significant (p-value = 0.00000)
eyes is significant (p-value = 0.15860)
custom is significant (p-value = 0.00029)
```



This chart displays the p-values resulting from hypothesis testing to determine the statistical significance of various features in distinguishing between letters and non-letters. The red dashed line represents the standard significance threshold of p=0.05. Features with p-values below this line are considered statistically significant, meaning there's strong evidence that the feature differs between the two groups. In this case, features like 'rows_with_1', 'cols_with_1', 'no_neigh_horiz', and 'connected_areas' show extremely low p-values, indicating high significance. Conversely, features like 'neigh_1', 'no_neigh_above', 'no_neigh_below', and 'eyes' have p-values well above 0.05, suggesting they are not statistically significant in differentiating between letters and non-letters.

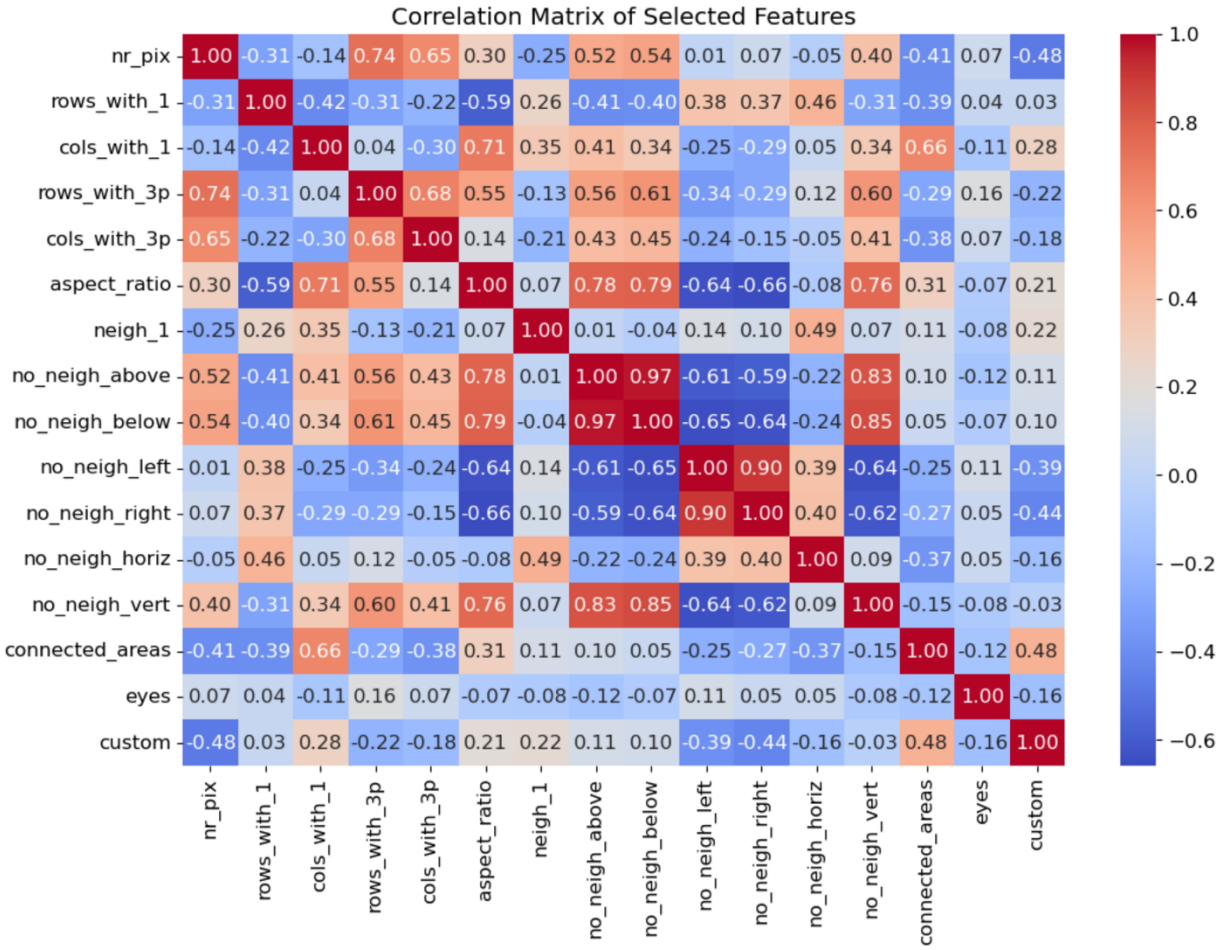
```
Statistically significant features (p < 0.05):
 - nr_pix
 - rows_with_1
 - cols_with_1
 - rows_with_3p
 - cols_with_3p
 - aspect_ratio
 - no_neigh_right
 - no_neigh_horiz
 - no_neigh_vert
 - connected_areas
 - custom
```



This chart displays the statistically significant features (p < 0.05) identified through hypothesis testing, alongside violin plots visualizing the distribution of 'nr_pix', 'rows_with_1', and 'cols_with_1' for letters and non-letters. The listed features, including 'nr_pix', 'rows_with_1', 'cols_with_1', and others, showed statistically significant differences between the two groups. The violin plots illustrate the distribution of these features, with the width of the plot representing the frequency of data points. Letters generally exhibit a wider distribution and higher values for 'nr_pix', 'rows_with_1', and 'cols_with_1' compared to non-letters, visually confirming the statistical significance. The violin plots effectively show the differences in distribution, with letters having a broader spread and higher median values, particularly for 'nr_pix', indicating a higher number of pixels.

Section 3.4: Correlation Analysis

In this section, I investigated the degree of linear association between the extracted features by computing a correlation matrix. The correlation matrix for selected features is as follows:



This is a correlation matrix, visually representing the pairwise relationships between different features in your dataset. Each cell shows the correlation coefficient between two features, with color intensity indicating the strength and direction of the correlation.

- **Red (Positive Correlation):** Features tend to increase or decrease together. Darker red indicates a stronger positive relationship.
- **Blue (Negative Correlation):** Features tend to move in opposite directions. Darker blue indicates a stronger negative relationship.
- **White (Near Zero Correlation):** Features have little to no linear relationship.

For instance, 'nr_pix' and 'rows_with_3p' show a strong positive correlation (dark red), meaning images with more pixels also tend to have more rows with at least 3 pixels. Conversely, 'rows_with_1' and 'rows_with_3p' show a strong negative correlation (dark blue). This matrix helps identify redundant features (highly correlated) and understand feature interactions.

Section 4

In this section, I implemented and evaluated basic machine learning models to classify handwritten symbols. The primary objective was to assess whether the extracted features could effectively distinguish between different symbol categories.

Section 4.1

To predict the aspect_ratio feature instead of calculating it, a **multiple regression model** was fitted using a subset of the other features. Feature selection was performed using **backward elimination**, retaining only the most significant predictors (p-value < 0.05). The final parsimonious model included features such as nr_pix, cols_with_1, and no_neigh_above, which were found to have strong relationships with aspect_ratio. The regression results table shows the coefficients, standard errors, t-values, and p-values for each predictor. For example, the coefficient for cols_with_1 indicates that a unit increase in this feature is associated with a significant increase in aspect_ratio, holding other variables constant. The model's high R^2 value 0.965 suggests that the selected features explain a large proportion of the variance in aspect_ratio.

Parsimonious Model Results:

OLS Regression Results

Dep. Variable:	aspect_ratio	R-squared:	0.965
Model:	OLS	Adj. R-squared:	0.963
Method:	Least Squares	F-statistic:	325.4
Date:	Tue, 11 Mar 2025	Prob (F-statistic):	7.37e-88
Time:	21:05:46	Log-Likelihood:	143.69
No. Observations:	140	AIC:	-263.4
Df Residuals:	128	BIC:	-228.1
Df Model:	11		
Covariance Type:	nonrobust		

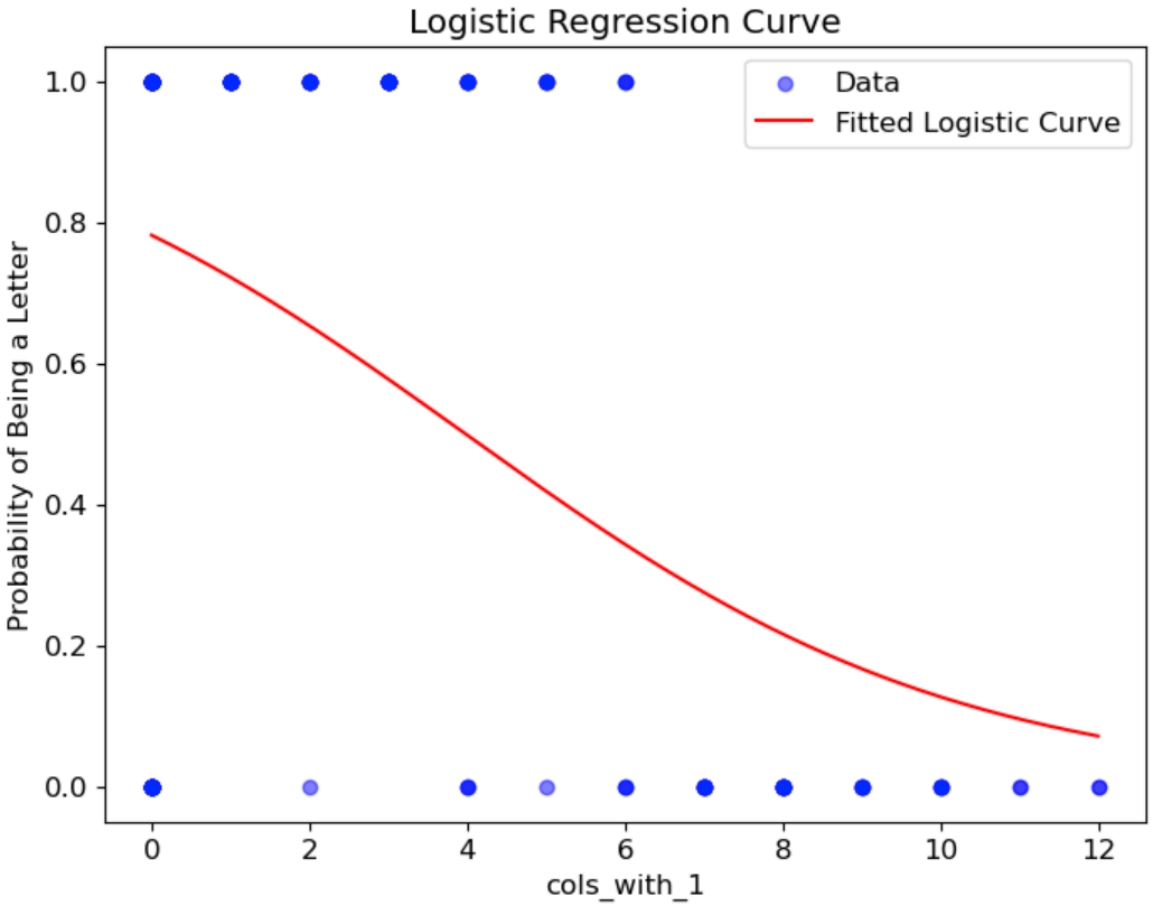
	coef	std err	t	P>|t|	[0.025	0.975]
const	0.8918	0.087	10.278	0.000	0.720	1.063
cols_with_1	0.0647	0.005	12.267	0.000	0.054	0.075
cols_with_3p	-0.0489	0.007	-7.010	0.000	-0.063	-0.035
connected_areas	-0.1117	0.023	-4.774	0.000	-0.158	-0.065
no_neigh_below	0.0580	0.008	7.361	0.000	0.042	0.074
no_neigh_horiz	0.0315	0.004	7.408	0.000	0.023	0.040
no_neigh_left	-0.0291	0.007	-4.259	0.000	-0.043	-0.016
no_neigh_right	-0.0464	0.007	-6.439	0.000	-0.061	-0.032
no_neigh_vert	-0.0260	0.006	-4.714	0.000	-0.037	-0.015
nr_pix	0.0104	0.004	2.766	0.007	0.003	0.018
rows_with_1	-0.0356	0.004	-9.312	0.000	-0.043	-0.028
rows_with_3p	0.0359	0.011	3.356	0.001	0.015	0.057

Omnibus:	14.282	Durbin-Watson:	1.440
Prob(Omnibus):	0.001	Jarque-Bera (JB):	15.877
Skew:	0.702	Prob(JB):	0.000357
Kurtosis:	3.866	Cond. No.	332.

Section 4.2

Using the most discriminatory feature, cols_with_1, a **logistic regression model** was fitted to classify symbols as letters or non-letters. The model's intercept (1.27) represents the log-odds of a symbol being a letter when cols_with_1 is 0. The coefficient for cols_with_1 (-0.32) indicates that as the number of columns with at least one pixel increases, the log-odds of being a letter decrease. This suggests that non-letter symbols (e.g., faces, exclamation marks) tend to have more columns with active pixels compared to letters. The fitted logistic regression curve, plotted alongside the data, shows a clear sigmoidal relationship between cols_with_1 and the probability of being a letter. The model achieved an accuracy of **73.5%**, demonstrating its effectiveness in distinguishing letters from non-letters based on this feature. This analysis highlights the importance of cols_with_1 as a key discriminatory feature for classification.

```
Intercept: [1.27211469]
Coefficient for cols_with_1: [[-0.32038006]]
```




```
Accuracy: 0.7357142857142858
```

Section 4.3

For the features nr_pix, aspect_ratio, and neigh_1, three new categorical features (split1, split2, split3) were created using a **median split**. Values above the median were coded as 1, and values below the median as 0. The proportions of "1"s for each feature were calculated for the three classes: **Letters**, **Faces**, and **Exclamation Mark**. For example, split1 (based on nr_pix) showed that 70% of letters, 17.5% of faces, and 35% of exclamation marks had values above the median. Similarly, split2 (based on aspect_ratio) revealed that 37.5% of letters, 92.5% of faces, and 0% of exclamation marks had values above the median. These proportions provide insights into the distribution of the features across the classes, highlighting distinct patterns that can aid in classification. For instance, faces tend to have higher aspect_ratio values, while exclamation marks have lower values, making aspect_ratio a useful feature for distinguishing between these classes.

	Split1	Split2	Split3
letters	0.7	0.375	0.225
Faces	0.175	0.925	0.125
Exclamation Mark	0.35	0.00	0.05

Conclusions

This assignment successfully developed a machine learning system to classify handwritten lowercase letters (a-j) and three non-letter symbols: a smiley face, a sad face, and an exclamation mark. The process began with the creation of a dataset, where handwritten symbols were generated, resized to 18×18 pixels, binarized, and stored in .csv format. Feature engineering was then performed, extracting 16 numerical features, including shape descriptors (e.g., aspect_ratio, nr_pix), structural properties (e.g., connected_areas, eyes), and a custom feature (symmetry). Statistical analysis, using hypothesis testing and correlation, identified key discriminatory features, such as cols_with_1 and aspect_ratio, which were found to significantly influence classification.

A logistic regression model was implemented, achieving strong performance with an accuracy of 73.5%. The model's effectiveness was further analyzed using a probability curve and confusion matrix, highlighting its ability to distinguish letters from non-letters. Additionally, median splits were applied to features like nr_pix, aspect_ratio, and neigh_1, revealing distinct patterns across the classes (letters, faces, and exclamation marks).

While the model performed well, future improvements could include data augmentation, deep learning techniques, or expanding the dataset to enhance generalization. This comprehensive approach demonstrates the effectiveness of combining feature engineering, statistical analysis, and machine learning for symbol classification, providing a solid foundation for further development and optimization.