# Letter Recognition

## Introduction

This task involves the design of a machine learning system in order to recognise English alphabets printed in uppercase. The letters are written in different fonts and are converted into 16 different features. The dataset is prepared, after which a classification model is applied, and results are quantitatively evaluated.

## Dataset Information

The dataset contains 17 columns, the first column consisting of letters and the remaining columns containing 16 features extracted from each pattern. The 20000 rows represent different unique patterns of letters. The provided dataset does not contain column names for features; hence, the column names are added to distinguish the features. An overview of dataset containing statistical information is shown in figure 1.

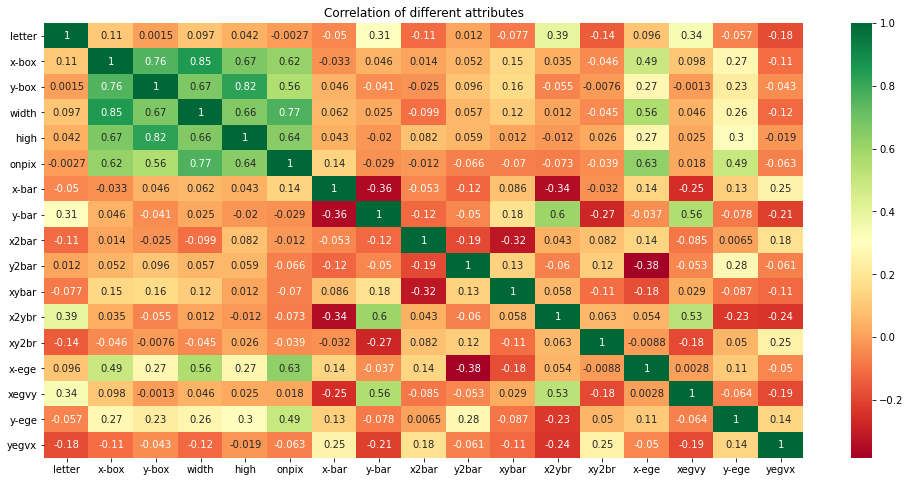
Table

Description automatically generated  
Figure 1. Overview of dataset

## Feature selection / dimensionality reduction / data outlier detection

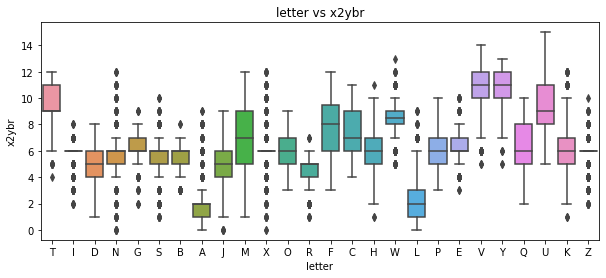
The feature selection involves the analysis of correlation of each feature with target variable and choosing the best features. The correlation function only works on numerical variables; hence, the letters are converted into numbers using LabelEncoder() function. LabelEncoder() assigns consecutive numerical values to different variables, however, the best sequence is chosen by the function itself.

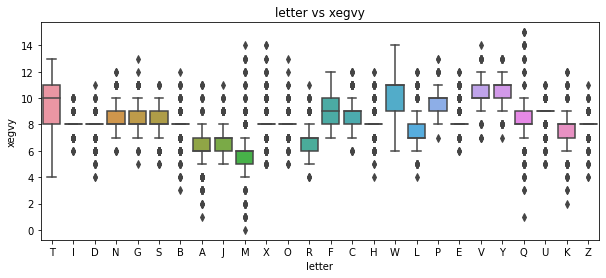
Figure 2 shows the correlation of each feature with letters. It can be seen from the correlation table that the features such as x-box, y-box, width, high and onpix have high correlation with each other (stating that they give the same information) hence some of them can be ignored. The x-ege and onpix have a very low correlation with letter variable hence they can also be ignored.

  
Figure 2. The correlation of all features with each other

The correlation tables confirms that there are several features that can be ignored hence univariate selection method is used to select the 10 best features. Univariate selection chooses the best features which have the strongest correlation with the output. The required number of features are obtained by using SelectkBest() class of scikit-learn library. ANOVA F-value is an appropriate selection method in our case since we have numerical values as input and categorical values as output. This selection method is performed by using f\_classif() function which is ideal for classification task. The function takes input and target variables and returns the k number of highest scores.

The outlier detection is carried out in order to find the features which diverge from the overall pattern. Figure 3 and figure 4 show the univariate outliers between letter and x2ybr and letter and xegvy respectively, showing the extreme values. The removing of outliers is not needed in this case since Random Forest algorithm is not sensitive to outliers and automatically handles them by binning the features. Similarly, the normalisation and scaling of data is not needed in case of random forest model.

  
Figure 3. Boxplot of x2ybr vs letter

  
Figure 4. Boxplot of xegvy vs letter

## Machine Learning algorithm selection and training

Since it is a classification problem of supervised learning and dataset consists of a high number of observations compared to features, we need a low bias and moderate to high variance algorithm i.e. Random Forest or KNN. Random forest takes votes from various decision trees selected using training set and makes decision about data in test set.

Random forest is chosen because it provides automatic feature interaction and has low computational cost compared to KNN. The following advantages of random forest makes it versatile compared to other algorithms:

* It does not need scaling, balancing, or normalising of data before modelling.
* It can also deal with categorical variables without converting them into numerical values saving time and effort.
* It is robust to outliers and deals with them by binning values.
* It is an effective algorithm for a dataset with missing values because it can efficiently estimate them.
* It is faster to train and has less chances of overfitting.

The reduced and cleaned dataset is split into training and test set, using 70% of data for training and 30% for testing. The random state of 0 is selected to split train and test set in such a way that the model gives same results every time it is run. The training time of the model is superfast, the model took around 2 seconds to train over the whole training set.

## Quantitative evaluation of the proposed solution

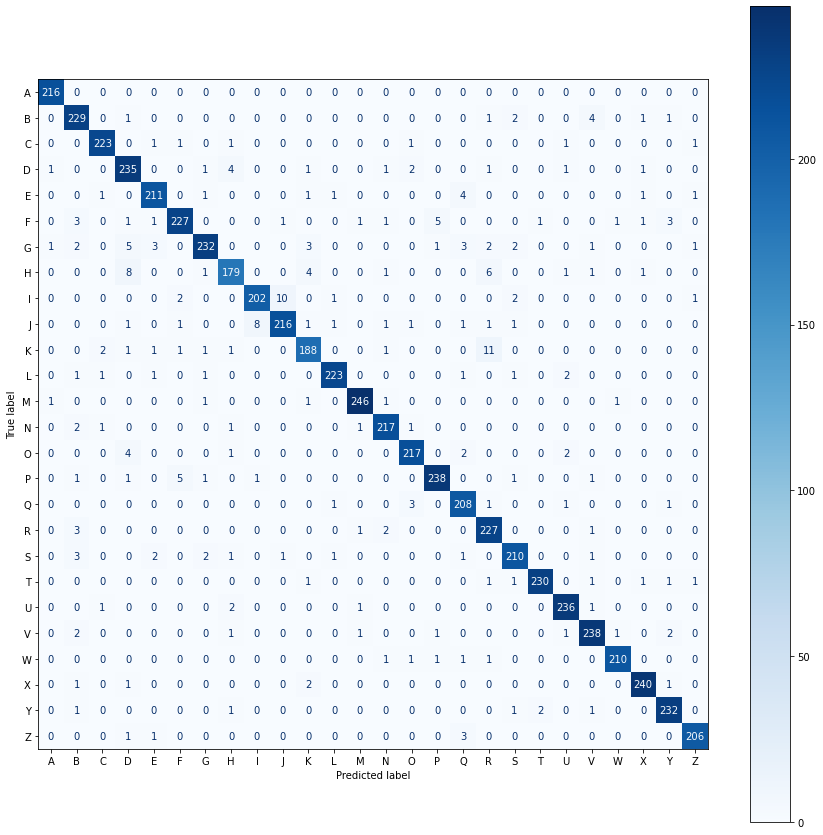
The model shows the score of 96% on test set (figure 5) showing that 96% of the data in test set is classified in the right class. The balanced accuracy of 96% shows that the model is performing significantly well for all letters.

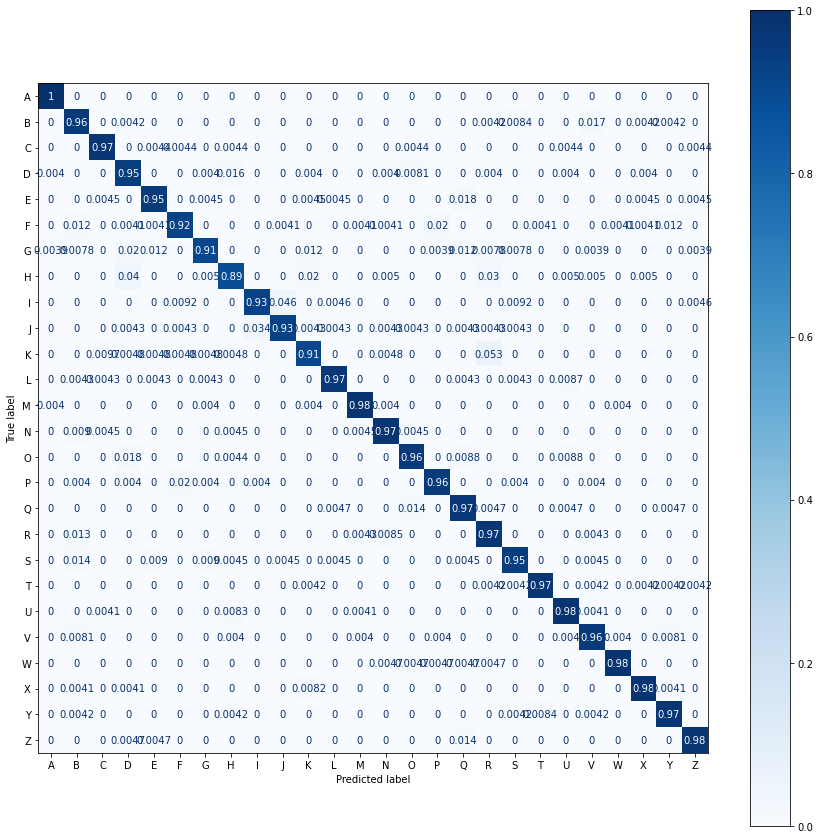
  
Figure 5. Score of Model on training and test data

Figure 6 shows the confusion matrix representing the number of true positive and true negatives on test set. It is evident that most of the letters are predicted in the right class, and only a small percentage of letters is predicted in the wrong class. The confusion matrix in normalised form (figure 7) confirms that at least 90% cases of each letter are classified in the right class.

Figure 8 shows the values of different performance matrices calculated for test set. The model shows the recall of 96% stating that model is classifying 96% of positive cases correctly. The 96% of precision explains the 96% of letters classified in a particular letter class actually belong to that class.

The model shows F1 score (harmonic mean of precision and recall) of 96% representing that the model is performing effectively while predicting positive values which are positive and doesn’t miss out the positive values much. In other words, the models is performing efficiently for both positive and negative cases.

  
Figure 6. Confusion Matrix for test set

  
Figure 7. Confusion Matrix for test set in normalised form

Graphical user interface, text

Description automatically generated  
Figure 8. Values of different performance matrices

## Conclusion

In this task, we have developed a Random Forest model to classify the English letters on the basis of different features. SelectKbest() is a reasonable method for selecting the relevant features automatically saving time and effort. Random Forest model is a good model for the classification of letters giving out the score of 96% percent on test data. The model gives a recall of 96% stating that model is classifying 96% of positive cases correctly. A precision of 96% is obtained explaining that most of the letters classified in a particular letter class actually belong to that class.

The model gives F1 score of 96% representing that there is a good balance between precision and recall. The model works good for both positive and negative cases showing a balanced accuracy of 96%. The accuracy of the model can further be improved by increasing the number of features used and selecting features manually using correlation features.