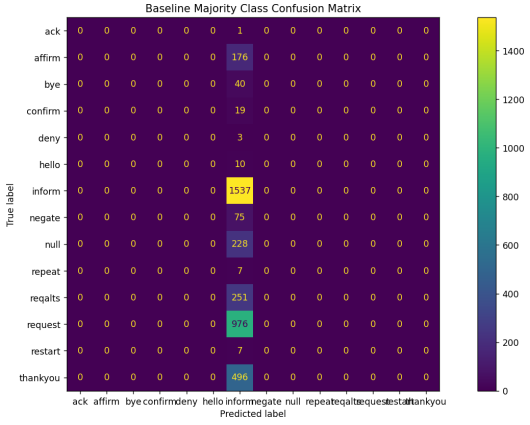
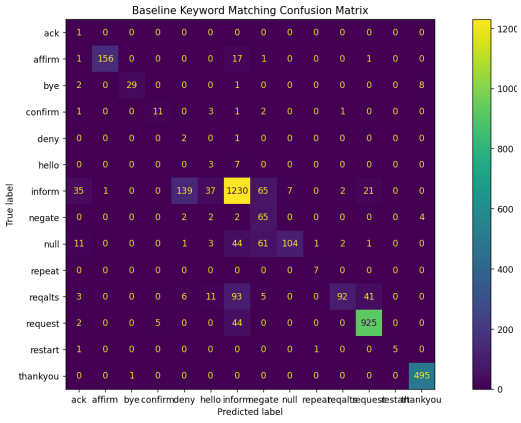
Group30 - Part1a - Text Classification - Evaluation

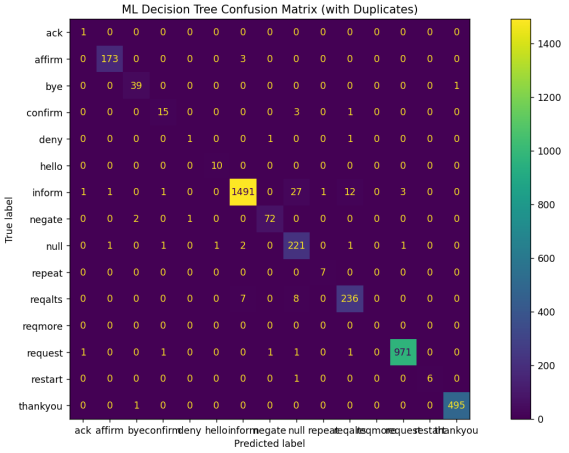
1. Quantitative evaluation: Evaluate your system based on one or more evaluation metrics. Choose and motivate which metrics you use.

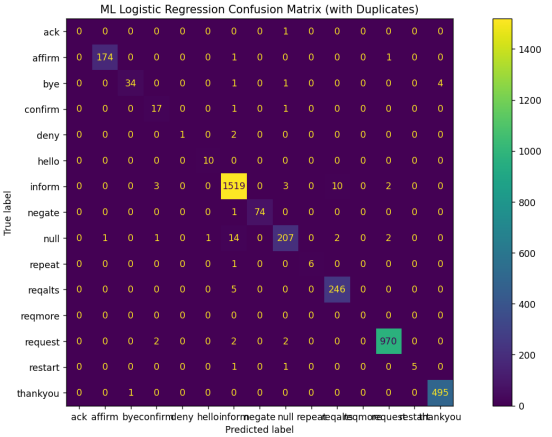
We decided to use accuracy, precision, recall, and f1-score to evaluate our systems. Each metric is biased/has its own strengths so using multiple metrics has the advantage of getting a broader view over the performance of the systems. It is limited to rely only on one metric e.g. accuracy because a model with high accuracy can be a very bad model if it always predicts the majority class. If the majority class is 95% of the data, then the accuracy will be 95% too but the model has not learned anything. Accuracy measures how many predictions were correct, but using this accuracy alone might be misleading as our example from before shows. Recall (also called true positive rate) is calculated by dividing the number of true positives by the number of true positives summed with the number of false negatives. Therefore, it measures how good the positive class can be recognized by the system. If it is not close to 1, it shows us that our system misses instances which should have been labeled positive. Precision is calculated by dividing the number of true positives with the number of true positives summed with the number of false positives. If it is not close to 1, it shows us that there are many instances falsely classified as positive. Furthermore, because we have chosen to use recall as an evaluation metric it naturally follows that we also use precision. Because recall measures how many of the positive instances were correctly predicted as positives and precision measures how many of the positive predictions were positive instances, it is good to consider both metrics. The F1-score is the harmonic mean of recall and precision. It therefore is more robust than accuracy regarding unbalanced classes, but on the other hand does not consider the true negatives.

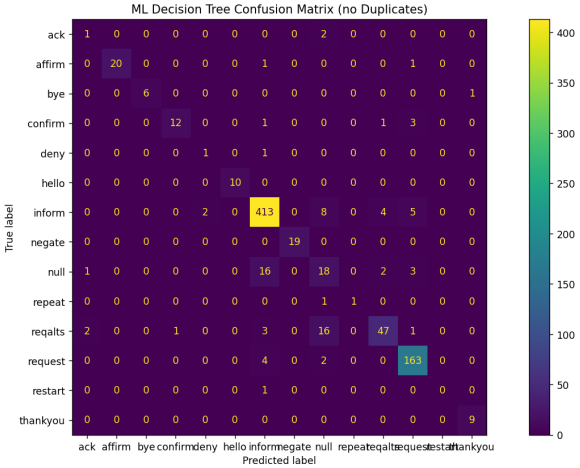
2. Error analysis: Are there specific dialog acts that are more difficult to classify? Are there particular utterances that are hard to classify (for all systems)? And why?

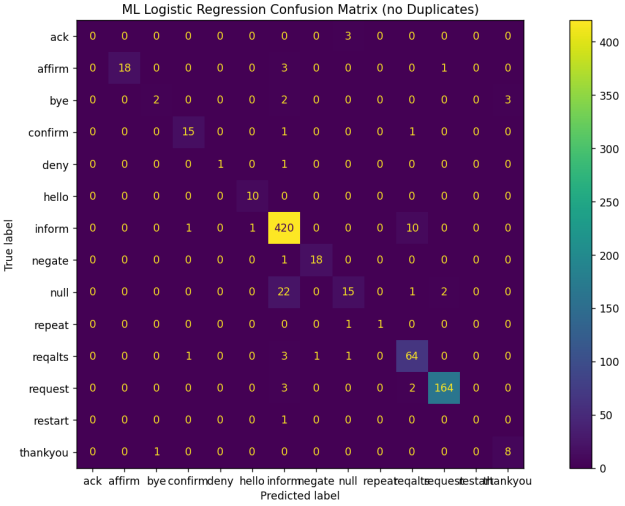












The rule-based Baseline model has difficulty correctly identifying utterances of the classes ‘hello’,’null’ and ‘reqalts’, as shown in the image above. The machine learning models generally perform well on all classes except sometimes for the smaller classes, e.g., the logistic regression models for the ‘ack’ class. The majority of misclassified utterances were classified as ‘informal’. This is to be expected, since the ‘informal’ class is the largest.

3. Difficult cases: Come up with two types of ‘difficult instances’, for example utterances that are not fluent (e.g. due to speech recognition issues) or the presence of negation (I don’t want an expensive restaurant). For each case, create test instances and evaluate how your systems perform on these cases.

The first ‘difficult instance’ type are utterances with shortenings. A test insurance is ‘thats okay’ which should be classified as ‘affirmation’. Another type of ‘difficult instances’ are instances with a negation. A test insurance for this case is ‘that is not good’ which should be classified as ‘deny’ or as ‘negate’.

Model	thats okay	that is not good
Majority Baseline	inform	inform
Rule-based Baseline	ack	deny
Decision Tree with duplicates	null	confirm
Decision Tree without duplicates	null	confirm
Logistic Regression with duplicates	null	confirm
Logistic regression without duplicates	null	confirm

4. System comparison: How do the systems compare against the baselines, and against each other? What is the influence of deduplication? Which one would you choose for your dialog system?

Baseline majority accuracy: 40.12%

Macro Avg.	Precision: 0.03	Recall: 0.07	F1-score: 0.04
Weighted Avg.	Precision: 0.16	Recall: 0.40	F1-score: 0.23

Baseline keyword accuracy: 80.92%

Macro Avg.	Precision: 0.70	Recall: 0.78	F1-score: 0.65
Weighted Avg.	Precision: 0.89	Recall: 0.81	F1-score: 0.83

Machine Learing Decision Tree Classifier accuracy (with Duplicates): 97.83%

Macro Avg.	Precision: 0.88	Recall: 0.88	F1-score: 0.88
Weighted Avg.	Precision: 0.98	Recall: 0.98	F1-score: 0.98

Machine Learing Decision Tree Classifier accuracy (without Duplicates): 87.94%

Macro Avg.	Precision: 0.83	Recall: 0.79	F1-score: 0.77
Weighted Avg.	Precision: 0.91	Recall: 0.88	F1-score: 0.89

Machine Learing Logistic Regression Classifier accuracy (with Duplicates): 97.57%

Macro Avg.	Precision: 0.77	Recall: 0.73	F1-score: 0.74
Weighted Avg.	Precision: 0.97	Recall: 0.98	F1-score: 0.97

Machine Learing Logistic Regression Classifier accuracy (without Duplicates): 89.68%

Macro Avg.	Precision: 0.65	Recall: 0.59	F1-score: 0.61
Weighted Avg.	Precision: 0.89	Recall: 0.90	F1-score: 0.89

The results show us that all machine learning systems have a higher accuracy and higher weighted average for precision, recall and F1 than the two baselines.

Comparing the logistic regression with decision trees where duplicates were in the data we can notice that the accuracies and the weighted average for precision, recall and F1-score are close to each other.

Comparing the logistic regression with decision trees where duplicates were left out of the data we notice that the accuracies are still close to each other, however the logistic regression was slightly better (with 2

percent). Furthermore when we compare the models and look at the weighted averages of precision, recall and F1-score we can notice that they are close to each other.

The influence of deduplication is that the accuracy gets lower as well as the weighted averages for precision, recall and F1-score.

With the duplicates the data consists of ca. 25000 samples, without them only ca. 5000 samples remain. If we train the models with duplicates, it is very likely that the models overfit to the duplicates but may underperform regarding the utterances that appear only once or to unseen data. This is acceptable if we are not interested in treating all utterances in the same way e.g. if we assume the ‘real world’ to be accurately represented in the training data. Generally it should be tried to achieve generalization therefore using the models without duplicates would be better because then we try to learn rules and not a distribution.
Therefore, in the following we will only consider models trained on data without duplicates.
The logistic regression model has a higher accuracy score than the decision tree model. If we care about the overall performance we would therefore choose the logistic regression model. On the other hand, the macro average values of the precision-, recall-, and f1-score are higher for the decision tree but the weighted average scores are almost equal. That means that the decision tree can predict the small classes better and the big classes worse than the logistic regression. Depending on whether we prefer to be able to predict the small or big classes more precisely we would choose either the logistic regression or the decision tree as our model.