Optimizing the ALMA Research Proposal Process with Machine Learning

Arnav Boppudi*†, Ryan Lipps*†, Noah McIntire*†, Kaleigh O'Hara*†, Brendan Puglisi*† Antonios Mamalakis*
*School of Data Science
University of Virginia, Charlottesville, Virginia
Email: npa4tg@virginia.edu
†These authors contributed equally to this work.

Abstract— **Every year, astronomers from around the world submit research proposals to the Atacama Large Millimeter Array (ALMA), the largest radio telescope array in the world. The aim of the current work is to streamline the proposal process for astronomers submitting projects to ALMA by suggesting frequency ranges that may be relevant to their research based on their proposal text. We introduce a pipeline of supervised and unsupervised machine learning models, each using various representations of the title and abstract of an incoming proposal. First, a logistic regression filters out proposed projects that are not expected to need specific technical setups. Second, if a technical setup is deemed necessary, our pipeline assigns an incoming project to one of 50 "similar project" groups, defined by topics generated from Latent Dirichlet Allocation (LDA). Third, we apply Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) to mine patterns in measurements ("areas of interest") made in previous projects, for each one of the 50 "similar project" groups. In parallel to the aforementioned topic modeling and HDBSCAN mining, we employ a Multinomial Naive Bayes classifier to predict the broad frequency range defined by the technical limitations of ALMA (frequency band) that we expect a project to make measurements in. Finally, we offer researchers a list of the mined "areas of interest" filtered by the predictions of the Multinomial Naive Bayes classifier. Ultimately, given a proposed project title and abstract, our pipeline generates several recommended "areas of interest" that one should consider measuring in.**

Regarding the performance of our models, we find that 67.17% of test projects match at least one of the recommended "areas of interest", with an average hit rate of 44.72% across measurements within each test project, when limiting to the top two band predictions. When we disregard band predictions, 88.81% of test projects match at least one recommended "area of interest" with an average hit rate of 60.00% across measurements within each test project. While the latter approach gives better results in terms of hit rates, we believe the combination of models provides a good balance of accuracy and precision.

Index Terms—**Modelling**

I. Introduction

The ALMA Observatory is a radio telescope array located in the Atacama Desert in Chile. ALMA is the largest astronomical project in existence and is open to anyone in the world to use via a project proposal process. Each submitted project proposal requires a technical plan outlining the specific settings and measurements the research team wishes to make. This can be daunting since the design of the telescope array allows ALMA to detect electromagnetic radiation on a fairly continuous range of frequencies from 35 GHz to 950 GHz at two-decimal-point precision. This continuous range of frequencies is broken up into ten discrete bands. Furthermore, the technical specifications of the telescope allow individual measurements to span a range of approximately 4 GHz. On top of the technical nature of project proposals, ALMA can only accept around 400 proposals to enact throughout the year, making it a valuable and competitive resource. Thus, it is paramount that researchers submit well designed and technically proficient proposals to increase their chances of acceptance.

Our work streamlines the ALMA proposal process by providing researchers insight and assistance on how to set their measurement requirements in the project proposal technical plan, with the intent of optimizing the insight they can extract from the data.

II. Methodology

Our process involves two subtasks — one to classify the type of project based on expected technical plan, the second to provide recommendations for a possible frequency to observe, based on information from task one.

The first subtask in our work classifies proposed projects as either "line" or "continuum" setups. For ALMA's technical specifications, "line" setups focus on detecting specific frequencies associated with spectral lines of molecules, revealing chemical compositions and physical conditions of celestial objects. "Continuum" setups measure broad-spectrum emissions to understand the energy output, temperature, and structure of astronomical bodies. Because continuum setups are often used for specific projects and can cover a wide range of frequencies in a single observation, we work solely with line setups when providing recommendations.

The second, more extensive task of this approach, involves creating an interpretable suite of models that assist in recommending frequency ranges for line setups, based on a project's title and abstract. This provides stakeholders with a tool to explore similar accepted projects based on the perceived topic of their proposal and the related frequencies

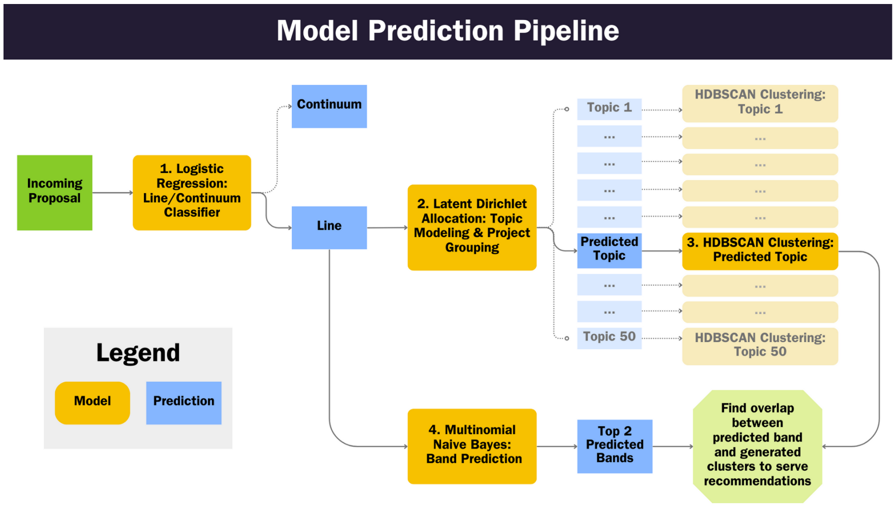
Model Prediction Pipeline
Incoming Proposal
1. Logistic Regression: Line/Continuum Classifier
Continuum
Line
2. Latent Dirichlet Allocation: Topic Modeling & Project Grouping
Topic 1
...
Predicted Topic
Topic 50
HDBSCAN Clustering: Topic 1
3. HDBSCAN Clustering: Predicted Topic
HDBSCAN Clustering: Topic 50
Legend
Model
Prediction
4. Multinomial Naive Bayes: Band Prediction
Top 2 Predicted Bands
Find overlap between predicted band and generated clusters to serve recommendations

Fig. 1: Full model pipeline. Our final model recommends frequency ranges for project proposals by using a Logistic Regression model and combining LDA and HDBSCAN with a Multinomial Naive Bayes Classification model.

that may be useful to observe. The full model prediction pipeline is shown in Fig. 1.

III. Data Discussion

The full dataset consists of 4,586 accepted projects, each with a title and abstract that we concatenated as input for our models. These combined texts vary from 16 to 560 characters in length. Each project may consist of numerous measurements at different frequencies. 3,628 of the accepted projects are “line” projects that account for 44,232 unique measurements. Fig. 2 shows the distribution of measurement count per project.

In Fig 2., we see outliers in the distribution of the number of measurements across projects. These outlier projects can often be attributed to specific uses of ALMA that are outside the scope and scale of our process. Therefore, we removed these projects to prevent them from having too much influence in model training. We identified outliers as projects with more than 26 measurements (3rd Quartile + 1.5 * Interquartile Range).

Measurements in the raw data are defined by a frequency lower-bound and upper-bound. From these, we engineered a median frequency, which is the mid-point between the high and low frequency for each measurement. This serves as a simple way to define measurements by a single value, with the assumption that a measurement is defined by its center, as opposed to its range.

We explored the median frequencies and counted the number of bands in each project and found that 81.54% of projects only have measurements within one band. Nonetheless, one project has measurements in 6 different bands. So that such projects do not disproportionately influence the predicted frequency ranges, we removed projects that have measurements in more than 2 bands. Additionally, we removed projects that have incorrectly formatted band information. It is worth noting that there are no measurements in band 2.

These cleaning efforts reduced the data to 3,178 projects and 23,482 measurements. Finally, we used 80% of the projects (2,383 projects with 17,638 measurements) to train our models, and 20% of the data (795 projects with 5,844 measurements) to test our models.

IV. Text Preprocessing Methods

In our work we applied various text preprocessing methods, outlined below.

- **Text Standardization:** Setting all text to lower case and removing any punctuation and non-alphanumeric characters.
- **Stop Word Removal:** Stop words are words that occur frequently in the English language but do not provide any significant meaning, for example, “the” and “and.” Additionally, we extended the stop word list to words that appear frequently across project proposals that do not provide significant meaning in the specific context of our work, such as “ALMA”.
- **Text Lemmatization:** Reduces words to their inflected forms. For example, “running”, “runs”, and “ran”, become “run.” This has the effect of attributing all sentiment from a word’s various representations into a single core representation of a word, allowing analysis to be more discriminant.

V. Text Vectorization

We vectorized the texts using Term Frequency Inverse Document Frequency (TF-IDF) as input for the Logistic Regression. This translates the words into a numerical value based on their frequency. Equation (1) is the TF-IDF function we used.

$$\text{TF-IDF}(q_i, p_a) = \frac{q_{ia}}{w_a} * [log\left(\frac{N+1}{u_i+1}\right) + 1], \qquad (1)$$

where q_i is the given word for $i = 1, 2, \ldots,$ the total number of words in p_a. p_a is the project text for $a = 1, 2, \ldots,$ the total number of project title and abstract texts. Furthermore, q_{ia} indicates the number of the times q_i appears in p_a, and w_a indicates the number of words in p_a. N is the total number of project title and abstract texts, and u_i is the total number of occurrences of q_i in N project title and abstract texts. [1]

All in all, TF-IDF computes the importance of a word to its text and creates a vector with these computations. This vector highlights terms that are frequent in a document but rare across the collection of documents and can be used in various machine learning models.

VI. Methods

A. Logistic Regression

To predict frequency set ups for incoming projects, we first classify them as "line" or "continuum", using Binary Logistic Regression, which we outline in Equation (2) below. It is important to note that a single project can consist of both line and continuum measurements. To account for this, we categorized a project as line if it had at least one line measurement. For this model, the input text is processed using the methods outlined in "Text Standardization". Equation (2) is this binary logistic regression model.

$$\pi(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}, \qquad (2)$$

where π is the probability that a project is a line, X is the TF-IDF vectorization of the processed text, and β_0 and β_1 are regression coefficients. [2]

For any project, p in the test set, we classified p as a line project if $\pi_p > 0.5$. Otherwise, we classified the project as a continuum project. When assessed with the testing data, this approach had an accuracy of 90.02%. Table I is a confusion matrix that specifically displays the correct and incorrect model predictions for both continuum and line projects.

Table I indicates that the dataset is heavily imbalanced and has significantly more line than continuum projects. Despite this imbalance, the logistic regression model is able to correctly predict 59.42% of continuum projects and 96.41% of line projects.

Table II further demonstrates that our model is more accurate at classifying line projects, with higher recall and precision for line projects than continuum projects. We prefer the model to have higher accuracy for classifying line projects since line projects are exclusively used for the rest of our modeling. Furthermore, the corresponding Receiver Operating Characteristic (ROC) curve for this model is shown in Fig. 6 in the appendix.

TABLE I: Confusion Matrix of Classification of Line and Continuum Projects

	Predicted Continuum	Predicted Line
True Continuum	104	71
True Line	26	699

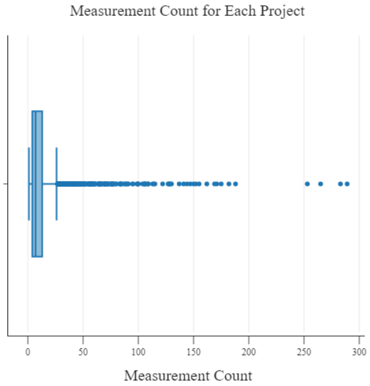

Fig. 2: Measurement Count for Each Project. 75% of projects have 13 measurements or less.

B. Topic Modeling and Project Grouping

Following the project type prediction, we employed Latent Dirichlet Allocation (LDA) to place similar projects into groups. LDA is an unsupervised machine learning algorithm that models topics in a corpus as a probabilistic mixture of the words in the corpus vocabulary. Subsequently, the algorithm generates a vector for each document that gives the weight of each topic within that document. Equation (3) outlines this process.

$$P(t_b|p_a) = \frac{P(r_w|t_b)}{P(r_w \cap t_b)}, \qquad (3)$$

where t is a topic for $b = 1, 2, \ldots,$ total number of topics, p_a is the given text for $a = 1, 2, \ldots$ total number of texts, and r_w is a given word in p_a for $w = 1, 2, \ldots,$ total number of words in p_a. [3]

The weight, $P(t_b|p_a)$, indicates the probability a project "belongs" to topic t_b. For our purposes, we assigned a project to the topic for which it has the highest weight. In this way, we attempted to surface project similarity by grouping them under the topic they most strongly align with. For example, the top 10 words most strongly associated with topic 25 are: *bar, gmcs, molecular, spiral, galaxy, arm, giant, gas, study, galactic*.

We chose to generate 50 topics, as this is large enough to effectively distinguish topics, but small enough that there

TABLE II: Precision and Recall of Classification of Line and Continuum Projects

	Precision	Recall
Continuum	0.85	0.78
Line	0.89	0.90

are an adequate number of projects in each topic. This brings us to an important assumptions in our approach: that the generated topics are salient and discriminant, which is difficult to measure. Additionally, we assume the grouped projects contain similar measurements, which we can mine to generate recommendations for the technical setup. To offset these assumptions, we provide researchers with a list of the most heavily weighted words for both their project's predicted topic and each of the other 49 generated topics.

This provides researchers insight into our model's interpretation of their project proposal and gives them the opportunity to explore recommendations for topics they believe align with their intent.

C. *Clustering Measurements Within Topics*

For each topic generated by LDA, we cluster the measurements of the grouped projects using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [4]. HDBSCAN is an extension of Density-Based Spatial Clustering of Applications with Noise (DBSCAN), an unsupervised machine learning algorithm that generates estimated clusters in the data based on the density of the data. DBSCAN is primarily parameterized by a reachability distance, ϵ, around each point and the minimum number of points (*min_cluster_size*) within ϵ that quantifies "dense". While DBSCAN is a useful starting point for generating clusters within topics, the static value of ϵ can generate clusters that are too large to be of value to a researcher submitting a proposal, due to the wide distribution of measurements. In contrast, HDBSCAN adjusts ϵ to account for varying densities across the input data. This is ideal for our purposes as it generates clusters that are representative of the overall distribution of measurements within a topic while highlighting areas where many measurements are made.

We used the same HDBSCAN parametrization for each topic, with a *min_cluster_size* of 5 measurements to generate a cluster. Measurements that are not matched to a cluster are labeled "-1" to indicate noise. In Fig. 3, we display the HDBSCAN generated clusters for the measurements in topic 25.

As HDBSCAN is an unsupervised approach, we evaluated the "health" of measurement clusters within a topic using a combination of noise proportions and cluster width. In general, we sought to parameterize HDBSCAN to ensure that the percentage of noise was no greater than 30%. [5]. On average, the noise percentage within a topic is 14.59% $\pm$ 0.05%. Furthermore, we inspected the individual cluster widths to ensure that the generated clusters were not too wide. While the point of HDBSCAN is to adapt to areas of varying density in the data, and thus should make wider clusters in areas of low density, the ultimate goal of clustering is to mine and serve more finely tuned recommendations to researchers. As such, recommending researchers make measurements in ranges of 100 GHz is not helpful, whereas a range of 10 GHz or less is more focused and thus more informative. In the case that we cannot reduce dense and wide clusters, we offer researchers a histogram of the measurements within the cluster to drill further down.

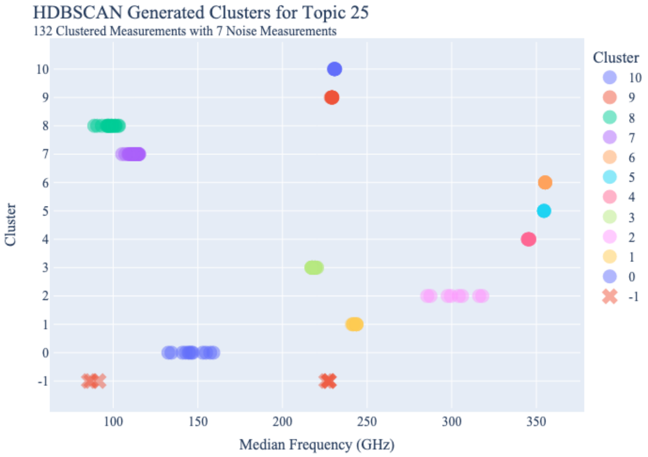

Fig. 3: Inspection of Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) Clusters for Topic 25. 18 training projects are assigned to topic 25 from the Latent Dirichlet Allocation (LDA) topic model, accounting for 139 measurements. HDBSCAN produces 12 clusters, with 7 measurements (0.5% of measurements within the cluster) identified as noise (labeled as -1). Note the different densities of cluster 1 (yellow, 241.00 - 243.87 GHz), cluster 2 (fuschia, 285.52 - 318.06 GHz), and cluster 4 (magenta, 344.96 - 345.81 GHz). These three clusters are all composed of 8 measurements, but span different ranges. This highlights the density adaptation of HDBSCAN.

Finally, we engineered "areas of interest" from the HDBSCAN clustering by taking the lowest median frequency and highest median frequency for each cluster, and counted the number of measurements and their associated projects in this range. These "areas of interest" provide measurement frequency ranges that are characteristic of the projects within that topic, and thus serve as the basis for our recommendations. For example, a measurement cluster that has 20 measurements all from one project is deemed less "interesting" than a cluster that has 10 measurements from 5 projects, as the latter has support from a larger group of projects. On average, projects in the testing data have 60.00% of their measurements falling within the recommended "areas of interest" for their predicted topic. Table III shows the hit rates for the 5 test projects predicted to match topic 25. Note that at least 65% of a given test project's measurements match a recommended "area of interest".

D. *Band Classification - Multinomial Naive Bayes*

To predict the probability a project includes measurements within a specific band, we implemented Multinomial Naive Bayes, a machine learning algorithm popular with text classification. Formally,

$$P(c|d) = \text{argmax}_{c \epsilon C} \left(P(c) \cdot \prod_{1 \leq k \leq n_d} P(t_k|c) \right), \quad (4)$$

where c is the given band $(1, 3, 4, 5, 6, 7, 8, 9, \text{ or } 10)$, d is the TF-IDF vectorized project text, n_d is the number of tokens in d, t_k is a term occurring in the vectorized project title and abstract text. [6] Since we removed outlier projects in more than two bands, the goal of this approach is to identify the two bands that are most likely to be of a wide range of interest for each project.

In Fig 4, we display the distribution of bands by measurement count in the training data. As shown in Fig. 4, the distribution of measurements within bands is not uniform. To account for this imbalance, we specified the prior probabilities of a measurement corresponding with each band using the following formula:

$$w_j = \frac{n}{(C * n_j)} \text{ for } j = 1, 3, 4, 5, 6, 7, 8, 9, \text{ or } 10, \quad (5)$$

where w_j is the weight for each band, j indicates which band, n indicates the total number of samples, C indicates the total number of bands (9), and n_j indicates the total number of instances of band j. [7]

This helps the model optimize accuracy for less common bands. We call this the *weighted* approach. Additionally, we assess how the accuracy performs without assigning weights to each band class and call this the *unweighted* approach. The *unweighted* approach causes the model to fit the data according to the percent of instances of each band in the training data.

We fit two models: one for the *weighted* approach and one for the *unweighted* approach. For both models, we processed the text using "Text Standardization", "Stop Word Removal", and "Text Lemmatization". Then, we fit each model where the TF-IDF vectorized text is the input, and the band for each measurement is the target value.

After fitting each model according to Multinomial Naive Bayes, for each project, we obtained a vector of predicted probabilities that the project corresponds with each band. We accepted the two bands with the highest probabilities for each project as the model's predicted set of bands. Under this assumption, with the testing data, the *weighted* model's set of two bands included the correct band approximately 69.70% of the time, and the *unweighted* model's set of two bands includes the correct band approximately 73.55% of the time.

TABLE III: Hit Rates for Test Projects in Topic 25

Test Project	Measurements	Hits	Hit Rate
1	4	3	0.75
2	2	2	1
3	17	11	0.65
4	7	7	1
5	17	17	1

Since these accuracies do not differ greatly, we explored how the accuracies by band, or overall model precision, compares between the *weighted* model and *unweighted* model. See Fig. 5.

The *weighted* model predicts the minority bands in some instances while the *unweighted* model does not. Therefore, the *weighted* model has higher precision than the unweighted model. We prefer the *weighted* model over the *unweighted* model despite approximately 3.85% less accuracy since it has less bias toward the majority bands. For the rest of this paper, we refer to this *weighted* model as *Band Classification (weighted).*

E. Combined Method

The HDBSCAN and *Band Classification (weighted)* methods adequately predict "areas of interest" for each project in the test set. HDBSCAN provides many diverse, typically narrow, areas of interest while *Band Classification (weighted)* provides two wide areas of interest (bands). In an attempt to maintain the narrowness of HDBSCAN's predictions while benefiting from the reduced number of "areas of interest" in *Band Classification (weighted)*'s results, we combined these two approaches. We refer to this approach as the *Combined (weighted)* approach. In this approach, we find the intersecting areas of interest in the topic predicted by HDBSCAN and the *Band Classification (weighted)* for each project. See an example in Table IV.

This approach is more precise than the HDBSCAN approach since it predicts fewer frequency ranges for each project. Furthermore, this approach is also more precise than the *Band Classification (weighted)* approaches since it predicts narrower area of interests.

We found that across the testing data, the *Combined (weighted)* approach predicts at least one correct measurement for 67.17% of the test set projects with an average of 44.72% of the measurements in a test project captured in the prediction.

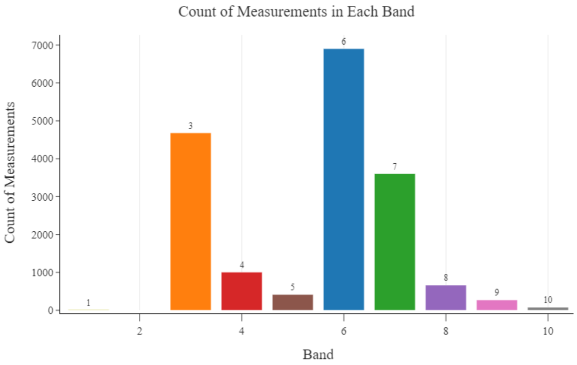

Fig. 4: Count of Measurements in Each Band. Bands 3, 6, and 7 are notably the most common bands across all measurements in the training data. Meanwhile, there are little to no measurements in bands 1 and 10. Recall there are no measurements in band 2.

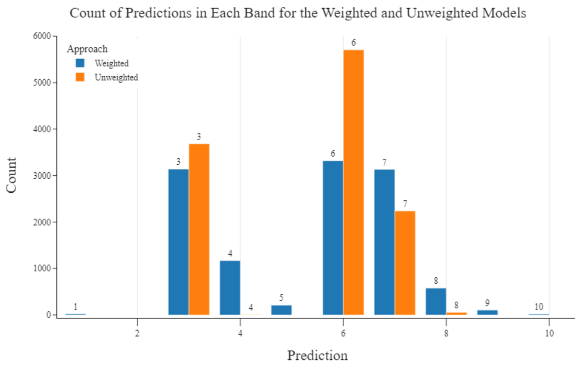

Fig. 5: Count of Predictions in Each Band for the Weighted and Unweighted Models. The *weighted* model predicts more various bands than the unweighted model. In fact, the *unweighted* model only predicts bands 3, 6, 7, and 8 while the *weighted* model predicts all bands at least once. Recall there are no measurements in band 2.

TABLE IV: Combined Predictions for Project 2017.1.00786.S

HDBSCAN Prediction Band: [Frequency Range]	**Band Classification** Bands	**Combined (unweighted) Prediction** Band: [Frequency Range]
3: [89.105, 101.005]	6	6: [213.095, 220.395]
3: [109.775, 115.160]	7	6: [227.095, 231.490]
6: [213.095, 220.395]		7: [335.090, 357.225]
6: [227.095, 231.490]		7: [344.980, 345.180]
7: [355.090, 357.225]		7: [345.785, 345.815]
7: [344.980, 345.180]		
7: [345.785, 345.815]		

VII. Future Work

Our work is adequate for recommending frequency ranges in line setups based on a proposed project title and abstract. Nevertheless, we have identified some concepts that may improve accuracy. For example, it is possible we can replace the Logistic Regression and the Multinomial Naive Bayes classification models with Bidirectional Encoder Representations from Transformers (BERT) models. Given BERT's previous success with natural language processing, we find this approach worth investigating.

To further improve the accuracy of the *Band Classification (weighted)* approach, we can investigate additional ways to fine tune the weights for the imbalanced distribution of band measurements. One possible approach is to balance the dataset by undersampling measurements in large bands (3, 6, and 7) so that they are closer to or the same as the other bands (1, 5, 8, 9, and 10). This approach comes with the cost of losing data, but nonetheless would be interesting to see how it affects the band prediction.

The *Combined (weighted)* process makes a strong assumption within the LDA topic modeling. Though we assess accuracy and hit-rates where possible in our work, there is no concrete way to measure the accuracy of the generated topics, and thus, the project similarity. Assuming that projects within a topic are similar is crucial to the relevance of the "areas of interest" generated with HDBSCAN. To ensure the topics are both salient and discriminant, we will consult subject matter experts and fine tune the text processing. In the meantime, we offer the solution of surfacing the top words associated with the topics and giving researchers the option to explore all topics and clusters.

Lastly, we assume that the all "areas of interest" already exist in the data. As the mining phase of our process does not make any predictions for "areas of interest," there will be no recommendations to measure where there are no measurements in the data. In the future we can explore methods to allow our process to extrapolate and recommend previously unmeasured "areas of interest".

Appendix

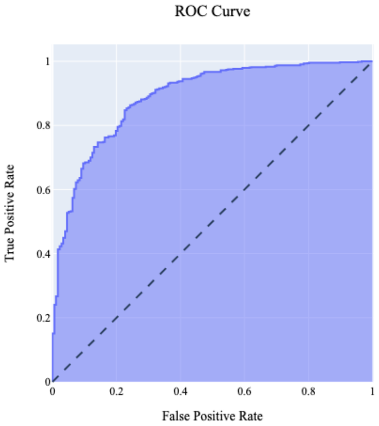

Fig. 6: ROC Curve. The Receiver operating characteristic curve shows the performance of the logistic regression model at various threshold values.

References

[1] W. Scott, "Tf-idf from scratch in python on a real-world dataset." [Online]. Available: https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a408

[2] "12.1 logistic regression." [Online]. Available: https://online.stat.psu.edu/stat462/node/207/

[3] R. Kulshrestha, "https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2."

[4] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," *Advances in Knowledge Discovery and Data Mining. PAKDD 2013. Lecture Notes in Computer Science()*, vol. 7819, no. null, pp. 16–172, 2013. [Online]. Available: https://doi.org/10.1007/978-3-642-37456-2_14

[5] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: Why and how you should (still) use dbscan," *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1–21, 2017. [Online]. Available: https://doi.org/10.1145/3068335

[6] "Naive bayes text classification." [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html

[7] K. Singh, "How to improve class imbalance using class weights in machine learning?" [Online]. Available: https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/