# X Community Notes - Further analysis

Following the insights gained during the initial exploratory data analysis described in the Initial Report, this document will set out a plan for more in depth analysis along 2 main avenues of enquiry:

1. **Improving the performance of the Multi-label classifier predicting harmful election-related content**: The pilot investigation described in the Initial Report used a multi-label classifier to predict the harmfulness level of the original tweets linked to individual CNs. Whilst the achieved performance using raw features from the Notes table was sufficient to make predictions at the population trend level, it may be possible to improve the performance further. During this two week sprint, one team member can investigate the potential of:

   (a) Feature engineering: The raw features (user-contributed tags on the type of content, e.g. misleading, factually inaccurate) are binary True/False. However they do fall into two broad categories: indicators against harm (e.g. factually accurate, clearly satire), and indicators of potential harm (e.g. manipulated media, unverified claims). These flags could potentially be used together to create features that may be more useful for prediction, for example, calculating a bipolar score from -1 (a note with ALL the potential harm flags) to +1 (a note with ALL the against harm flags).

   (b) Model tuning: The current model selection was based on the suitability for neural network models for binary input data, and used an 'out-of-the-box' model from Scikit-Learn. Additional performance gains may be possible by hyperparameter tuning and/or changes to the model structure (e.g. number/arrangement of hidden layers).

   As the results of the feature engineering stage of this line of work may result in qualitatively different input features, it is possible that models other than a Neural Network may be desirable. Therefore it is recommended that these two tasks be completed sequentially, with the precise nature of the model tuning task depending on any changes to input features made as a result of feature engineering.

   **Potential insights: greater ability to predict potentially harmful content related to elections at the level of single notes/tweets.**

   **Packages used: Scikit-Learn, Twitter API**

2. **Building an ontology of election misinformation**: By using the tweet IDs in the CN dataset alongside the ability to predict potentially harmful election related tweets, we can use the Twitter API to investigate the harmful election-related tweets themselves, as well as the CNs linked to them. By running named entity recognition (NER) on this text data, we may be able to start constructing an ontology/knowledge graph of the persons, accounts, organisations and ideologies involved in potentially harmful election-related discourse:

   (a) Refinement of unsupervised NER (e.g. additional entity types such as organisations and twitter accounts, fuzzy matching/deduplication of extracted entities)

   (b) Construction of heterogenous graphs w.r.t. to node *and* edges (e.g. edges able to represent different relationships between nodes, such as common membership of an organisation, political rivalry)

   (c) Develop code for network querying of important actors/communities: Use of degree/betweenness centrality analysis to find central nodes and connected entities.

   (d) Sentiment analysis of harmful election related content related to specific persons/organisations/communities.

   (e) Migration of graph networks to database (Neo4J) to enable efficient longer term storage and querying of graph properties.

   **Potential insights: An election misinformation knowledge graph would allow querying of new persons/entities of interest, and allow them to be located within social/ideological networks. In the case of 'person' nodes this could illuminate prominent subjects of misinformation, or when done on e.g. twitter accounts linked to harmful posts this could illuminate whether misinformation is being spread in a centralised or decentralised manner. Derived metrics (e.g. calculated negative sentiment) could be summed/averaged over sub-networks**

   **Packages used: NetworkX/Neo4J, regex, spacy/NLTK**