| | |
|---|---|
| **Course code and name:** | F20DL: Data Mining and Machine Learning |
| **Type of assessment:** | Group |
| **Coursework Title:** | Data Analysis and Bayes Nets |
| **Student Names:** | Hana Khan, Neil Patrao, Yasir Rayamarakar Veetil, George Chandy, Fatima Patel |
| **Student ID Numbers:** | H00379151, H00385850, H00385488, H00377409, H00339652 |

**Declaration of authorship. By signing and uploading this form:**

□ **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

□ I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

□ I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

**Student Signature:** *Hana Khan, Neil Patrao, Yasir Rayamarakar Veetil, George Chandy, Fatima Patel*
**Date**: *23/10/2023*

Please fill in the above and upload to Canvas.

**Your work will not be marked if a signed copy of this form is not included with your submission.**

Data Mining and Machine Learning

# Data Analysis and Bayes Nets

F20DL CW PART ONE

*Group11 {Hana Khan, Neil Patrao, Yasir Rayamarakar Veetil, George Chandy, Fatima Patel}*

*Stop Sign Dataset Notebook:*
https://colab.research.google.com/drive/1uXxENq8Ltam_Sd_kfaixYP_kWj5pHeDg?usp=sharing

*Additional Datasets Notebook:*
https://drive.google.com/file/d/1ebdISJrLwY8L7qWw5nh7jm0k_oLSpmDY/view?usp=sharing

**ABBREVIATIONS**
NBC – Naïve Bayes Classifier
CV- Cross Validation

## About the Data

- They were a total of 2304 features with no missing values, duplicates, and no outliers since all features (pixel values) were in range 0 and 255. Without any feature reduction techniques, the dataset is very high dimensional and could suffer from "*The Curse of Dimensionality.*"
- When plotted, some images are too bright, dark, misaligned, or blurry.
- Therefore, pixel intensity histogram was plotted and there was greater pixel frequency closer to 0 and 255 meaning many pixels were close to black and white indicating there's relatively high contrast (lightness/darkness). The lower mid-range frequencies are because of the lack of details in the images.
- The classes were highly imbalanced, and this could lead to model bias towards the larger classes, misleading metrics (e.g., accuracy) and difficulty in evaluating whether the model is performing well.
- Min-max normalization was used and scales pixel values between 0 and 1, therefore, the relative differences between the pixel values are preserved. This allows inherent patterns and features within the image to be maintained. All datasets were normalized before running on NBC.

## Naïve Bayes Classifier (NBC) on Pre-processed Data

After running the NBC on the data without any preprocessing:

- Class 0, 6 and 7 have relatively high recall rates, AUC but low precision and relatively low number of samples.
    - The high recall suggests the classifier is highly capable of distinguishing between positive and negative instances of the Classes, however, the low precision indicates that out of all the instances the classifier predicted as that Class, a significant proportion of them were false positives.
- For class 7, recall was 0.69 which is relatively high, TP was 37 compared to FP which was 338 which means the classifier classified a lot of images as class 7 but only a small portion of these were correct. Using only the recall/TP withoout checking the FP rate would have let us to believe the classifier was doing its job well.
- This could be because:
  Feature Noise: If features representing a Class are not distinct from other classes, the model might struggle to differentiate that Class from other classes and identify patterns that are not truly representative of the Class leading to false positives.
  Class Boundaries: the class has overlapping feature space with other classes.
  Inadequate Model Complexity.
- Classes 1, 2, 4, and 8 have AUC values closer to 0.5, which means the classifier's ability to distinguish between the positive and negative classes is not much better than random guessing for these classes.

These results show that plotting ROC curves are not enough on their own for imbalanced datasets. ROC curves tend to be overly optimistic for imbalanced datasets since it considers both the positive and negative classes equally. Therefore, precision-recall curves were also plotted for some of the classifiers.

- Interpreting the precision-recall curve for the NBC on the unprocessed data:
    - Most of the curves for individual classes touch or cross the f1=0.2 line, indicating that, at some threshold, they achieve at least an F1 score of 0.2. Fewer classes reach the f1=0.4 line, and even fewer touch the f1=0.6 line.
    - Some classes have low AP values and most of those classes are underrepresented in the dataset, making them harder to classify. Additionally, they could be more challenging to differentiate due to overlapping feature distributions.

- Cohens Kappa score was 0.1566 indicating slight agreement and that the results are slightly better than that by random chance.

### *Cross Validation (CV):*

- train_test_split() is a holdout method so the error estimate may not be very reliable.
- When CV was done on the dataset without any preprocessing techniques applied, the accuracy dropped to 18.65% compared to 24.01% for NBC. This could be due to the data imbalance and some folds in cross-validation might end up with very few samples of a minority class, making it hard for the model to learn that class. To address this problem, we used stratified 10-fold CV.

### *Stratified 10-fold Cross Validation:*

- ten-fold CV was used since experiments show this is the best option to get accurate estimates.
- In a normal cross-validation, samples are randomly divided into folds which doesn't preserve the original class distribution. Stratification ensures that each fold maintains the original class distribution and reduces the estimate's variance.
- This was proved since the accuracy increased to 22.73% when running on the NBC without any preprocessing. Stratified 10-fold CV was applied to all the NBC after feature extraction and the resulting accuracies ranged higher or lower 2%.

## Feature Extraction, Naïve Bayes Classifier (NBC) Results & Conclusions

Two feature extraction techniques were used separately and then combined:

- Filter feature extraction using top correlating features.
- PCA (Principal Component Analysis)

All the feature extraction methods were evaluated using Naïve Bayes Classifier metrics, TP/FP, Cohens Kappa, ROC curves, Precision-Recall Curves, and macro averaged ROC curves.

**1. Extracting top 20, 10 and 5 most correlating features with the target class**

- To reduce the dimensionality of the data and only keep the most relevant features in relation to the target class.
- For NBC on the top features, the AUC, macro precision, recall, accuracy, kappa increased.
- The metrics for top 5 was slightly better than top 10 which was better than top 20.

**2. Extracting top 20, 10 and 5 most correlating features each other the target class with a correlation < 95% with each other**

- Since it's an image dataset, it contains spatial dependencies between the raw pixels and structures. Naive Bayes assumes that features are conditionally independent given the class label. This assumption is often violated in image datasets, where neighboring pixels are often highly correlated.
- Therefore, any features that had a correlation greater than 95% in the top 20/10/5 datasets, one feature was dropped. The threshold for 95% was chosen after running 80%, 85%, 90% and checking which classifier outputted the best metrics. There was a gradual increase in the accuracy as the % threshold increased but after 95%, the accuracies would dip.
- Compared to method 1 above, the accuracy, macro-Ave precision, recall and f1-score increased. For this method, top 10 features returned the best metrics, while top 5 and 10 returned similar metrics which were slightly worse.

**3. PCA on all 2304 features**

- PCA is a dimensionality reduction technique and is used to transform a set of possibly correlated features into a smaller set of uncorrelated linear combinations called principal components.
- PCA with 500 components was randomly chosen and the metrics had already significantly improved with macro precision increasing from 0.34 to 0.5 and the accuracy increased from 24% to 30%. The f1-score increased to 0.34 from 0.26 for NBC of unprocessed data.
- PCA 129, which was the number of components capturing 97% of the data variance yielded even better macro-Ave f1-score of 0.5 and accuracy of 45%.
- The 97% threshold was chosen after experimenting with different % and checking the metrics. 97% yielded the best in this case.
- This is because multicollinearity and dimensionality were reduced improving the models' predictions.

**4. PCA on top 20, 10 and 5 most correlating features**

- Collinearity with the top features can be reduced by PCA. The effect of this problem (of multicollinearity) has been reduced in method 2 above by removing features that have a correlation less than 95% to each other. This is another possible technique to mitigate this problem.
- This technique yielded the highest results compared to all the other methods. PCA on top 5 features yielded an accuracy of 59.92%, kappa score 0.51, AUC of 0.91, macro-Ave f1 score of 0.53 compared to 0.24 in the NBC on the unprocessed data.
- To pick the number of components for the PCA on top 20/10/5 features, different variance thresholds were experimented with and for top 20, 97% yielded the best. And likewise, was done for top 10 and 5.

**5. PCA before filter extraction of top 20, 10 and 5 most correlating components to the target class**

- When applying PCA first, the data is transformed to a set of principal components, some of which may not have a strong relationship with the target. Using a filter method after could help drop the least correlated components.
- Two PCA's were tested with varying number of components before correlation filter selection:

PCA with 73 components: 73 components capture 97% variance of the original dataset and was found to yield one of the highest accuracies (after running Naive Bayes on the dataset with 73 components)

PCA with 500 components**:** a relatively high number of components was chosen so that there are more components to compare correlation with the target class.

*ASSUMPTION: This would mean each component is a combination of fewer features and therefore when finding the correlation with the target class, only combinations that are highly correlated would be kept in the final dataset. Would this mean the dataset would better train the model since it's better fitted?*

- The assumption turned out to be correct, and filter feature extraction on the PCA with 500 components yielded better results with 44.74% accuracy vs 42.67% and an f1-score of 0.47 vs 0.44.

### *Overall Conclusions*

Since the dataset was extremely imbalanced, multiple metrics were used together to evaluate the feature extraction methods. Filter feature extraction using top correlations helped with the model's prediction but pairing it with PCA significantly increased the models ability to classify. Models trained on fewer most relevant features tend to perform better than extremely high dimensional data since they avoid the curse of dimensionality.