**Evaluation classifier**

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

We will use accuracy for the evaluation of the classifier, because this gives an overall correctness rate which is simple to interpret and compare between models. We will also use the F1 score, because it balances precision (correctly predicted positives) and recall (actual positives). This means you will only get a high F1 when both precision and recall are high. Accuracy further complements F1 by showing how often the model is right.

*Accuracy:*

| **Classifier** | **With duplicates** | **Without duplicates** |
| --- | --- | --- |
| Decision Trees | 0.9754312598013591 | 0.8917910447761194 |
| Logistic Regression | 0.9806586513329848 | 0.9253731343283582 |
| MLP | 0.9866701515943544 | 0.9340796019900498 |

*F1 score:*

| **Classifier** | **With duplicates** | **Without duplicates** |
| --- | --- | --- |
| Decision Trees | 0.9196380199984053 | 0.7348332157091416 |
| Logistic Regression | 0.9136553046026682 | 0.688384161719927 |
| MLP | 0.9332724507631597 | 0.6966839391606229 |

As we can see, accuracy drops when duplicates are removed. This could be because duplicates make the dataset “easier” by having repeated examples. MLP has the highest accuracy, followed by logistic regression, and then decision trees.

The F1 score drops even more than accuracy when duplicates are removed. This can mean that some classes are harder to classify correctly without duplicates, just like accuracy. With duplicates MLP scores the highest, but without them, decision trees achieve a better score.

**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? Note: this analysis is about real utterances from the dataset.**

Below are the three most difficult to classify dialog acts per classifier:

| **Classifier** | **With duplicates** | **Without duplicates** |
| --- | --- | --- |
| Decision Trees | Ack, deny, confirm | Deny, ack, bye |
| Logistic Regression | Ack, deny, repeat | Deny, ack, bye |
| MLP | Ack, deny, repeat | Deny, ack, bye |

Acknowledgements, deny, confirm and dialog end seem to be the hardest to classify for all systems. Acknowledgements can be difficult because they often are short, and ambiguous. They can also overlap with confirmations, like “yes, that is right”. Deny, just like acknowledgement, is also difficult because they often are short. They can also be misclassified because of negated confirmation. Confirmation may be difficult for the same reason as acknowledgement, because they can overlap. Dialog ends can be difficult because they are short, and because it could be hard to recognize more subtle ways of ending the dialog.

**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. Note: this analysis is about sentences that you write yourself with the goal to have the system produce an incorrect result.**

As discussed before, negation and final utterances appear to be difficult, which is what we will base our test instances on. We will use the following test instances to evaluate how the model performs on them:

1. “I don’t want to eat anything that is not Italian” (double negation, inform)
2. “I don’t disagree with your choice of restaurant” (double negation, confirm)
3. “Great, that’s all I needed” (implicit final utterance, bye)
4. “Okay, I think I’m good now” (implicit final utterance, bye)

*With duplicate data:*

| **System** | **1** | **2** | **3** | **4** |
| --- | --- | --- | --- | --- |
| Decision Trees | inform | inform | nan | nan |
| Logistic Regression | inform | request | nan | ack |
| MLP | inform | inform | nan | ack |

*Without duplicate data:*

| **System** | **1** | **2** | **3** | **4** |
| --- | --- | --- | --- | --- |
| Decision Trees | inform | inform | deny | ack |
| Logistic Regression | deny | request | nan | ack |
| MLP | deny | inform | nan | ack |

If the label of a double negation is inform, this is easy to predict for all models, especially with duplicate data. However, it is harder when the negation is based on rarer acts like confirm, both with and without duplicates. The models misinterpret the negation and this results in the wrong class.

Implicit final utterances are always either misclassified or produce NaN. The models fail to generalize on polite, implicit closings. The effect of duplicate data once again is present; without duplicate data, errors increase. It appears that instance 3 is hard to even classify, and in instance 4, the word okay causes it to be misclassified as acknowledgement.

Decision trees handle simple inform double negation well, but misclassify rarer cases.

Logistic regression does struggle with double negation, but sometimes predicts closer to the right class than decision trees.

Logistic regression has some inconsistent behavior, misclassifying double negations.

All models fail consistently in final utterances, showing that this is the most difficult category.

**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?**

We will compare the systems, again, based on accuracy and on the F1 measure.

*Accuracy:*

| **System** | **With duplicates** | **Without duplicates** |
| --- | --- | --- |
| Majority Label Baseline | 0.40224777835859904 | 0.5733830845771144 |
| Rule-Based Baseline | 0.8128593831677993 | 0.75 |
| Decision Trees | 0.9754312598013591 | 0.8917910447761194 |
| Logistic Regression | 0.9806586513329848 | 0.9253731343283582 |
| MLP | 0.9866701515943544 | 0.9340796019900498 |

*F1 score:*

| **System** | **With duplicates** | **Without duplicates** |
| --- | --- | --- |
| Majority Label Baseline | 0.03824790307548928 | 0.05206098249576511 |
| Rule-Based Baseline | 0.3253361715768405 | 0.2806219170140761 |
| Decision Trees | 0.9196380199984053 | 0.7348332157091416 |
| Logistic Regression | 0.9136553046026682 | 0.688384161719927 |
| MLP | 0.9332724507631597 | 0.6966839391606229 |

As seen above, the systems do very well against both the baselines. Both with and without duplicates, the systems achieve far better scores. The majority baseline scores very poorly in F1, because it only predicts the most common label. Rule-based baseline is better than it, in both accuracy and F1, but still falls short against the other models.

In almost all cases, accuracy and F1 drop for every model when the duplicates are removed. This means that models have to generalize more because they have not seen the same instance twice, which makes it harder and results in a lower performance.

When using data with duplicates, all 3 machine learning models perform very similarly. But without them, MLP and logistic regression are slightly better than decision trees, especially in accuracy. But decision trees outperform the others on F1 without duplicates.

Overall, MLP has the highest accuracy, but decision trees have the highest F1. This shows that decision trees are better at balancing precision and recall across all classes. If the goal is the most amount of correct predictions, we should use MLP. But if we want a balanced performance, the choice should be decision trees.