COMP9417 Project: Can I Speak to the manager? Machine Learning for Customer Feedback Classification

March 31, 2025

Project Description

In modern manufacturing, efficiently managing customer feedback is essential for improving products and addressing concerns. This project focuses on developing a machine learning model to automatically classify customer comments related to **28 different products** and direct them to the appropriate departments within the company. The dataset consists of **10,000 training instances**, with each comment represented by **300 features** extracted using *natural language processing* (NLP) techniques. These features capture key linguistic and contextual elements to enhance classification accuracy. The goal is to build a robust **multiclass classification model** that ensures each comment is assigned to the correct department, streamlining the feedback management process. As a data scientist, your role is to develop a solution that enhances response efficiency and optimizes workflow within the company by accurately classifying and directing customer feedback to the appropriate departments. The dataset will be made available at 5pm Monday 31st March.

Description of the Data

The dataset is provided in CSV format and consists of three subsets:

- Training set:
$$\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, 10000\}$$
- Test set 1:
$$\mathcal{D}_{\text{test1}} = \{\mathbf{x}_j \mid j = 1, \dots, 1000\}$$
- Test set 2:
$$\mathcal{D}_{\text{test2}} = \{(\mathbf{x}_k, y_k) \mid k = 1, \dots, 202\} \cup \{\mathbf{x}_k \mid k = 1, \dots, 1818\}$$

Note that Test set 2 only needs to be used for the last part of the project on ”Unexpected Model Performance” (see below). In Test set 2, you are given access to 202 labeled points and 1818 unlabeled points. Each instance $\mathbf{x}_i \in \mathbb{R}^{300}$ represents a customer comment transformed into a feature vector of dimension 300 using NLP techniques. The corresponding label y_i is assigned from a set of **28 distinct categories**:

$$y_i \in \mathcal{C} = \{c_1, c_2, \dots, c_{28}\}$$

where each category c_k $(k = 1, \dots, 28)$ corresponds to a department responsible for handling feedback on a specific product. The task is to learn a function:

$$f : \mathbb{R}^{300} \to \mathcal{C}$$

that maps each feature vector $\mathbf{x}_i$ to the correct category y_i, ensuring accurate classification of customer feedback.

The final model will be evaluated based on its classification accuracy in assigning unseen test instances $\mathbf{x}_j$ to the correct category in $\mathcal{C}$.

Important Aspects

The following problems should be considered and discussed in detail in your report:

- **Data:** Perform exploratory data analysis (EDA). This should include a pre-processing step in which the data is cleaned. You should pay particular attention to the following questions:

  1. Which features are most likely to be predictive of each target class?
  2. How does the class imbalance affect the learning process, and what methods can mitigate this issue?
  3. What are the appropriate evaluation metrics for this task, and why do accuracy-based metrics fail in imbalanced classification?

- **Research:** Provide a summary of state-of-the-art methods for handling imbalanced multi-class classification tasks. Be sure to rigorously explain some of the algorithms that are used. It is a good idea to pick one or two areas to explore further. The report should be well-written and well-referenced.

- **Modeling:** The approach to modeling is open-ended, and you should think carefully about the types of models you wish to deploy. Instead of building a large number of generic models, focus on well-justified choices and their impact. Regardless of the models you choose, you need to:

  1. Construct a model that performs well in terms of classification metrics suitable for imbalanced data.
  2. Compare different strategies for handling class imbalance within machine learning models.
  3. Investigate ensemble techniques to improve performance over individual models.
  4. Evaluate models not only on overall performance but also on per-class metrics to assess minority class performance.

- **Discussion:** Provide a detailed discussion of the problem, your approach, and your results. Explain whether your final approach was better than a simple baseline classifier and justify why. Discuss limitations and potential future improvements.

- **Unexpected Model Performance in New Test Data Deployment:** After deploying a customer feedback classification model, the manufacturing departments report a significant drop in classification accuracy for newly received feedback messages. Some departments are mistakenly receiving messages unrelated to their operations, leading to inefficiencies in workflow and customer service.

  This issue suggests that the data distribution encountered during deployment differs from the one used during training, a phenomenon known as distribution shift.

  To better understand and address this problem, a new test dataset, $\mathcal{D}_{\text{test2}}$, has been created to evaluate model performance under these changing conditions. Your mission includes the following tasks:

  1. Diagnose the Problem: Investigate potential reasons for the observed performance drop. Do you think there is a distribution shift issue occurring? If so, what type of distribution shift is it?
  2. Discuss why traditional machine learning models trained under one distribution may fail when applied to data from a different distribution. Explore Techniques for Handling the distribution shift in this particular problem.
  3. Read about and implement methods to detect and mitigate distribution shifts. Summarize key techniques and explain how they could be applied in this scenario to improve model generalization. For background reading, you may refer to this resource: Distribution shift and defenses. You are strongly encouraged to look further into the literature on this topic as part of the project.

Overview of Guidelines

- The deadline to submit the report, code and presentation is 5pm April 28th.
- You must complete this work in a group of 4-5, and this group must be declared on Moodle under Group Project Member Selection.
- Submission will be via the Moodle page. Only one student in the group needs to make a submission.
- The project will contribute 20% of your final grade for the course.

- Recall the guidance regarding plagiarism in the course introduction: this applies to all aspects of this project as well, and if evidence of plagiarism is detected it may result in penalties ranging from loss of marks to suspension.
- Late submissions will incur a penalty of 5% per day **from the maximum achievable grade**. For example, if you achieve a grade of 80/100 but you submitted 3 days late, then your final grade will be $80 - 3 \times 5 = 65$. Submissions that are more than 5 days late will receive a mark of zero. The late penalty applies to all group members.
- All group members must submit a peer-review survey by 5pm 2nd May. **Failure to complete the survey will result in a 10% penalty to that student.**

Objectives

In this project, your group will use what they have learned in COMP9417 to construct a predictive model for the specific task described above, as well as write a detailed report outlining your exploration of the data and approach to modeling. The report is expected to be a maximum of 6 pages long (12 pt font size with a single column, 1.5 line spacing), and easy to read. The body of the report should contain the main parts of the presentation, and any supplementary material should be deferred to the appendix. For example, only include a plot if it is important to get your message across. The guidelines for the report are as follows:

1. Title Page: tile of the project, name of the group and all group members (names and zIDs), and link to OneDrive folder containing video presentation. The title page is not counted in the page count.
2. Introduction: a brief summary of the task, the main issues for the task and a short description of how you approached these issues.
3. Exploratory Data Analysis and Literature review: this is a crucial aspect of this project and should be done carefully given the lack of domain information. Some (potential) questions for consideration: are all features relevant? What is the distribution of the targets? What are the relationships between the features? What are the relationships between the targets? How has this sort of task been approached in the literature? etc.
4. Methodology: A detailed explanation and justification of methods developed, method selection, feature selection, hyper-parameter tuning, evaluation metrics, design choices, etc. State which method has been selected for the final test and its hyper-parameters.
5. Results: Include the results achieved by the different models implemented in your work, with a focus on the f1 score. Be sure to explain how each of the models was trained, and how you chose your final model.
6. Discussion: Compare different models, their features and their performance. What insights have you gained?
7. Conclusion: Give a brief summary of the project and your findings, and what could be improved on if you had more time.
8. References: list of all literature that you have used in your project, if any. You are encouraged to go beyond the scope of the course content for this project. References are not counted in the page count.

You must follow this outline, and each section should be standalone. This means for example, that you should not display results in your methodology section.

Project implementation

Each group must implement a model and generate predictions for the provided test set. You are free to select the types of models, features and tune the methods for best performance as you see fit, but your approach must be outlined in sufficient detail in the report. You may also make use of any machine learning algorithm, even if it has not been covered in the course, as long as you provide an explanation of the algorithm in the report and justify why it is appropriate for the task. You can use any open-source libraries for the project, as long as they are cited in your work. You can use all the provided features or a subset of features; however, you are expected to give a justification for your choice. You may run some exploratory analysis or some feature selection techniques to select your features. There is no restriction on how you choose your features as long as you are able to justify it. In your justification of selecting methods, parameters and features you may refer to published results of similar experiments.

Video Presentation

Each team is required to submit a 2-minute video presentation that outlines the problem and the group's approach to modeling. The purpose is to provide a high-level summary of the project and highlight key insights, rather than focusing on technical details or minutiae. Please ensure that the video is exactly 2 minutes long in real time; videos played at faster speeds will incur penalties. Place your video presentation in a OneDrive folder, and include a link to this folder on the title page of your submitted report. It is your responsibility to check that the video file is not corrupted (double-check audio and video are working, and check that the link works).

Code and report submission

You should submit this on moodle under the moodle object `Group Project - Reports`. Only 1 member of the group needs to submit. Please submit the code files as a separate `.zip` file alongside the report, which must be in `.pdf` format. Your project should consist of multiple `.py` files (e.g., separate files for different models and/or specific processing steps, etc.) containing well commented and easy to ready code, with a README that provides instructions on how to run the code. While you may use Jupyter notebooks for exploratory data analysis, they should not be the primary method for running your code (e.g., you can extract .py files from Jupyter notebook). Penalties will apply if the `.pdf` file is not submitted separately (do not include the PDF within the zip file).

Predictions submission

You should submit this on moodle under the moodle object `Group Project - Predictions`. Only 1 member of the group needs to submit. You are to submit predictions for both test set 1 (the 1000 unlabelled points) and test set 2 (the 1818 unlabelled points). These predictions should be provided in a zip file containing two `.npy` files. The zip file should be named `'GROUPNAME'.zip`. The two `.npy` files should be named `preds_1.npy` and `preds_2.npy`, respectively. `preds_1.npy` must be a NumPy array of predictions for test set 1, and should be of size 1000 × 28. `preds_2.npy` must be a NumPy array of predictions for test set 2, and should be of size 1818 × 28. Failure to follow these instructions may lead to a grade of zero for your model predictive performance portion of the grading criteria. Your predictions will be evaluated using a weighted cross-entropy loss, where the weights are determined by the inverse frequency of the classes in the respective test sets. Mathematically, the weighted cross-entropy loss is given by:

$$L = -\sum_{i=1}^{N} w_{y_i} \log \hat{p}_i$$

where:

- N is the number of samples,
- y_i is the true class label of the i-th sample,
- $\hat{p}_i$ is the predicted probability for the true class,
- w_{y_i} is the weight for class y_i, typically defined as $w_{y_i} = \frac{1}{f_{y_i}}$, where f_{y_i} is the frequency of class y_i in the test set.

This formulation ensures that less frequent classes receive higher weights, helping to mitigate class imbalance issues. For your benefit, the following code snippet is provided to ensure your submission meets the requirements and to provide familiarity with the loss function.

```python
import numpy as np
import pandas as pd
import zipfile

# open zip file containing preds_1.npy and preds_2.npy
with zipfile.ZipFile('GROUPNAME.zip', 'r') as zip_ref:
    zip_ref.extractall('extracted_files')  # Extract all files into the 'extracted_files' folder

preds_1 = np.load('extracted_files/preds_1.npy')
preds_2 = np.load('extracted_files/preds_2.npy')

# Check if preds_1 is of size 1000x28 and preds_2 is of size 1818x28
if preds_1.shape != (1000, 28):
    raise ValueError(f"preds_1 has size {preds_1.shape}, but expected 1000x28")

if preds_2.shape != (1818, 28):
    raise ValueError(f"preds_2 has size {preds_2.shape}, but expected 1818x28")

def weighted_log_loss(y_true, y_pred):
    """
    Compute the weighted cross-entropy (log loss) given true labels and predicted probabilities.

    Parameters:
    - y_true: (N, C) One-hot encoded true labels
    - y_pred: (N, C) Predicted probabilities

    Returns:
    - Weighted log loss (scalar).
    """

    # Compute class frequencies
    class_counts = np.sum(y_true, axis=0)  # Sum over samples to get counts per class
    class_weights = 1.0 / class_counts
    class_weights /= np.sum(class_weights)  # Normalize weights to sum to 1

    # Compute weighted loss
    sample_weights = np.sum(y_true * class_weights, axis=1)  # Get weight for each sample
    loss = -np.mean(sample_weights * np.sum(y_true * np.log(y_pred), axis=1))

    return loss

# y_test_1_ohe is the one hot encoded array of true labels in test set 1
# y_test_2_ohe is the one hot encoded array of true labels in test set 2
# you do not have access to either, here are RANDOMLY generated ohe labels to ensure code runs
y_test_1_ohe = (np.arange(28) == np.random.choice(28, size=1000)[:, None]).astype(int)
y_test_2_ohe = (np.arange(28) == np.random.choice(28, size=1818)[:, None]).astype(int)

loss_1 = weighted_log_loss(y_test_1_ohe, preds_1)
loss_2 = weighted_log_loss(y_test_2_ohe, preds_2)
```

Peer review

Individual contributions to the project will be assessed through a peer-review process, which will be announced later, after the reports are submitted. This will be used to scale the mark based on contribution, and 80% of the final group project mark will be weighted based on individual contributions. Anyone who does not complete the peer review by the 5pm 2nd May will be deemed to have not contributed to the assignment. Peer review is a confidential process and group members are not allowed to disclose their review to their peers.

Project help

Consult Python package online documentation for using methods, metrics and scores. There are many other resources on the Internet and in the classification literature. When using these resources, please keep in mind the guidance regarding plagiarism in the course introduction. General questions regarding group project should be posted in the Group project forum in the course Moodle page.