Scholarly Impact Forecasting
Machine Learning Project CSE 575

Mehrdad Zakershahrak, (1211435014), mzakersh@asu.edu
Shakiba Yaghoubi, (1209573063), syaghoub@asu.edu
Daniel Molina, (1204828140), d.molina@asu.edu
Mehdi Ghasemirahaghi, (1211452408), mghasem1@asu.edu

1 Introduction

Predicting the success of scientific work has been attracting extensive research attention in recent years. Scholars, especially junior scholars, who could master the key to producing high-impact work could attract more attention, as well as research resources; and thus, put themselves in a better position in their career development. High impact work remains as one of the most important criteria for various organizations (e.g. companies, universities, and governments) to identify top talent, especially at their early stages. It is highly desirable for researchers to judiciously search for the right literature that can best benefit their research [1]. One of the many factors that indicate the success/impact of a scientific work is its frequency of use, namely its citation count [2].

Predicting the "long-term" impact of a scientific work has been made possible in recent years. In [2], a model for citation dynamics of individual papers is used to predict a paper's long-term citation based on its short-term citations. In [3], an author's reputation is used for forecasting the citation count. Despite advances in scientific impact prediction, solutions to the general problem are challenging seeing as: (1) many factors contribute to a paper's impact, some of which might be of greater importance; (2) the features might have a very complicated relationship with the citation count; a relation which is more complicated than a linear relationship; and (3) all the features do not have numeric values.

In this project, we try to predict the citation count of a paper based on features like its authors, publication venue, title, abstract, and publication year. We tried to come up with the best methodologies to predict the citation count of a paper using a static model of features. Some of the methods we tried are k-nearest neighbors, linear/nonlinear regression, decision trees, linear SVM regression, random forest and neural nets (multilayer perceptrons). In what follows, we will (1) describe how our raw data looks and how we extracted the features out of one entry of the data, (2) study the probabilistic properties of the data, (3) apply different regression models for citation prediction, (4) analyze the results, and (5) compare classification and regression models.

2 Problem description

As mentioned above, different factors are important to be taken into account when deciding to predict the academic attention (citation count) for a paper. For instance, the author names

listed in a paper can have effect on how many times a paper is cited. There are other factors such as publication year, publication venue, and keywords in the tile and abstract that affects the citation count. Predicting the number of citation for each paper based on these factors is challenging due to two main reasons. Firstly, all the factors do not have numeric values to be used in regression and classification models. Secondly, the relationship between citation count and these factors is not straightforward to find.

3 Dataset

The used dataset in this project is from the Aminer Citation Network [4]. This database is designed only for research purposes. The dataset is taken from DBLP, ACM, MAG (Microsoft Academic Graph), and other resources. The first version of this dataset contains 629814 papers and 632752 citations. In this project, we have used DBLP-citation-Network-V6 [5]. This version has 2084055 papers and 2244018 citation relationships. Each paper in the dataset is associated with title, authors, year, venue, citation count, and abstract. A sample dataset entry has been shown in the Fig. 1. There should be a parser to extract different fields of data from each line of dataset. Another important consideration is that all the features do not have numeric values.

```
#*Spatial Data Structures.
#@Hanan Samet
#year1995
#confModern Database Systems
#citation3253
#index25
#arnetid27
#%165
#!An overview is presented of the use of spatial data structures
in spatial databases. The focus is on hierarchical data
structures, including a number of variants of quadtrees, which
sort the data with respect to the space occupied by it. Such
techniques are known as spatial indexing methods. Hierarchical
data structures are based on the principle of recursive
decomposition. They are attractive because they are compact and
depending on the nature of the data they save space as well as
time and also facilitate operations such as search. Examples are
given of the use of these data structures in the representation
of different data types such as regions, points, rectangles,
lines, and volumes.
```

Figure 1: A sample entry of dataset

4 Methodology

In this section, we discuss preprocessing the dataset and the way we extracted the different features. In the sequel, we talk about different regression models used to predict the number of citations.

4.1 Preprocessing and Feature selection

In order to predict the number of citations for each paper, the dataset is first parsed using our own parser code in Python. Firstly, different fields of a paper are extracted from the dataset,

ignoring the delimiters and separators. These fields are stored into a dictionary data structure in Python. In the sequel, all the fields are converted to numeric values. Each paper has five fields: title, authors, year, publication venue, and abstract. Citation count is considered as the metric to predict.

For the author field, we have calculated the total number of citations an author has in whole dataset. Likewise, we have calculated the total number of citations for the papers in a specific publication venue in the whole dataset. The publication year is directly extracted from the dataset, since it is a numerical feature.

Title and abstract field do not have numeric values and the numeric representation of text documents is a challenging task in machine learning. There are some methods to handle text documents such as bag of words and Latent Dirichlet Allocation (LDA) [6]. When using bag of words, you are opt to lose many important details which potentially could be a good representation of a string feature (e.g. consideration of word ordering). In LDA, it is hard to identify keywords to be extracted and the results are hard to evaluate.

In this project, we have used word2vec to convert title and abstract to numeric features. The overall scheme of the approach is illustrated in Fig 2. In this method, English common

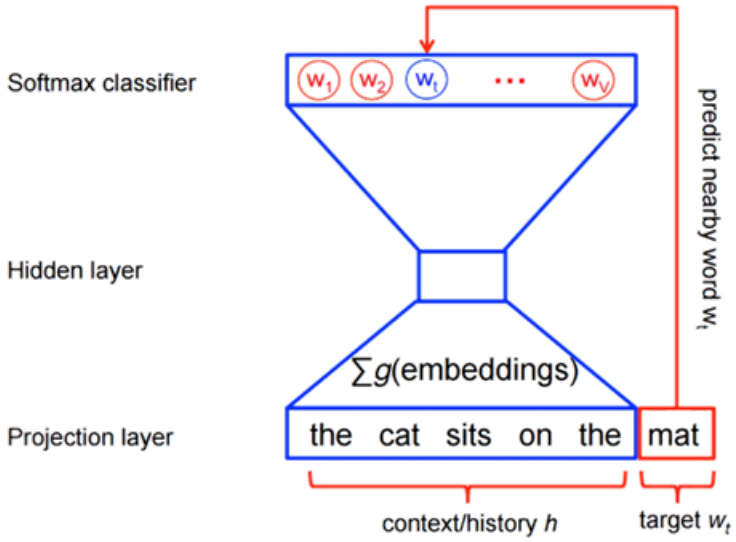


Figure 2: Word2vec method [7]

words (stop words) that should not affect the conversion to numeric values are excluded before Softmax calculation. Probabilistic language models use maximum likelihood (ML) to maximize the probability of the next word w_t given previous words h In the used approach, the probability of $P(w_t|h)$ is calculated using the Softmax function with the following formula:

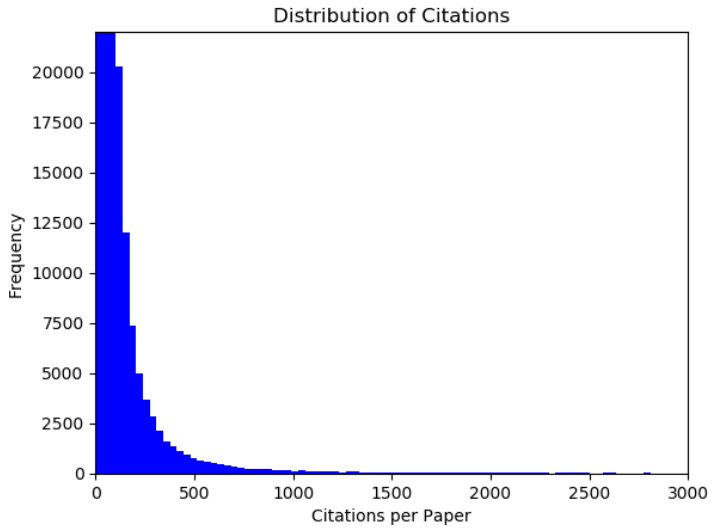
$$P(w_t|h) = Softmax(score(w_t, h)) = \frac{exp(score(w_t, h))}{\sum_{Word w' in Vocab} exp(score(w', h)))} \quad (1)$$

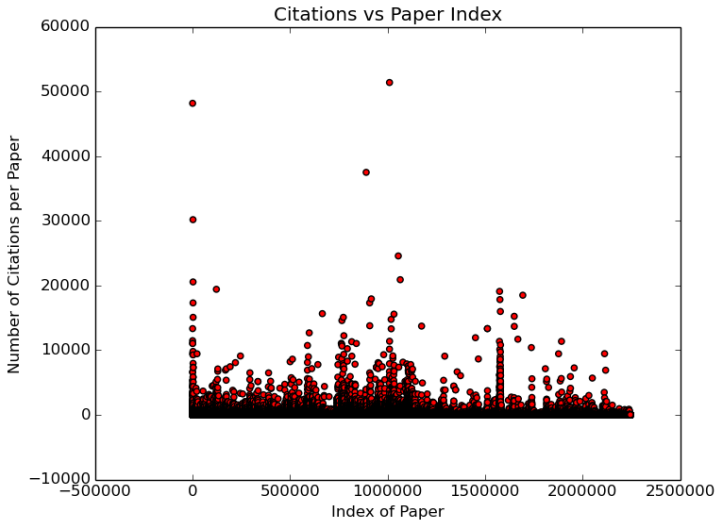
where $score(w_t|h)$ is the compatibility of word w_t with the context h. The model is trained using maximizing log-likelihood on the training set which is maximizing:

$$J_{ML} = logP(w_t|h) = score(w_t, h) - log(\sum_{Word w' in Vocab} exp(score(w', h))) \quad (2)$$

In this project, we have converted title and abstract to 32 features each.

In the sequel, we have extracted plots for the distribution of citation count and number of citations for each paper id, along with statistical information about some of the features, in Fig. 3. As it can be seen, most of the papers have less than 500 citations. Also, the histogram for different features has been illustrated in Fig. 4. The most recent papers in the dataset are from the year 2013. Ultimately, the scatter plot of citation count versus different features has been depicted in Fig. 5.





	Mean	Standard Deviation	Min	25% Quartile	50% Quartile	75% Quartile	Max
Citation	17.8	131.6	0	0	2	11	51373
Author Citation	3268.2	7577.5	0	127	821	3133	616219
Publication Venue Citation	56079.7	117692.4	0	1813	10592	48793	681023
Year	2004	11.5	-1	2001	2007	2010	2013

Figure 3: Histogram and scatter plot of the citation counts, along with statistical information about some of the features

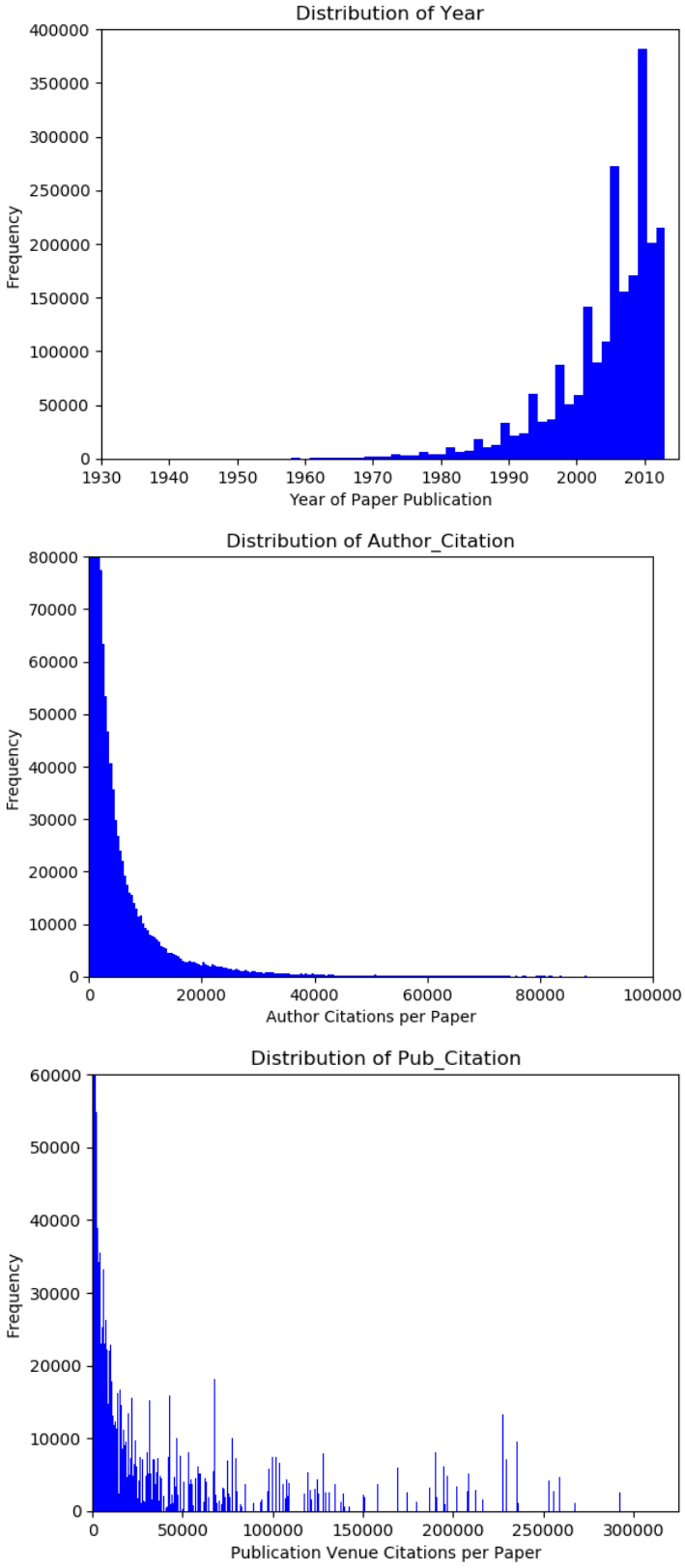


Figure 4: Histograms of some of the features

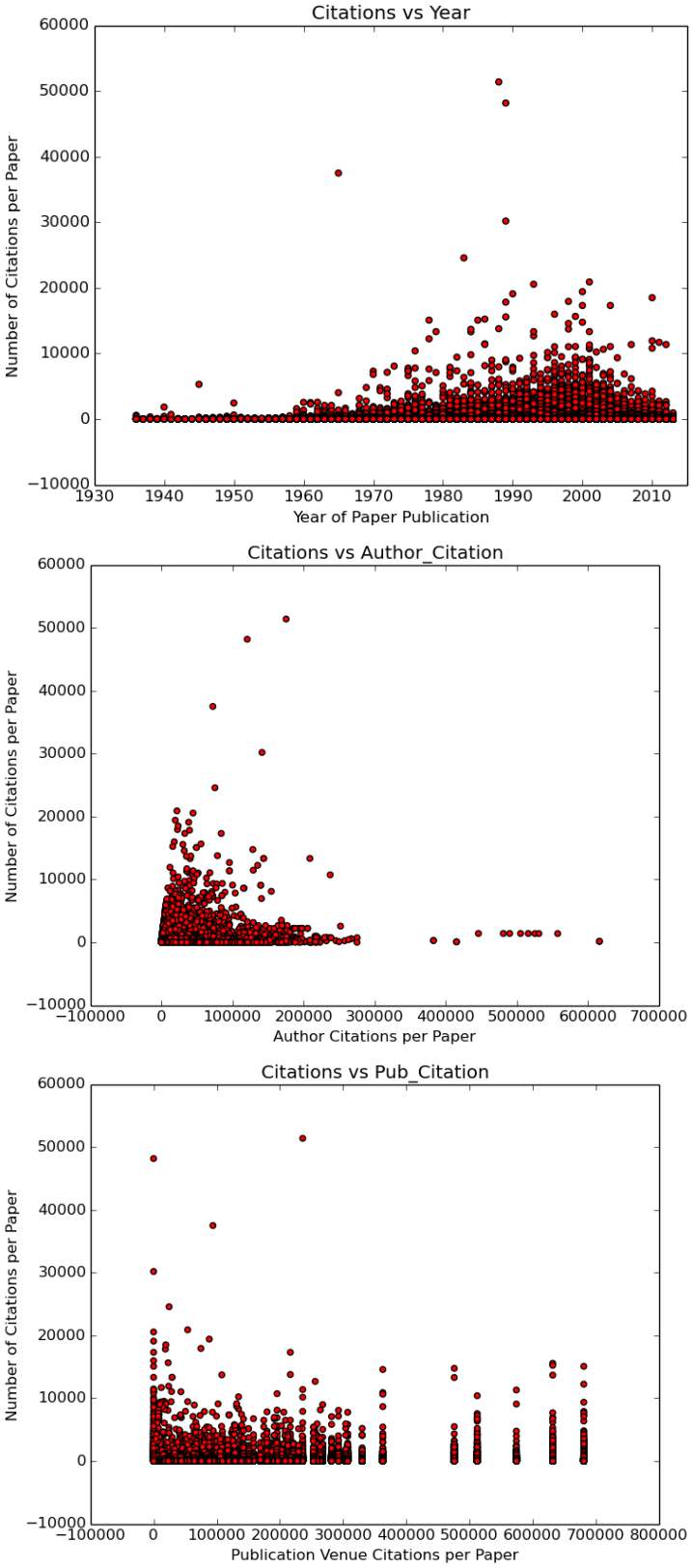


Figure 5: Scatter plots of the citation counts vs their corresponding features

4.2 Models

The following models have been used for prediction:

- Linear Regression (LR): A matrix B that minimizes the MSE of the prediction $\hat{Y} = B * [1, X]$ is trained that is going to be used for prediction on the test set.
- Nonlinear Regression (NLR): While in linear regression, multipliers of a first order model are predicted, in the NLR, the multipliers of a polynomial model of degree n is predicted. In this project we used 2nd order polynomials. This means that we added features created based on multiplication of two of previous features.
- k-Nearest Neighbor (kNN): We used kNN with k=10 in this project.
- Linear SVM Regression LSVM): This method fits a linear model that is trained based on the support vectors of the data.
- Decision regression tree (RT): In this model, observations on features (represented in branches) are used to make decisions about the output.
- Multi-layer Perceptron Neural Nets (NN(MLP)): In order to predict the citation count, we have used the MLP regressor. Adam solver is used to expedite the process of solving neural network (Adam solver works efficiently for large datasets). The activation function in the neural network is tanh function. The maximum iteration of the algorithm is set to 500. The hidden layer size is set to 100.

5 Results

In this section, we see the results of applying different regression models. In the sequel, we analyze the obtained results.

5.1 Regression results

In this section, the results of regression models will be discussed. In order to compare different algorithms, the whole dataset has been randomly split into training (about 90%) and test (about 10%) sets. Ten different random splits have been tested and the average result has been calculated.

Figure 6 depicts the results of different methods in terms of mean absolute error. As can be seen in the figure, the k-nearest neighbor (kNN) has the best performance when comparing different methods based on mean absolute error. Figure 7 depicts the results of different methods based on the average root mean squared error. As it is shown, the non-linear regression (NLR) method has the best performance when comparing different methods based on root mean squared error. As a conclusion, if we want the best results on average, we choose the method with the least mean squared error (i.e. NLR). This model is however sensitive to outliers. If we want a robust model that performs best on average, we choose the one with least mean absolute error (i.e. kNN). This model may have big bias for outliers.

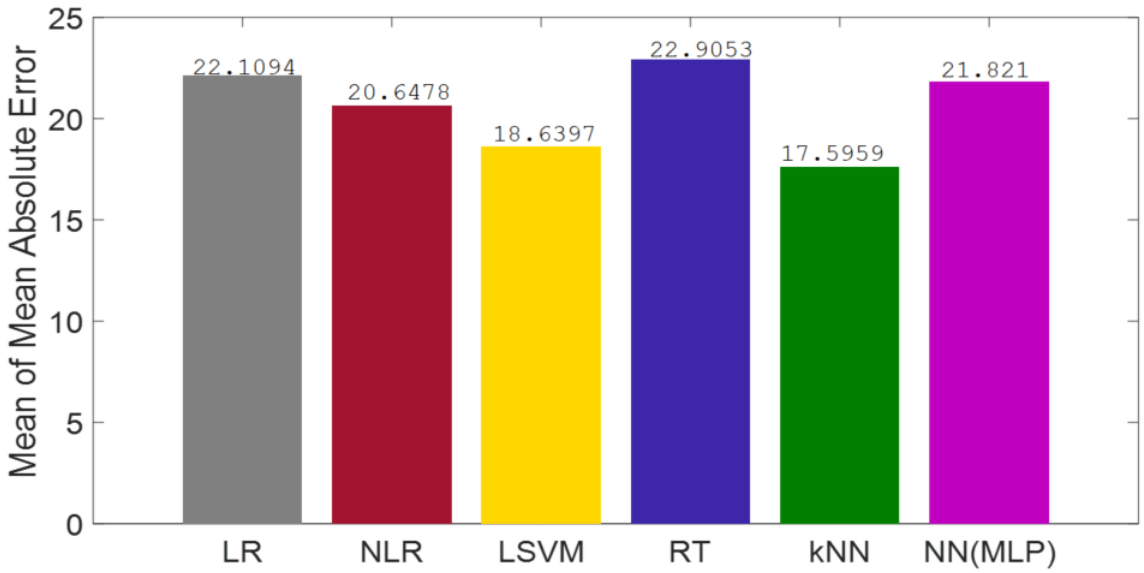


Figure 6: Mean of mean absolute error for different runs

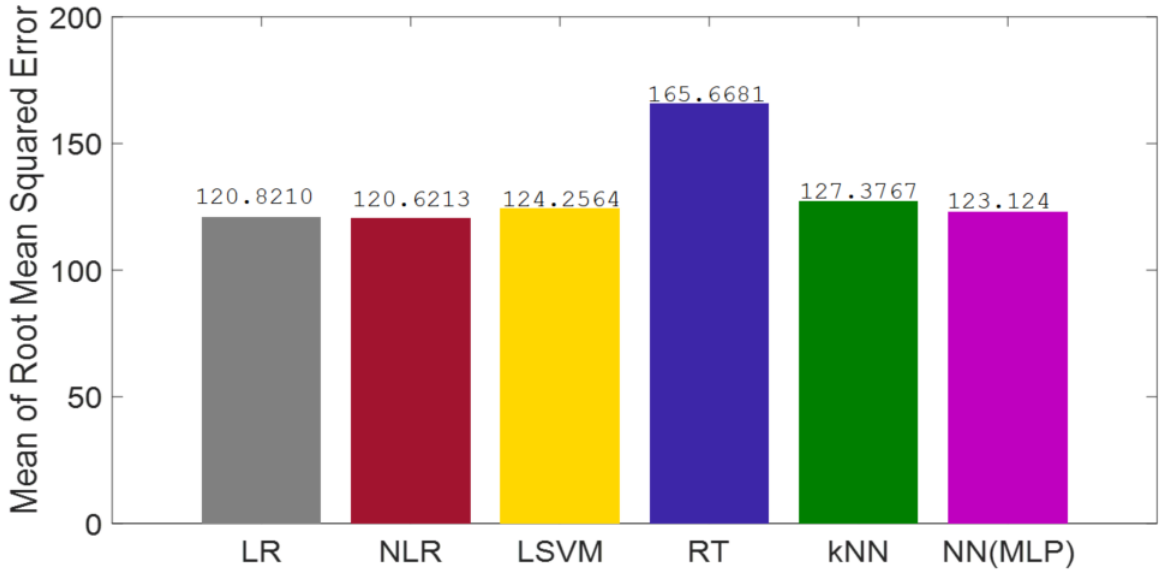


Figure 7: Mean of root mean squared error for different runs

5.2 Regression analysis

The following factors might have affected our results. Taking these factors into account might help improve our results. There is no dynamics in our models, we basically use static features to predict the model, the only feature that takes into account the importance of time is the year of publication, while most of previous works use yearly citation count (which was not provided in our data set.) It was even shown in [1] that using the summation of citations in the first three years as the only feature for citation prediction is almost as accurate as using it along with other features such as topic, author feature,etc. We could exclude papers that did not have enough chance to get cited (We had a lot of 2013 papers and the data set was also published in 2013. We assume that we should have omitted papers with less than 1-2 years history since the papers

that could have cited them are going to be published later)

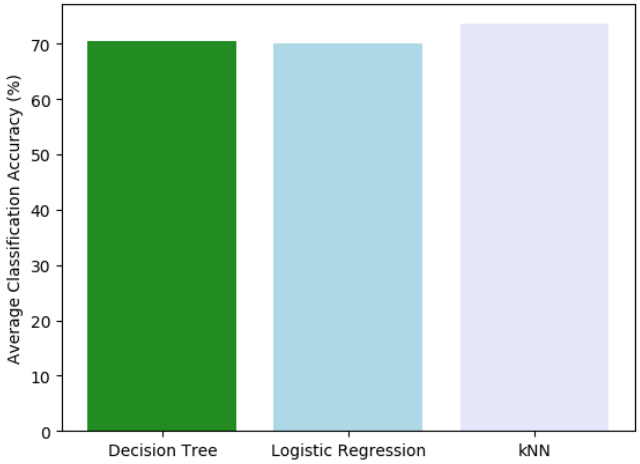
The complex relationship between the features and the impact cannot be well characterized by simple linear models, that is why we should not have expected to get good results using linear regression or linear SVM regression.

The data set is very large, covering scientific work in different domains. While the impact of scientific work in different fields might behave differently; yet some closely related fields could still share certain commonalities. Thus, a one-size-fits-all or one-size-fits-one solution might be sub-optimal. This property should somehow be taken into account.

5.3 Classification vs Regression

Predicting a paper's exact citation count can be prone to error due to the various factors discussed above; thus, one might be more interested in a simpler question: whether or not a paper will be cited at all. Thus, we briefly consider the binary classification task of determining if a paper will or will not be cited.

Due to our dataset having a high number of papers with 0 citation (38%), we figured that this classification task would be a viable problem to explore. We decided to compare the following classifiers: Decision Tree, Logistic Regression, and k-Nearest Neighbors. As it can



	Average Accuracy	Average Precision	Average Recall	Average F1 Score
Decision Tree	0.705	[0.605, 0.768]	[0.62, 0.757]	[0.612, 0.763]
Logistic Regression (C = 1)	0.7	[0.629, 0.729]	[0.49, 0.826]	[0.55, 0.775]
kNN (k = 10)	0.736	[0.655, 0.782]	[0.63, 0.8]	[0.642, 0.791]

Figure 8: Mean classification accuracies, along with diagnostics, across the three classifiers

be seen in Fig 8, kNN edges out both decision tree and logistic regression in almost all of the evaluation metrics, except in average recall for one of the two classes. Also, it is worth mentioning that even though decision tree and logistic regression have similar classification accuracies, decision tree does a lot better job than logistic regression in classifying papers with class 0, as exemplified by the higher mean precision, recall, and F1 score.

Thus, as in the regression analysis, if we want a robust model that performs best on average, we choose kNN. Again, the only issue is its susceptibility to outliers.

Ultimately, the classifiers do a decent job of predicting whether or not a paper is expected to be cited, but they are far from perfect. The main issue is that there is only so much a model can learn about a dynamic problem, such as predicting a paper's constantly changing citation count, given static features. This was further exemplified when we attempted to learn more than two classes. Regardless of how we assigned classes, the classification accuracy would never be much greater than 50%. Thus, it is fair to conclude that citation classification is a far too nuanced task for our given dataset.

6 Conclusion

There are different factors that affect the total citation number of a research paper. There are two challenges to predict the citation count of a paper. Firstly, the relation between citation count and these factors is not straightforward to find. In other words, linear regression may not lead to a good solution in this case. Secondly, all the factors do not have numeric values. In this project, all of the different factors which have an effect on the citation count of a paper have been translated into numeric values. The dataset has been parsed, and the numeric value for each feature were obtained. Then, different methods were applied to predict the citation of a paper. Different methods were compared in terms of mean square and mean absolute error. It was shown that if we want the best results on average, we should choose the method with the least mean squared error, which is NLR. On the other hand, if we want a robust model with the best average, we choose the method with the least mean absolute error, which is kNN. In addition to the regression approach, we also applied classification models to predict whether a paper will get citation or not. As shown, binary classification resulted in mediocre results; and any other attempt at classification resulted in abysmal results. Taking accuracy, precision, recall, and F1 score into consideration, kNN ended up outperforming the other classifiers.

Through considering both regression and classification approaches, we realized how difficult and nuanced a task like citation prediction ends up being.

7 Future work

As the future work, we can predict the citation count of different papers by the passage of time. Sequential models can be helpful to predict the yearly citation of a paper. The papers should be sorted based on the publication year. Then, a sequential model such as Long Short-Term Memory (LSTM) can be applied to predict the number of citations for a paper. The citation count can be also predicted using the relationship between different papers. The relationship of different papers can be modeled using a graph.

References

[1] L. Li and H. Tong, "The child is father of the man: Foresee the success at the early stage," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 655–664.

[2] D. Wang, C. Song, and A.-L. Barabási, "Quantifying long-term scientific impact," *Science*, vol. 342, no. 6154, pp. 127–132, 2013.

[3] C. Castillo, D. Donato, and A. Gionis, "Estimating number of citations using author reputation," in *String processing and information retrieval*. Springer, 2007, pp. 107–117.

[4] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 990–998.

[5] "Citation network dataset," https://aminer.org/citation, Accessed: 2017-11-30.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[7] "Vector representations of words," https://www.tensorflow.org/tutorials/word2vec, Accessed: 2017-11-30.