Predicting Misinformation on Twitter

CAPP 30254 | Trollbane | Final Report

Fabian Araneda Baltierra
University of Chicago
baltierra@uchicago.edu

Maria Gabriela Ayala
University of Chicago
mariagabrielaa@uchicago.edu

Ken Kliesner
University of Chicago
kenkliesner@uchicago.edu

Manuel Martinez
University of Chicago
manmart@uchicago.edu

Andrew Warfield
University of Chicago
awarfield@uchicago.edu

June 2, 2022

1 Introduction

Misinformation Detection (MID) in social networks is considered an emerging area of research interest, as social media has become increasingly prevalent in daily life, with a large presence of deceptive online activities misleading users. In MID machicne learning classification, there are two main approaches: linguistic-based (ie, NLP methods such as bag-of-words where term occurrences are counted within the text and n-grams, which tokenizes word sequences) and network-based (ie, using the characteristics of users and connection nodes/edges to classify the propagation of misinformation).

For our project, we are interested in understanding the reliability and challenges behind annotated training data. Inherently, most machine learning models rely on data that is tagged or manually classified by humans. As it is well known, humans are subject to bias and this bias can often be captured by machine learning algorithms.

We will be using open data from Birdwatch, Twitter's own community-driven project that seeks to classify misleading Tweets through a crowd-sourcing approach. This is a novel approach from a social media platform to distinguish misinformation with the aid of users. Unlike other annotated datasets on different subjects, there is a grey line in differentiating disinformation and misinformation from the truth. Often, this can be subjective and disputed by different schools of thought or personal views. As disinformation exponentially becomes an ever-growing phenomenon, we are interested in analyzing the role of public annotations in training models to effectively classify misinformation.

2 Dataset

We limit our analysis to a single dataset: Twitter's Birdwatch. Birdwatch is a community-driven pilot being run by Twitter that "aims to create a better informed world, by empowering people on Twitter to collaboratively add helpful notes to tweets that might be misleading."[1]. At this stage of the project, Twitter users voluntarily sign up as Contributors, which allows them to identify tweets they believe are misleading, write notes that provide context to the tweet, and rate the quality of other Contributors' notes. Aiming to leverage a crowdsourced scoring system from Contributor's annotations, this dataset is comprised of two main tables: *ratings* and *notes*. Appendix A.1 includes a preliminary data exploration.

2.1 Notes

Contributors can annotate any Tweet they consider to be misleading. This table is composed of multiple choice questions and an open field to provide an explanation for the user's categorization, including sources to back it up. The notes database contains 118,947 records (more detail Appendix A.2).

Notes that are rated as helpful appear directly on Tweets for a small, randomized set of users in the United States and can be publicly accessed and downloaded at any time. More detail on the fields available in the Notes data is provided in Appendix II.

Specifically, to distinguish helpful notes, Twitter employs a Matrix Factorization technique. An issue identified by the social media platform is that "most raters do not rate most notes - and this sparsity leads to outliers and noise in the data." To address this problem, Twitter applies regularization on the intercept terms, which captures the helpfulness of a note that is not explained by covariates such as viewpoint agreement and note embeddings.

[1] Birdwatch Website

Concretely, the intercept term captures the note's helpfulness score. For the intercept to be high, the note must be rated as helpful by Contributors from different points of view. This embedding is an application of Funk's approach in the 2006 Netflix prize recommender competition, except that here, it is used to calculate a single global helpfulness score instead of user recommendations.

2.2 Ratings

Contributors can rate notes they find most helpful. This rating system is at the core of Birdwatch's strategy to distinguish the quality of annotations. Notes highly rated by users from different perspectives are considered more trust worthy, hence they are given more visibility by the algorithm. Birdwatch also sustains a reputation model to recognize users whose contributions are consistently highly rated. The ratings database contains 740,947 records (more detail in Appendix A.3).

2.3 Summary Statistics

Through the Birdwatch Project, over 30.000 tweets have been classified by Contributors. It is worth noting that the distribution of the classification data is *very unbalanced*. Specifically, 85% of tweets in the sample have been classified as misinformed or misleading, as can be seen in Figure 1. This is indicative of selection bias, as we can expect users who sign up as Contributors to be more prone to actively classify tweets as misinformation.

The Notes data mainly contains binary variables that represent Contributors' responses from a multiple-option questionnaire. In terms of the process of tweet-classification we find that:

- 78% of Contributors' claim to have linked their classification to credible sources of information
- 67% are believed to cause potential harm if believed by many
- 25% of tweets are considered challenging to classify

What constitutes a credible news source is subjective, as is the process of annotating tweets. Further, Contributors are also asked to account for why a particular tweet is deemed as misinformation.

- 57% of tweets are reported to contain a factual error
- 56% have missing context
- 46% present unverified claims as fact
- 10% contain outdated information
- 5% of tweets reference "manipulated media"

As a visualizing exercise, we also created a word cloud of the top words used in the Notes field 2. This exercise reflects the most common topics around misinformation within the Birdwatch Project, mostly focused on COVID-19 and politics (with Trump and Biden among the most frequently used words).

As an advantage, we find that this data can be useful in terms of finding trending or controversial topics in the field of misinformation. Many of the multiple choice options, however, are highly subjective and must be taken with a grain of salt. Nonetheless, the goal of our project is not to predict misinformation itself but rather, how likely are Contributors to classify tweets as misinformation 3.

3 Method

3.1 Using Text Summary from Notes

Each note classified by a Contributor is associated with a text summary from the annotator explaining their reasoning. A detailed description on data pre-processing can be found in Appendix B.1.

We could leverage these text annotations and try to use the summary text to predict whether some arbitrary, user-emitted text will result in a user categorizing a particular tweet as "Not Misleading" or "Misinformed or Potentially Misleading". That is, the application of a classifier using the text can be extended well beyond the Birdwatch pilot, and be applied to tweets posted by all users in the platform. Whether the model is able to generalize beyond the Birdwatch pilot is a question that we will try to answer in more detail. Particularly, if we train a classifier that uses the note's text summary as the input to predict the binary label that indicates whether the tweet is "Not misleading", or "Misinformed or potentially misleading" from a user's written input regarding the tweet.

3.2 Using Credibility Scores

We created a model that used logistic regression and an engineered feature that was supposed to quantify the credibility of a note from the ratings it had gotten. This approach assumed that there would be a relationship between the ratings and the notes, such as a bias against certain types of notes by raters. Given that Birdwatch is a crowd-sourced dataset, this was a distinct possibility.

However, when we evaluated this model we found that it had essentially no predictive power, so we decided to focus our efforts on our Text Summary-based models.

4 Experiment Setup

4.1 Logistic Classifier

Our initial approach is to use a simple logistic regression classifier with two different feature engineering approaches. The first is a bag-of-words (BOW) vectorization, and the second is a term frequency-inverse document frequency (TF-IDF) vectorization. Note that the class distribution of these data is skewed. 88% of the notes are tagged as "Misinformed of Potentially Misleading", while 11% of notes are tagged as "Not Misleading". More detail on the feature engineering can be found in *Appendix B.2*.

4.2 Neural-Network (NN) Classifier

Our second approach is to use a more expressive classifier than a simple logistic regression. We decided to build a simple feed-forward neural network with an embedding layer with PyTorch. We forgo pre-trained embeddings by learning our embedding layer from our data. The preprocessing is the same as our logistic classifier, so we refer the reader to *Appendix B.2*.

Our architecture has three layers:

- Input layer: The size of our vocabulary.
- Embedding layer: The embedding layer has a size of 64. This layer projects our BOW sparse vectors into dense vectors in $\mathbb{R}^{64}$ space.
- Output layer: The output layer is simply a linear layer that combines the embeddings into the two output classes.

Our training process uses batch stochastic gradient descent, with a batch size of 64. At each batch we shuffle our data, train the classifier and compute the loss. Across the batches there is a cumulative measure of the accuracy, which is reported in the results section.

4.2.1 Long Short-Term Memory (LSTM) Model

After seeing no improvement from the implementation of a simple neural network model, we decided to see if implementing a more complex one might yield better results. To test this hypothesis we decided to use the Long short-term memory neural net model (LSTM).

The LSTM model is from the Recurrent Neural Networks field of study in Deep Learning. Its advantage is that it enables the model to learn from long sequences and also creates a numerical abstraction for long and short term memories, being able to substitute one for another, if needed.

In terms of architecture, LSTM has a gated structure which is a combination of some mathematical operations that make the information flow or be retained from that point on the computational graph. Because of that, it is able to "decide" between its long and short-term memory and output reliable predictions on sequence data. I consider a "forget" gate, "input" gate and an "output" gate (Srivastava, 2017).

Our training process utilized word sequences, but using the same text pre processing as in our feed-forward NN model.

5 Results

5.1 Logistic Classifier

After generating the new features, for both classifiers we split our dataset on a 80/20 split, where a random 80% of our observations are used for training our classifier and 20% is left out as a testing set or validation set. We then perform hyper-parameter tuning for the inverse of regularization strength using the area under the receiver operating characteristic curve (AUC ROC) as a way to select the most predictive model.

Table 1 shows training and validation scores for various metrics for both feature approaches. We can see that TF-IDF vectorization approach achieves an AUC of 0.809, slightly higher than the simple BOW approach. Figure 4 shows the ROC curve for the BOW approach and figure 5 shows the ROC curve for the TF-IDF approach. Figure 6 and figure 7 also show the largest coefficients in the model (both positive and negative) and maps them to the respective tokens in the model. We can see that they are very similar across the two feature approaches.

5.2 Neural-Network (NN) Classifier

Our Neural-Network does not outperform our simple logistic classifier. This might be entirely due to our selection of architecture, including layer functional forms. Our NN approach reaches a similar accuracy as our simple logistic regression (89.6%) on the validation set. We do not report any tables or figures relating to this classifier.

The time required to train the classifier is also significantly larger, even using an RTX 3070 Ti GPU with 6144 CUDA cores. As mentioned above, we do not believe that our base NN classifier is comparable to the simpler logistic regression classifier because there are many more details that go into determining the architecture of the NN model than does for the logistic regression classifier, and hyper-parameter tuning for NN such as number and size of layers, and feature transformation functions is very expensive.

5.3 LSTM NN Model

Our LSTM model behaves similarly to the NN classifier in terms of performance, reaching a maximum training accuracy of 92.7% and an overall test accuracy of 89.1% after 5 epochs (the recall was 88.6%). In this case the general architecture is not as customizable, in terms of selecting numbers of layers and neurons per layer as other NN approaches; we can tune our model by adjusting the size of the hidden layer (which regulates how much of the new information will be introduced on the memory) and other functioning parameters, as well as the selection of activation and loss function.

Again, the time required to train the classifier is also significantly larger

6 Closing Thoughts

6.1 What We Learned

Our first lessons have to do with the nature of the Birdwatch datasets. The notes dataset has a very high class imbalance between annotations that tag a tweet as misleading versus those that tag a tweet as not misleading. This might be reflective of the population distribution of content online, as it is not a sensible assumption to believe that most of the content online is not designed explicitly to mislead. Another potential issue with the data has to do with the way that Twitter is running the Birdwatch pilot, if a user wants to become an annotator, i.e. tag a tweet as misleading or not, anybody can sign up. This is a potential source of selection bias, since users are sorting themselves into the pilot. Another issue in the same vein is that of which tweets end up being annotated versus those that do not. Again, annotators have the freedom to annotate any tweet that they think warrants it, another sort of selection. This might also explain the class imbalance: since annotators can choose which Tweets to annotate and which ones not to, they might think that it's more important to annotate misleading Tweets versus those that they do not believe are worth annotating. A corollary of this is that most tweets that were annotated as "not misleading" are also reported as misleading by other annotators, which implies that there are no "pure" non-misleading tweets on the annotations dataset, but only those who some annotators believe to be misleading and others who don't.

Our first approach, in implementing a classification model, was to predict the annotator's classification of a tweet using the text explanation of the note. We thought that this might be useful in the case that the classifier would extrapolate well to regular twitter thread comments. However, we do not think that this might be the case, primarily because the way that people annotate the tweets and the way that people respond to tweets in threads might be very different. Although the approach might not be generalizable, we achieve modest performance using an extremely simple logistic classifier. Due to our class imbalance, accuracy is not a good measure, since even a classifier that would always predict an annotation as misleading, would be right around 85% of the time. We achieve a recall on the validation set of 0.994 and an AUC of 0.809 using a TF-IDF vectorization approach. We find that the difference between TF-IDF and classic BOW is not that significant, although TF-IDF performs marginally better.

Finally, and perhaps more importantly, we learned that the problem of classifying misinformation is considerably harder than we thought originally. Misinformation is not just factually inaccurate, but it takes into account the context of what is being said to elicit a particular interpretation from the listener, and context is largely topic-oriented in social media. Training a misinformation classifier for each separate topic might be cumbersome, but perhaps we can do this given the current state of NLP. Although massive-scale transformer architectures like BERT achieve human-like performance in text creation and summarization, that performance is not translated to all areas of NLP, particularly the area of NLP research that focuses on the validation of knowledge. There is much active research today regarding automated knowledge discovery, but there is something about establishing truth from text that eludes NLP today, much as it has done many individuals in the past.

6.2 Conclusion

Misinformation is not a new phenomenon. However, access and speed to information has been transformed by technology, and especially by social media giants. In this new and mostly incontinent online space, these companies have a responsibility to help users think more analytically and shield them from misinformation, especially when it can be potentially dangerous for those more jumping-to-conclusions type users. We do not know how many measures Twitter is taking against misinformation behind the scenes, but fortunately, the fact that Twitter is attempting to pilot a serious attempt at combating misinformation via crowd-source means that they think that veracity is an important feature for its users.

If someone is running a race and they pass the person in second place, what place are they in? Most people would answer either first place (more intuitive answer) or second place (more analytical answer). However, only one of these answers is correct: the analytical answer. This is a question often asked in standardized testing or interview environments to help determine how analytical a person is. If they come to a conclusion quickly without much evidence, they are usually more intuitive and have more Jump-to-Conclusion (JTC) bias, whereas if they take their time to ponder the question before answering, they are usually more analytical. People who respond more analytically have been proven to be less susceptible to believe misinformation and conspiracy theories. The lesson here is that the more one can coach and teach people to think critically and analytically, the lower the likelihood that they will end up believing and spreading misinformation. This is the key reason of why big social media companies like Twitter have an obligation to address the misinformation and conspiracy theory issue affecting our society today. However, the issue with misinformation is its multi-modal nature.

Compounding the issue, is the fact that most classifiers are trained with human-labeled data. At this point, researchers usually face the trade-off between more annotated data and the costs associated with it. Complex classifiers usually need a substantial amount of data to train, but this is of course associated with a high cost of creating the annotated dataset. Another trade-off that researchers must deal with is the veracity of the annotation. Some of them hypothesize that the truth lies with the opinion of the majority, so they go the route of creating a large volume of annotated data via crowd-sourcing where they simply ask people about their own opinion as to whether a piece of information is misleading or not. Some other researchers acknowledge the issue with this approach and they depend more on a human-based fact-checking route. This of course, is an issue, since not all misinformation is factually false. Another issue with the latter approach is that it is very cost-intensive.

Although these issues are salient, headway can still be made with traditional tools; especially by agents that have access to large amounts of training data within the context through which the misinformation is being spread. Even though Twitter's Birdwatch approach is not one of classification but rather recommendation, we decided to try classifying misinformation based on the written notes from the user annotating the Tweet, with the idea of extending this classifier to general thread comments. The main issue is that we do not know, and cannot show, whether our classifier performs well in the "real" world, since we do not have access to annotated Tweets regarding whether they are misinformation or not. However we do think that the performance of our extremely simple classifier was promising. It is plausible that better architecture decision can achieve better performance. Finally, we think that classifying something as "misinformation" might be too ambitious; we think that classifying on softer categories, i.e. reliability, factual, intent, sentiment, might be more useful for solving the hard problem of online misinformation.

A Appendix: Description of Birdwatch Dataset's Tables

A.1 Preliminary Data Exploration

The Birdwatch data is released as two separate files: one containing a table representing all Birdwatch notes and one containing a table representing all Birdwatch note ratings. These tables can be joined together on the noteId field to create a combined dataset with information about notes and their ratings. The data is released in two separate tables/files to reduce the dataset size by avoiding data duplication (this is known as a normalized data model). Currently, we have one cumulative file each for notes and note ratings.

The snapshot of this dataset is a cumulative file and contains all non-deleted notes and note ratings ever contributed to Birdwatch, as of 48 hours before the dataset release time, and subsequently the snapshot used in this analysis is always 48 hours old.

Each data snapshot table is stored in a tsv (tab-separated values) file format with a header row. This means that each row is separated by a newline, each column is separated by a tab, and the first row contains the column names instead of data. The note and note rating data is directly taken from the user-submitted note creation and note rating forms, with only minimal added metadata (like ids and timestamp).

All Birdwatch notes start out with the Needs More Ratings status until they receive at least 5 total ratings. Once a note has received at least 5 ratings, it is assigned a Note Helpfulness Score according to the algorithm described below. Notes with a Note Helpfulness Score of –0.08 and below are assigned Currently Not Rated Helpful, and notes with a score of 0.40 and above are assigned Currently Rated Helpful. Notes with scores in between –0.08 and 0.40 remain labeled as Needs more Ratings. In addition to Currently Rated / Not Rated Helpful status, labels also show the two most commonly chosen explanation tags which describe the reason the note was rated helpful/unhelpful. Notes with the status Needs More Ratings remain sorted by recency (newest first), and notes with a Currently Rated / Not Rated Helpful status are sorted by their Helpfulness Score.

Our original file from Notes contained 118,947 records, and our original file from Ratings contained 740,947 records.

A.2 Notes Table

Attributes in Notes Data		
Field	**Description**	**Response Options**
noteId	Unique note ID	N/A
participantId	Author user identifier	N/A
createdAtMillis	Timenote was created, in milliseconds	N/A
tweetId	The tweetId for the tweet that the note is about	N/A
classification	“Given current evidence, I believe this tweet is:”	“Not misleading”, “Misinformed or potentially misleading”
believable	“If this tweet were widely spread, its message would likely be believed by:”	“Believable by few”, “Believable by many”
harmful	“If many believed this tweet, it might cause:”	“Little harm”, “Considerable harm”
validationDifficulty	“Finding and understanding the correct information would be:”	“Easy”, “Challenging”
misleadingOther	“Why do you believe this tweet may be misleading?”	1 if “Other” is selected, else 0
misleadingFactualError	“Why do you believe this tweet may be misleading?”	1 if “It contains a factual error” selected, else 0
misleadingManipulatedMedia	“Why do you believe this tweet may be misleading?”	1 if “It contains a digitally altered photo or video” selected, else 0
misleadingOutdatedInformation	“Why do you believe this tweet may be misleading?”	1 if “It contains outdated information that may be misleading” is selected, else 0
misleadingMissingImportantContext	“Why do you believe this tweet may be misleading?”	1 if “It is a misrepresentation or missing important context” is selected, else 0
misleadingUnverifiedClaimAsFact	Why do you believe this tweet may be misleading?”	1 if “It presents an unverified claim as a fact” is selected, else 0
misleadingSatire	Why do you believe this tweet may be misleading?”	1 if “It is a joke or satire that might be misinterpreted as a fact” is selected, else 0
notMisleadingOther	“Why do you believe this tweet is not misleading?”	1 if “Other” is selected, else 0
notMisleadingFactuallyCorrect	“Why do you believe this tweet is not misleading?”	1 if “It expresses a factually correct claim” is selected, else 0
notMisleadingOutdatedButNotWhenWritten	“Why do you believe this tweet is not misleading?”	1 if “ It is clearly satirical/joking” is selected, else 0.
notMisleadingPersonalOpinion	“Why do you believe this tweet is not misleading?”	1 if “It expresses a personal opinion” is selected, else 0.
trustworthySources	“Did you link to sources you believe most people would consider trustworthy?”	1 if “Yes” is selected, 0 if “No” is selected
summary	“Please explain the evidence behind your choices, to help others who see this tweet understand why it is not misleading”	User entered text explanation

A.3 Ratings Table

Attributes in Ratings Data		
Field	**Description**	**Response Options**
noteId	Unique note ID	N/A
participantId	Author user identifier	N/A
createdAtMillis	Timenote was created, in milliseconds	N/A
agree	“Do you agree with its conclusion?”	1 if “Yes” is selected, 0 if “No” is selected
disagree	Do you agree with its conclusion?”	1 if “No” is selected, 0 if “Yes” is selected
helpfulnessLevel	"Is this note helpful”	“Not helpful” “Somewhat helpful” “Helpful”
helpfulOther	“What about this note was helpful to you?”	1 if “Other” is selected, else 0.
helpfulClear	“What about this note was helpful to you?”	1 if “Clear and/or well-written” is selected, else 0
helpfulGoodSources	“What about this note was helpful to you?”	1 if “Cites high-quality sources” is selected, else 0
helpfulAddressesClaim	What was helpful about it?”	1 if “Directly addresses the Tweet’s claim” is selected, else 0
helpfulImportantContext	“What was helpful about it?”	1 if “Provides important context” is selected, else 0
helpfulUnbiasedLanguage	What was helpful about it?”	1 if “Neutral or unbiased language” is selected, else 0
notHelpfulOther	“Help us understand why this note was unhelpful”	1 if “Other” is selected, else 0
notHelpfulIncorrect	“Help us understand why this note was unhelpful”	1 if “Incorrect information” is selected, else 0
notHelpfulSourcesMissingOrUnreliable	“Help us understand why this note was unhelpful”	1 if “Sources missing or unreliable” is selected, else 0
notHelpfulMissingKeyPoints	“Help us understand why this note was unhelpful”	1 if “Misses key points or irrelevant” is selected, else 0
notHelpfulHardToUnderstand	"Help us understand why this note was unhelpful”	1 if “Hard to understand” is selected, else 0
notHelpfulArgumentativeOrBiased	“Help us understand why this note was unhelpful”	1 if “Argumentative or biased language is selected, else 0
notHelpfulSpamHarassmentOrAbuse	“Help us understand why this note was unhelpful”	1 if “Spam, harassment, or abuse” is selected, else 0
notHelpfulIrrelevantSources	“What was unhelpful about it?”	1 if “Sources do not support note” is selected, else 0
notHelpfulOpinionSpeculation	“What was unhelpful about it?”	1 if “Opinion or speculation” is selected, else 0
notHelpfulNoteNotNeeded	What was unhelpful about it?”	1 if “Note not needed on this Tweet” is selected, else 0

B Appendix for "Text Summary from Notes" Model

B.1 Data Preprocessing

Using the 2022/05/15 snapshot of the *notes* Birdwatch dataset, we start with 30162 unique tweet-annotations pairs. Some of these notes are written in languages that are not English, but that share Latin characters with English, while some other notes do not share any Latin-based characters; for example there are some notes that are written in Arabic. We drop all of the notes that have no natural ASCII encoding, leaving us with 30154 notes.

We then continue to remove all URLs from the summary texts. Note that we do not drop these observations, but simply remove the URL text from the summary text. Also note that our total observations go down to 30152 now, because two notes had only a URL text on their summary text and nothing else.

It turns out that many annotators put URLs containing what they believe to be evidence for their categorization. It should be noted that 73% of tweets annotated as "Misinformed or potentially misleading" have URLs, while 42% of annotations labeled as "Not Misleading" have URLs; indicating that perhaps creating binary feature as to whether a note contains a URL in the text might be useful in classifying each note.

We also drop notes that seem to be testing notes. Our measure is extremely coarse and it is plausible that there are still test annotations that we have not detected. Our approach is to drop all observations for which the text starts with the string *[TEST, NOT REAL]*, we are left with 30151 notes.

Now we turn to traditional natural-language processing (NLP) preprocessing steps that actually change the text, instead of filtering observations. We first remove all English stop words from the text, we also remove all punctuation from text. Finally we tokenize the text and lemmatize the tokens using a lemmatizer created by the authors of WordNet.

The lemmatizer matches a string and uses a set of morphology functions to match the string into one of the primitive roots that are found in the WordNet dataset.

We finally drop all observations that contain the token "birdwatch" in the summary. After first training our model, we realized that the strongest feature of our model was this token. After looking at the notes that contained this token, we again realized that many of these were testing notes and so we drop them at the end of the preprocessing stage, leaving with a total of 29687 notes.

References

Srivastava Pranj. Essentials of Deep Learning : Introduction to Long Short Term Memory. 2017.

List of Figures

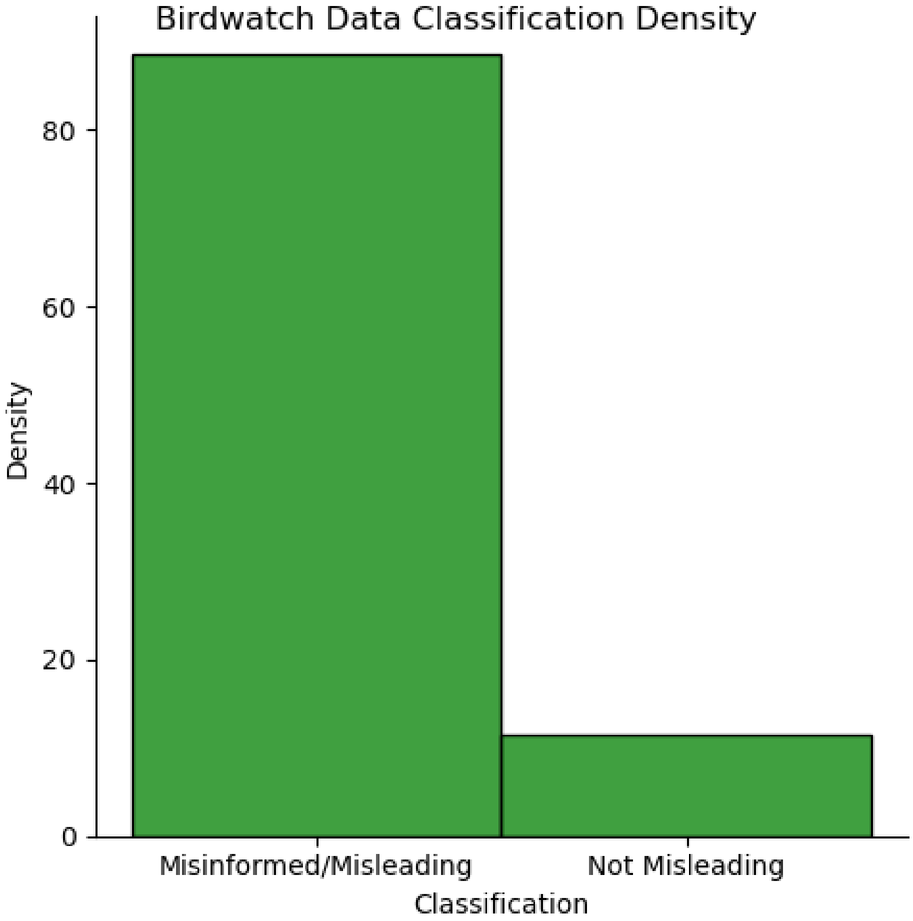

Figure 1: Distribution of Misinformation Classification Data

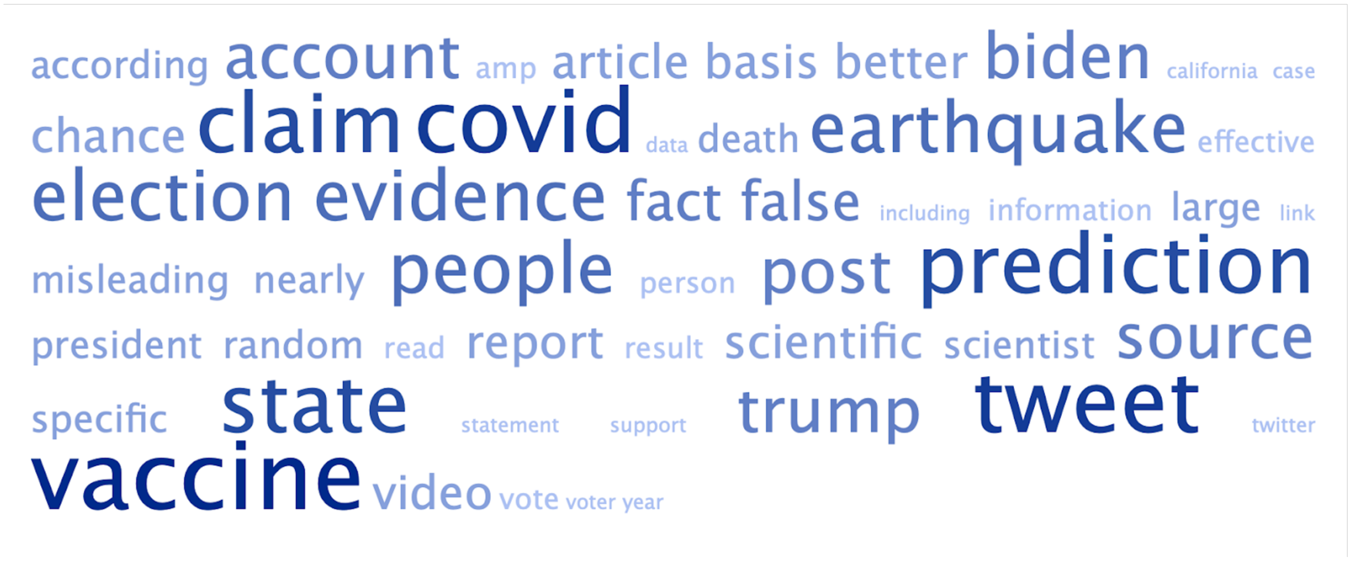

Figure 2: Top Words Word Cloud

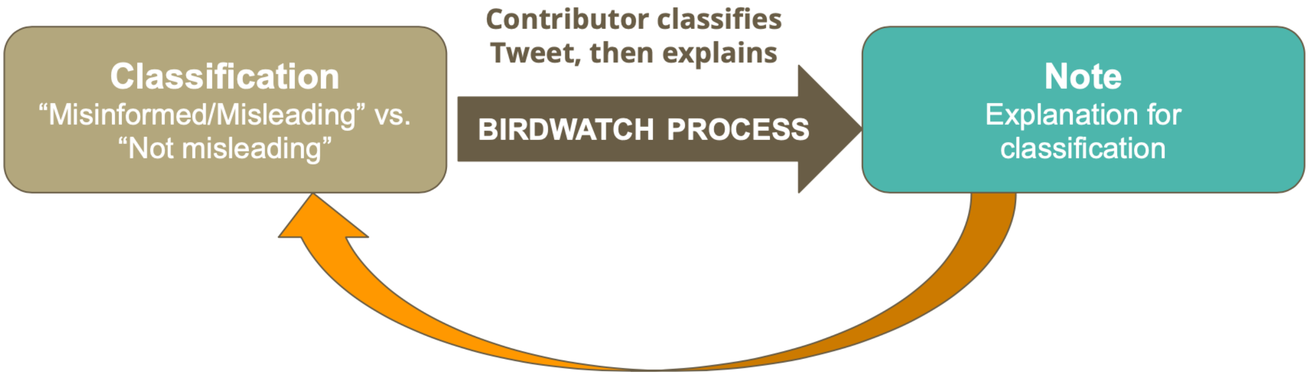
Contributor classifies Tweet, then explains
Classification
“Misinformed/Misleading” vs. “Not misleading”
BIRDWATCH PROCESS
Note
Explanation for classification

Figure 3: Machine Learning Approach

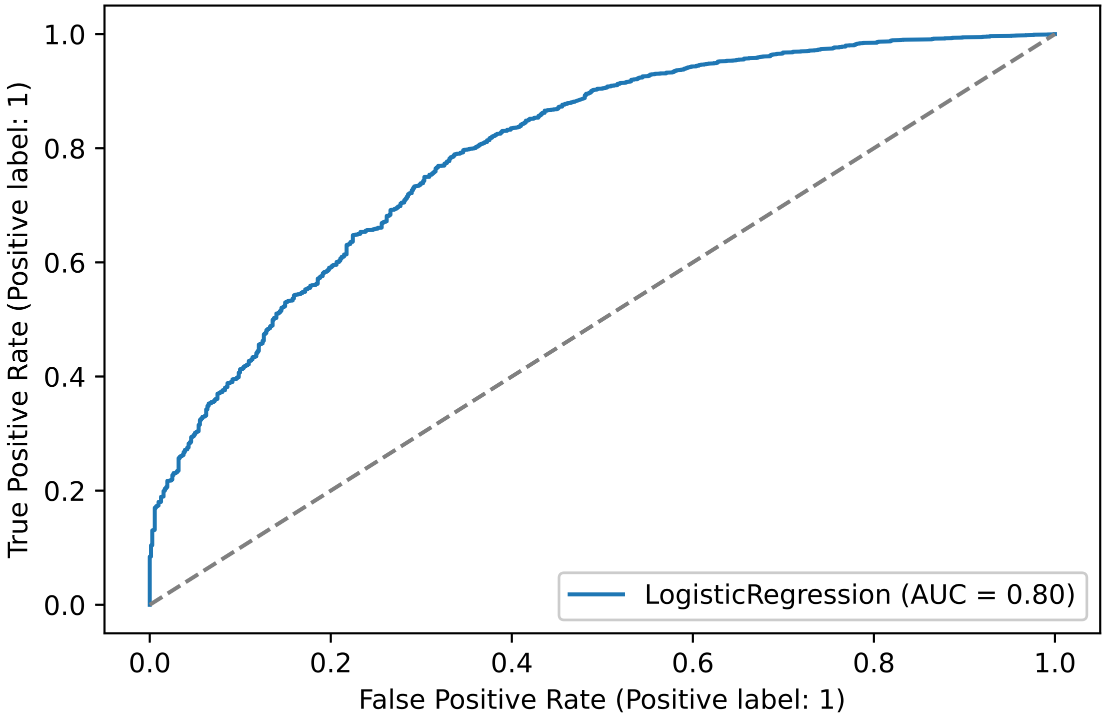

Figure 4: Bag-of-Words Vectorization | ROC Curve

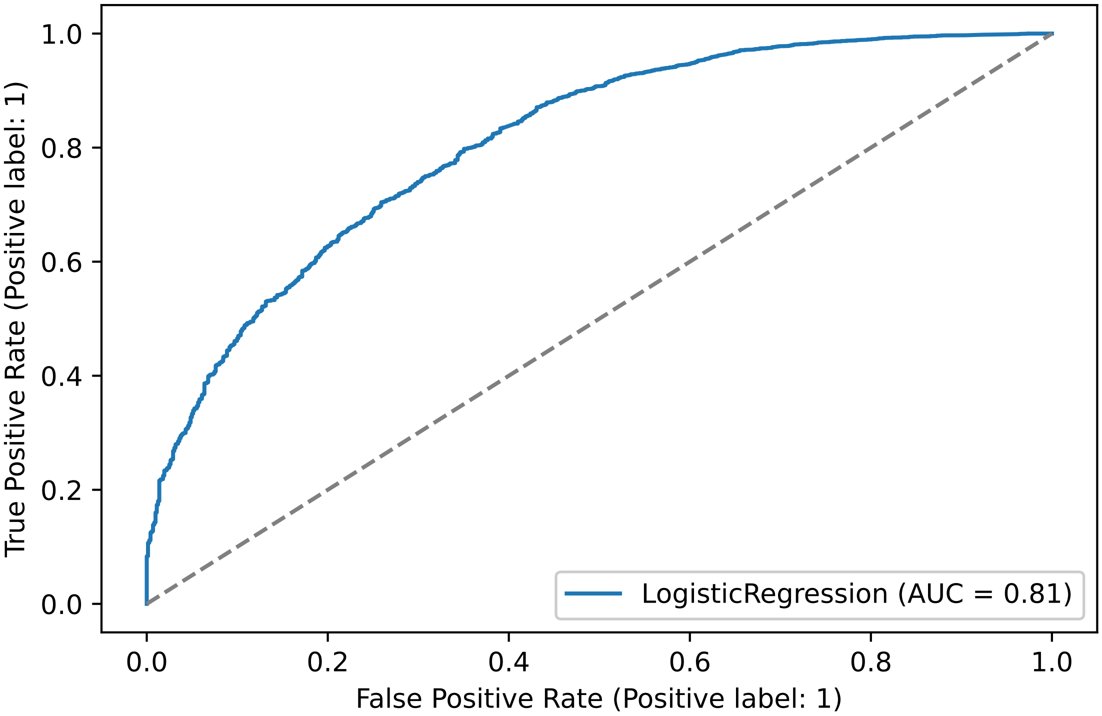

Figure 5: TF-IDF Vectorization | ROC Curve

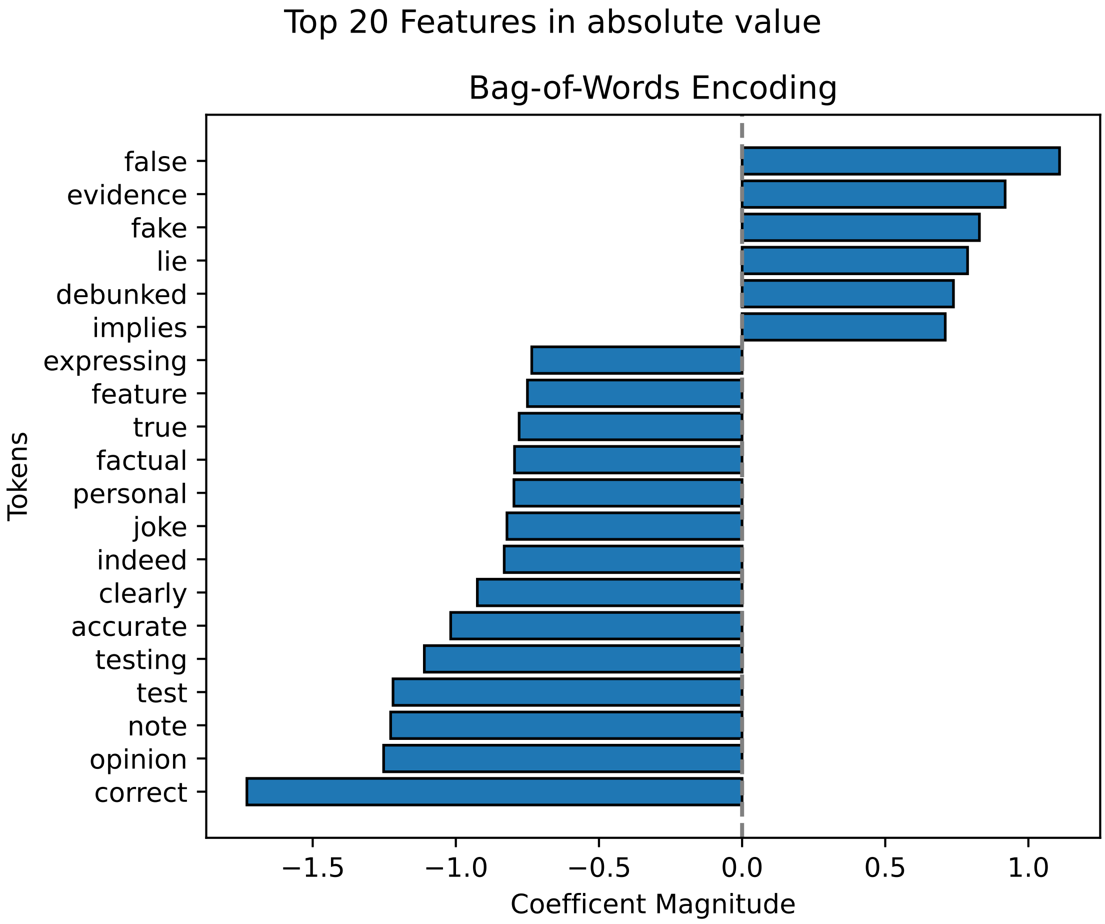

Figure 6: Bag-of-Words Vectorization | Largest Token Coefficients

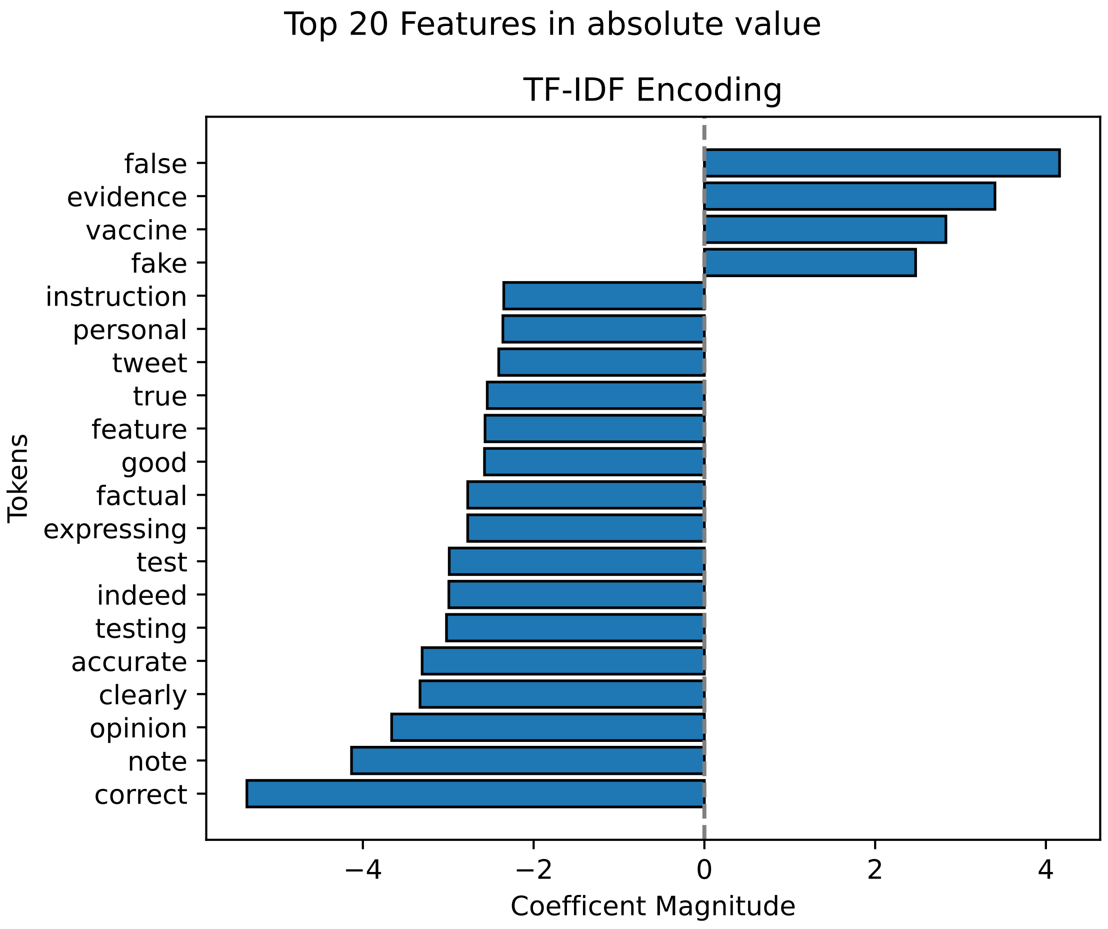

Figure 7: TF-IDF Vectorization | Largest Token Coefficients

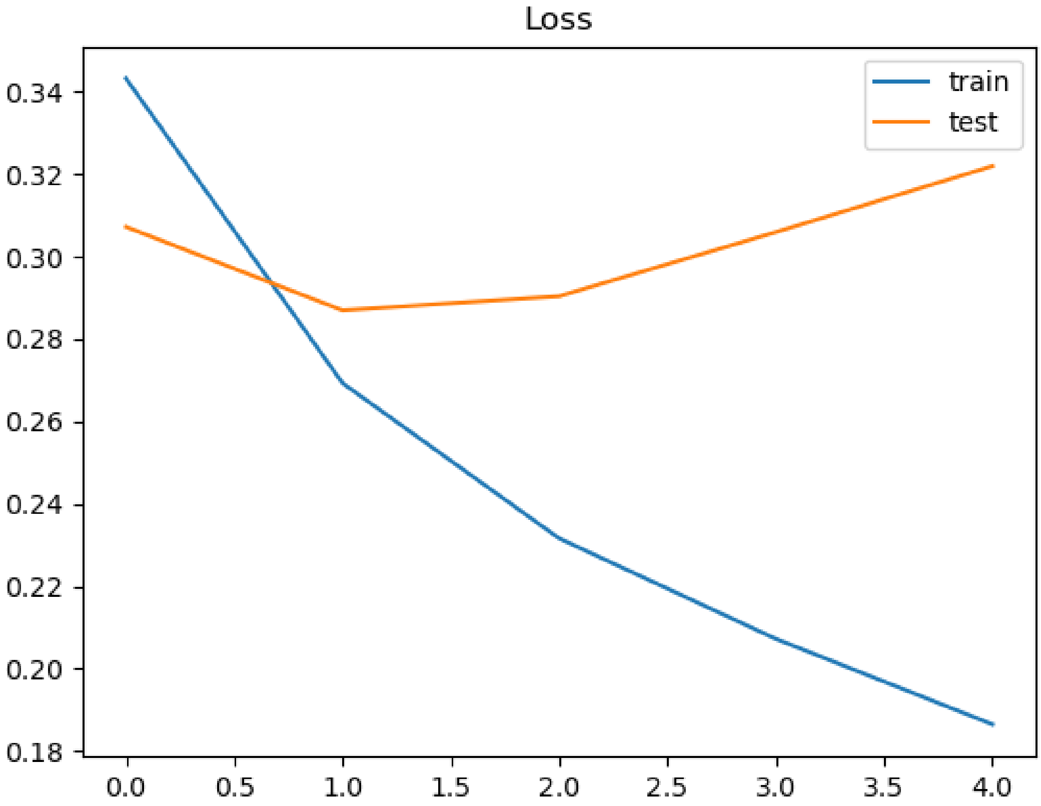

Figure 8: LTSM NN Model | Train: Accuracy vs. Epochs

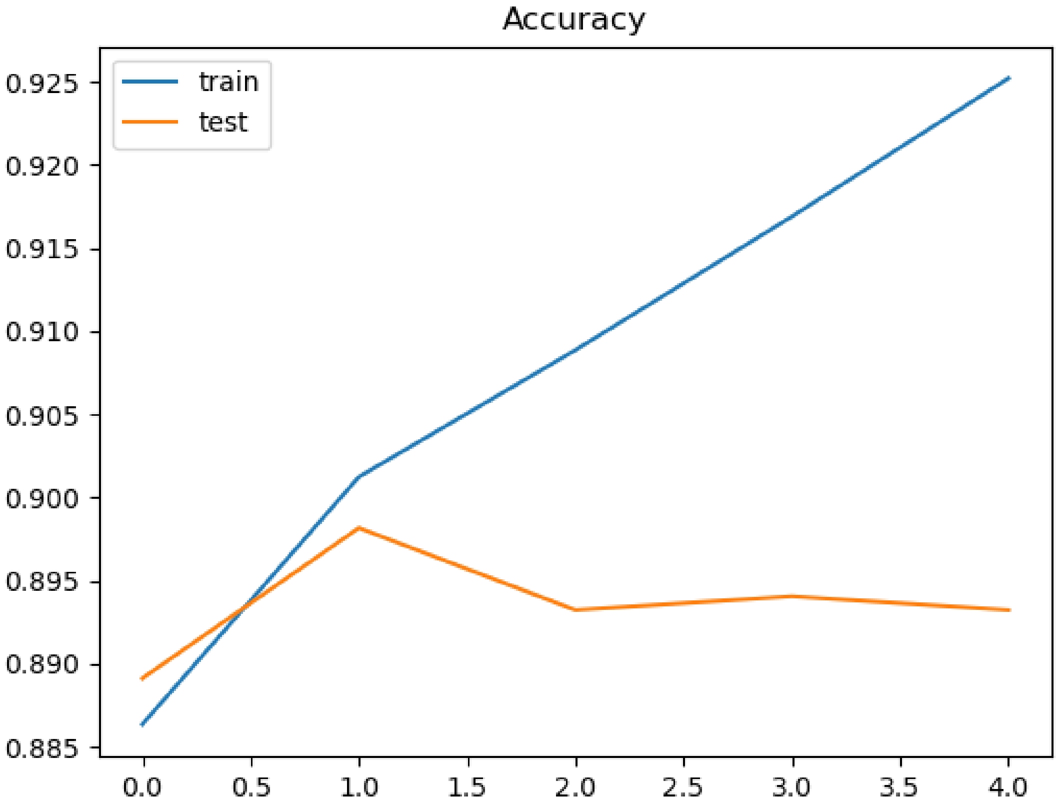

Figure 9: LTSM NN Model | Test: Accuracy vs. Epochs

List of Tables

	TF-IDF	BOW
Train Accuracy	0.905	0.900
Test Accuracy	0.895	0.888
Train Precision	0.907	0.901
Test Precision	0.898	0.892
Train Recall	0.996	0.996
Test Recall	0.994	0.994
Train F1-Score	0.943	0.940
Test F1-Score	0.949	0.946
Train AUC	0.932	0.912
Test AUC	0.809	0.797

Table 1: Logistic Regression Model Scores