Lab 3: Machine Learning and Networks

Week 5

Tod Van Gunten, Clare Llewellyn

Feb 2026

1 Introduction

In this lab, we will again work with a dataset based on social media posts (tweets) by members of UK parliament and other public figures/organisations on Twitter (now X). These data correspond to the years 2017-2019, a period of Brexit negotiation and leading up to the 2019 general election.

1.1 Machine Learning

In the first half of the lab, you will be using machine learning to categorise social media users as to the type of political conversation they are involved in as defined by a political party label. This will be extended in the second part of the lab on networks.

You will download a data set that has already been produced and partially prepared. It contains a list of users identified by their social media username. These are all public figures and organisations (MPs, political parties, media outlets). In the data table you will see details of how many times they have been reposted by other political parties. Some of these users are Members of the UK Parliament; if they are a member of parliament, then we know that those users exist in the conversational space of their party. This has been used to assign a party value label. We want to use these labels to predict what part of the conversation the non-labelled users are in. We are going to look at using a decision tree to do this. We will be following the process of preparing the data, splitting the data (into train and test), training the model, testing and scoring the model, evaluating against a baseline, and then running the model on new data.

1.1.1 Using KNIME

- Open KNIME.
- If required go to preferences (in the top right-hand corner) and perform an update and Install extensions (analytics platform).

- Each node may have arrows coming into the node on the left or leaving the node on the right. Throughout this document we will call these input ports and output ports.

1.1.2 Uploading the Data

- Download the data file (lab_retweet_ML.csv) from GitHub
- Start a new project called LAB3ML. Remember to save this file periodically.
- Pick the CSV Reader node. Do this by clicking on nodes on the far left of the screen. Search for CSV Reader. Drag this node into the central space. This will read in our data file, it is a comma separated value file (CSV).
- Open the dialog by clicking on the node, it the right-hand window you will see Open Dialog, click this.
- Find the correct file by clicking the button Browse which appears next to the File box (this should be wherever you have downloaded lab_retweet ML.csv to). Click apply to upload the data (figure 1).
- In the bottom window look at the table and the statistics. You will see a list of public figures and organisations (MPs, political parties, media outlets). You can Google some of these names and it will show you who the figure is). Each user has made a social media post that has been reposted. There are counts of how many times it has been retweeted by different political parties.

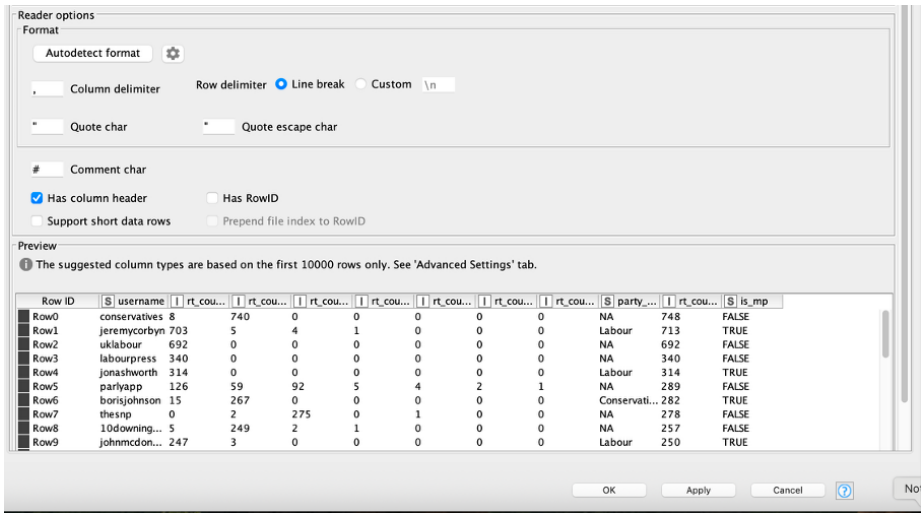

Figure 1: Reading in the Data

1.1.3 Visualising the Data

In the data there is a column called is_mp which has either a value of TRUE or FALSE. If the account was an MP at the time the data was gathered, you have the value TRUE otherwise the value is FALSE. We are going to visualise this column.

- Add in the node Pie Chart (you can search for this in the nodes window on the left). Link the output port from CSV Reader to input port of Pie Chart (figure 2).

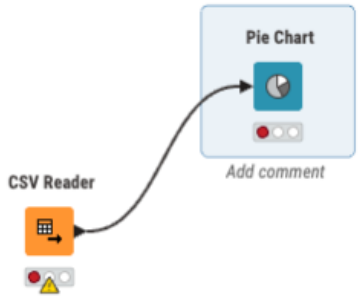

Figure 2: Connecting the CSV Reader to the Pie Chart

- Open the dialog window for the Pie Chart. Set the category dimension to is_mp and the aggregation to Occurrence count. Apply and Execute this node. You will see a pie chart in the bottom window (figure 3).

Figure 3: The Pie Chart

- We can also visualise the party of each user. Add in a Bar chart node (search in the nodes window on the left). Connect the import port to

the output port of CSV Reader. Open the dialog box. Set the category dimension to party_value and the Aggregation to Occurrence count. Click Apply and Execute to see the graph (figure 4).

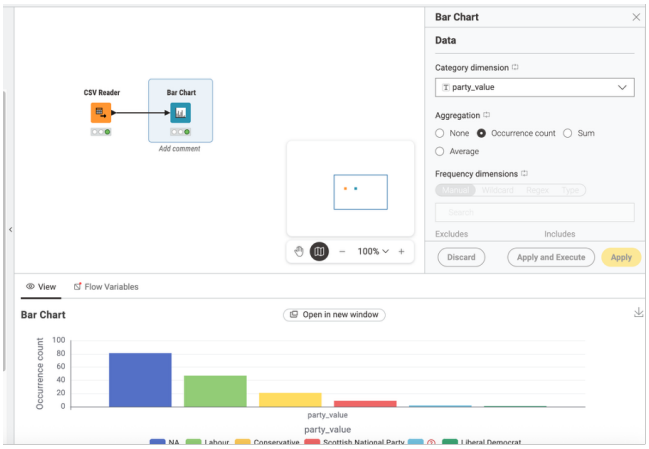

Figure 4: The BarChart

- You can try looking at more general statistics. Add in the Statistics node. Join the input port to the CSV Reader. Apply and execute. Look at the table and the statistics.
- Open the Dialog Box for the statistics node. Adjust the options using the arrows in the centre to match the figure 5).

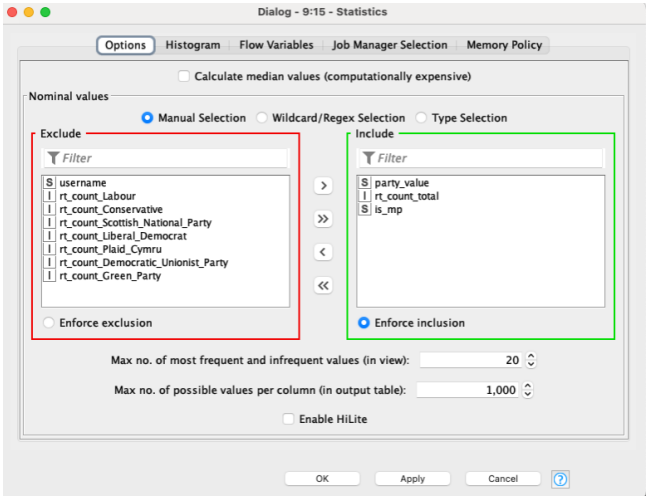

Figure 5: Setting up the Statistics Node

Apply and execute. What do you notice about the statistics of the smaller classes (the ones with very few occurrences of that label)? What affect do you think this will have on training and testing the model?

1.1.4 Splitting the Data to TRAIN and TEST

We are going to split the data so we can train and test it. 80% will be for training and 20% for testing.

- Add in a Row Filter node to find only the MPs. Connect the input port to the CSV Reader. Set the filter column to is_mp and the Value to TRUE.
- Add in a Table Partitioner. Link the input port to the output port of the Row Filter. Set the relative size to 80 (this will give an 80:20 split in train and test).
- Select a sampling strategy of Stratified (this will make sure the smaller classes appear in the test sample). Apply and Execute.

1.1.5 Training the Model

We train the model on 80% of the data so it can learn the rules.

- Add in the Decision Tree Learner node. Link the output port from the top of the Table Partitioner to the Decision Tree Learner (this is the 80% port). Open the dialog box. Set the Class column to party_value. Apply and execute.

1.1.6 Testing the Model

We are testing on 20% ofn the data so we can see how well our model has learned.

- Add in a decision Tree Predictor node. Link the blue box port input port to the blue box output port for Decision Tree Learner and the and the arrow to the bottom arrow of Table Partitioner (the 20%). Apply and execute. Look at the table in the bottom panel. Is it that you would expect?

1.1.7 Evaluating and Visualising the Results

We want to know how well our model has learned. We do this my looking at an evaluation metric. Here we use F-measure (this also can be called F-score or F1 in KNIME, it is called F-measure). We visualise the results so we can see if anything looks odd, this may point to any errors.

- Add in a Scorer node and link to the Decision Tree Predictor. Open the Dialog box. Set the First Column to party_value and the Second Column to Prediction (party_value). Apply and execute.

- Look at both the confusion matrix and the accuracy statistics. You will be able to see that the decision tree has performed perfectly and has a F-measure of 1. Can you see precision and recall numbers? Look back at the lecture notes and you can see how they and the F-measure are calculated.
- You can visualise the results using the Scatter Plot node.

1.1.8 Creating a Baseline

We want to create a simple baseline to see if we have improved from a simple model by using machine learning. Here we are using a majority baseline, this is a commonly used baseline. We are comparing if the machine learning is better than just assigning every user to the largest class, which is Labour.

- Add in a Constant Value Column Appender. Connect the input port to the bottom output port of the Table Partitioner.
- Use the Dialog box to Append a column. Call this new column Majority Class. The column type is String and the Custom value is Labour (this is the most common label in our set). Apply and execute.
- Add in a scorer node. Connect it to the output port of the Constant Value Column Appender. Open the Dialog box. Set the First Column to party_value and the Second Column as Majority_class. Apply and execute. Look at the Accuracy statistics. What is the F-measure here?

1.1.9 Predicting on Unlabelled Data

We now want to use our machine learning. We know it has worked well on our test set so we will apply it to the data that we did not have any labels for.

- Add in another Decision Tree Predictor node. Connect the input port to the Row Filter node that you used to filter is_mp as FALSE. Connect the blue box input to the Decision Tree Learner. Check that it is predicting 'party_value'. Apply and execute.
- Look at the results. You can manually select a sample of the name and use resources like Wikipedia to see if you think the correct prediction has been made. This process is called human validation.

1.1.10 Further Questions

- Do you think the predictions are appropriate? Consider the problem formulation of this task and what issues there may be with the data. Look at the lecture notes. Is this an appropriate sample to train on? If you were to redesign the study, what might you do differently?

- Think about the ethics of this process. Is it an ethically appropriate study? Here we have used public figures. Do you think we could use this on the public? How would you feel if your social media account had been included in this set?

1.1.11 Stretch Task

Can you use another ML algorithm for this task? Try using KNIME to create another workflow, for example you could use a Random Forrest Predictor.

1.2 Network analysis

We will now use another dataset derived from MP's political conversations on Twitter. This dataset contains retweets of MPs and other public official/organisation accounts in the second quarter (April-June) of 2019. These data represent a retweet network. A tie/connection/link in this network means that one account retweeted another at least once during this period. A question we can ask about with this dataset is: was UK political twitter polarised during this period?

- First,download the data (public-retweet-net-2019-02.graphml) from Github.
- Import the data into Gephi Lite (`lite.gephi.org/`).

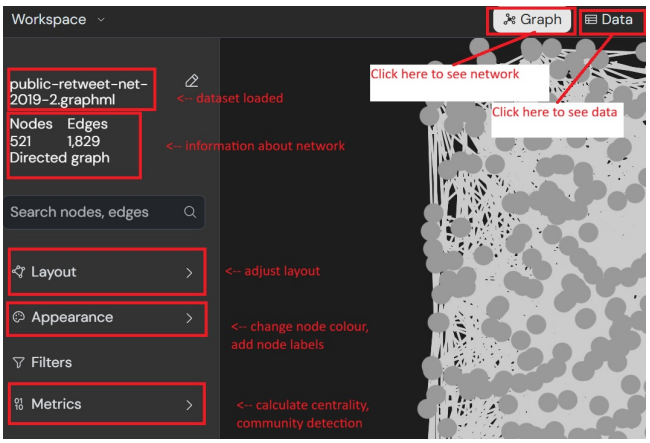

Figure 6: The Gephi lite interface

- Set the node color to political party (Appearance → nodes → color → set color to party_color.
- Adjust the layout: in the layout menu, try different options, making sure to try the options force directed and ForceAtlas2.

- Add node labels to the network (Appearance → labels → Set label from → chose name). Using the scrollover function, see which MPs you can identify.
- Change the node color to the predicted political party: this adds learned party labels from the ML portion of the lab. What differences do you see
- Calculate degree centrality (Metrics → degree → choose "Outgoing degree" and change the degree attribute name to something like "out_degree". Click 'compute metric'
  - What does this metric capture?
  - Go to the data view, and click on the degree column to sort from high to low. Who are the most retweeted accounts? Do they belong to MPs?
  - Go back to the graph view and scale nodes to degree (Appearance → Nodes → Size, set size from out degree. Explore the network and identify the high-degree nodes you spotted earlier.
- Other centrality metrics
  - Repeat the steps above to calculate pagerank centrality. Which accounts are most central on this metric? How can you interpret this centrality measure in this context.
  - Repeat the steps above to calculate betweenness centrality. Which accounts are most central on this metric? HOw can you interpret this centrality measure in this context.
- Louvain community detection: identify subgroups based on the network structure. How closely do network-based subgroups/communities conform to political parties? Try lowering the resolution parameter to get a better correspondence to the party labels.

1.2.1 Stretch task

Repeat the analysis using Gephi desktop. Import the data files rt-public-el-q-2019-2.csv and mp_party.csv into Gephi, rather than using the graphml file.

- To import network data in edgelist format → File → Import Spreadsheet → select rt-public-el-q-2019-2.csv → Next: → Finish: →Leave 'New workspace' selected. NOTE: the imported file **must** have the columns 'Source' and 'Target'
- To import node attributes: File: Import spreadsheet → select mp_party.csv → (Leave default → Import as → Nodes table) → Next → Finish → IMPORTANT: Change radio button to 'append to existing workspace.' NOTE: the attributes file **must** contain an 'id' column corresponding to (in this case) username.

- Repeat the steps above, bearing in mind that options are in a different location in Gephi desktop

1.2.2 Extra stretch task

Extend the ML exercise above by creating a new 'predictor' (for either Random Forest or Decision Tree methods) that predicts party labels for all accounts (including MPs). Connect this to a 'csv writer' to export the data. Now add this data file to Gephi using the same method for importing node attributes above. (Hint: to do this, you need to rename the 'username' column to 'id'). Ensure that you can visualise the predicted party labels for non-MP accounts in the retweet network.