Statistical Analysis and Forecasting the number of Research Papers in DBLP Database

Sabeeh-ur-Rehman, Tayyab Hassan and Muhammad Talha

National University of Computer and Emerging Sciences, Islamabad

Contents

[History of dblp: 3](#_Toc59396962)

[Technique and methods: 6](#_Toc59396963)

[Preprocessing 6](#_Toc59396964)

[Parsing the data: 6](#_Toc59396965)

[Feature selection: 7](#_Toc59396966)

[Null values 7](#_Toc59396967)

[Splitting data: 8](#_Toc59396968)

[Feature Conversion: 8](#_Toc59396969)

[Prediction Models on Articles: 9](#_Toc59396970)

[Linear Regressor 9](#_Toc59396971)

[Decision Tree Regressor 10](#_Toc59396972)

[Decision Tree Regressor with K-fold 10](#_Toc59396973)

[Decision Tree Regressor (Unseen Data) 11](#_Toc59396974)

[Decision Tree Regressor with K-fold (Unseen Data) 11](#_Toc59396975)

[Random Forest Regressor (on testing data with k-fold) 11](#_Toc59396976)

[Random Forest Regressor (on unseen data with k-fold) 12](#_Toc59396977)

[Model Training on In Proceeding 13](#_Toc59396978)

[Decision Tree Regressor 13](#_Toc59396979)

[Decision Tree Regressor (Unseen Data) 13](#_Toc59396980)

[Random Forest Regressor (on testing data with k-fold) 14](#_Toc59396981)

[Random Forest Regressor (on unseen data with k-fold) 15](#_Toc59396982)

[Comparison of techniques 15](#_Toc59396983)

[Accuracies of Articles 15](#_Toc59396984)

[Accuracies of In Proceedings 16](#_Toc59396985)

[Combined Data 16](#_Toc59396986)

[Conclusion 17](#_Toc59396987)

[References 17](#_Toc59396988)

[Table 1 Testing data Accuracies 15](#_Toc59396799)

[Table 2 Unseen data Accuracies 16](#_Toc59396800)

[Table 3 Testing data Accuracies 16](#_Toc59396801)

[Table 4 Unseen data Accuracies 16](#_Toc59396802)

[Table 5 Combined data accuracy on testing data 16](#_Toc59396803)

[Table 6 Combined Data Accuracy on unseen data 17](#_Toc59396804)

[Figure 1Count of Features in the dblp Dataset 4](#_Toc59396834)

[Figure 2Year wise publications 5](#_Toc59396835)

[Figure 3 Xml Structure of dblp 6](#_Toc59396836)

[Figure 4 Features of dataframe 7](#_Toc59396837)

[Figure 5 Null value Details 7](#_Toc59396838)

[Figure 6 t-sne Applied After Encoding 8](#_Toc59396839)

[Figure 7 Dataset Snippet 9](#_Toc59396840)

[Figure 8 Accuracy of Linear Regressor 9](#_Toc59396841)

[Figure 9 Decision Tree Regressor 10](#_Toc59396842)

[Figure 10 Decision Tree Regressor with K-fold 10](#_Toc59396843)

[Figure 11 Decision tree on unseen data 11](#_Toc59396844)

[Figure 12 Decision Tree with K-fold on unseen data 11](#_Toc59396845)

[Figure 13 Random Forest (also contain k-fold results) 12](#_Toc59396846)

[Figure 14 Random Forest Regressor (includes cross-validation) 12](#_Toc59396847)

[Figure 15 Decision Tree Regressor with and without k-fold 13](#_Toc59396848)

[Figure 16 Decision tree on unseen data 14](#_Toc59396849)

[Figure 17 Random Forest (also contain k-fold results) 14](#_Toc59396850)

[Figure 18 Random Forest Regressor (includes cross-validation) 15](#_Toc59396851)

# History of dblp:

The dblp (database systems and logic programming) computer science bibliography is the on-line reference for bibliographic information on major computer science publications. It has evolved from an early small experimental web server to a popular open-data service for the whole computer science community. Their mission at dblp is to support computer science researchers in their daily efforts by providing free access to high-quality bibliographic meta-data and links to the electronic editions of publications. As of January 2019, dblp indexes over 4.4 million publications, published by more than 2.2 million authors. To this end, dblp indexes about than 40,000 journal volumes, more than 39,000 conference and workshop proceedings, and more than 80,000 monographs. The key of a DBLP record, you may retrieve the record from the URL

<http://dblp.uni-trier.de/rec/bibtex/key.xml> e.g. if you replace key by journals/acta/BayerM72, you will get the bibliographic record of the famous B-tree paper:

<?xml version="1.0"?>

<dblp>

<article key="journals/acta/BayerM72"

mdate="2003-11-25">

<author>Rudolf Bayer</author>

<author>Edward M. McCreight</author>

<title>Organization and Maintenance

of Large Ordered Indices</title>

...

</article>

</dblp>

Obviously, this looks like a version of the huge dblp.xml file which has been shrunken to just one record. But there is an important difference: The header does not refer to an DTD and we do not use symbolic entities. All non-ASCII characters are encoded by numeric entities, in the header the encoding intentionally has not been specified. This pure ASCII XML document without symbolic entities may be parsed very fast by a lightweight parser. The same encoding is used by the other services

Chart

Description automatically generated

Figure Count of Features in the dblp Dataset

From the above visualization all of the information residing in the dblp xml file can be seen. It can be observed that the number of authors are highest in the xml file.

Chart

Description automatically generated

Figure Year wise publications

From the above visualization it can be observed that there are only 12 articles published in 1936. However this number have been exponentially grow to 1.06 million articles in 2016.

# Technique and methods:

# Preprocessing

## Parsing the data:

The xml file contain data for {"article", "inproceedings", "proceedings", "book", "incollection", "phdthesis", "mastersthesis", "www"}. After reviewing the requirement in the project, it has been observed that: to predict journals and conferences, only the data of “article” and “inproceedings” are needed to be parsed. Therefore, the rest of the data in the xml file is not converted to the csv format.

*Diagram

Description automatically generated*

Figure Xml Structure of dblp

This diagram shows the xml structure of the dblp dataset. To parse the xml file dblp is considered as root and data for inproceedings and article is extracted from their respective tags. Beside the start page, the search pages, and a number of [other special pages](https://dblp.org/faq/1474664.html), the dblp website primarily hosts four different kind of web pages:

* index pages,
* publication stream pages,
* table of content pages, and
* person pages.

## Feature selection:

After finding the correlation of the features it has been observed there is no correlation between “title” “author” and “pages” but if we use NLP model then we could have used keywords from “title” to predict associated “journal/ inproceeding”. As in the course we did not covered NLP models therefore, we dropped “title” from the dataframe as well. Furthermore, we are grouping the data with respect to year and counting the number of occurrences of the journals name and created a third column named “paper per year”.

Graphical user interface, text, application, table

Description automatically generated

Figure Features of dataframe

## Null values

There are total 3 variables including 2 categorical and 1 numerical variable. From the whole dataset there are only 230 missing cells which is less than 1% of the total dataset. This is a small portion of the whole dataset therefore we drop the null values.

Graphical user interface

Description automatically generated

Figure Null value Details

## Splitting data:

The data has been split into two parts as advised in the project: First, data before 2016 is separated for testing and training of ML model. Second, rest of the data has been saved in new dataframe as unseen data.

## Feature Conversion:

Machine Learning algorithms can only be trained on numerical data. At this point, in the dataframe “journal” is nominal variable which cannot be used to train the forecasting model. Therefore, it has to be converted into numerical. To convert the categorical feature into numerical we had following options.

* label encoding

Label encoding converts the labels into numeric form to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

* dummy encoding

Dummy coding provides one way of using categorical predictor variables in various kinds of estimation models such as, linear regression. Dummy coding uses only ones and zeros to convey all of the necessary information on group membership.

* One hot encoding

In this strategy, each category value is converted into a new column and assigned a 1 or 0 (notation for true/false) value to the column. Let’s consider the previous example of bridge type and safety levels with one-hot encoding.

Chart, scatter chart

Description automatically generated

Figure t-sne Applied After Encoding

As advised in the project guidelines, t-sne is applied on the dataset after applying encoding. As the above figure shows that the data have been clustered into many small patches, which shows the dimension reduction of the t-sne algorithm.

All the above three techniques have been applied one by one on dataframe. It has been found that the accuracy with dummy encoding and one-hot encoding was relatively much less than label encoding. Therefore, in the whole project label encoding has been applied.

Table

Description automatically generated

Figure Dataset Snippet

# Prediction Models on Articles:

## Linear Regressor

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

****

Figure Accuracy of Linear Regressor

After pre-processing the data, linear regressor have been applied on the test data. As the growth of data in dblp is exponential. The accuracy of linear regressor is only 24%. The reason of low accuracy of the linear regressor is that the linear regressor works on linear data but in this case the growth of data is exponential. Therefore, linear regressor did not performed well in the accuracy.

## Decision Tree Regressor

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

**Graphical user interface, text, application, email

Description automatically generated**

Figure Decision Tree Regressor

As advised in the project, the testing and training data have been obtained from the data before 2016. In this model testing and training data has been portioned with 80%-20% ratio for testing and training. The results (87% accuracy) are much improved with respect to linear regressor.

## Decision Tree Regressor with K-fold

K-Fold CV is where a given data set is split into a K number of sections/folds where each fold is used as a testing set at some point. Let’s take the scenario of 5-Fold cross validation(K=5). Here, the data set is split into 5 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, 2nd fold is used as the testing set while the rest serve as the training set. This process is repeated until each fold of the 5 folds have been used as the testing set.

Graphical user interface, text, application

Description automatically generated

Figure Decision Tree Regressor with K-fold

Decision tree regressor with k-fold have outperformed simple decision tree regressor with the accuracy of 89%. Which is still good accuracy but to get better accuracy we have tried random forest regressor, which has been described next.

## Decision Tree Regressor (Unseen Data)

Previously, the predictions were made on the data before 2016. Now, the models are being tested on unseen data (after 2016).

Graphical user interface, text, application, email

Description automatically generated

Figure Decision tree on unseen data

The accuracy of decision tree on unseen data can be seen which is only 72%. Keeping in mind that the growth of dblp dataset is exponential. Therefore, the results may be compromised.

## Decision Tree Regressor with K-fold (Unseen Data)

The prediction with decision tree was not impressive. Therefore, K-fold is being used to try if the accuracy is improved.

Graphical user interface, text, application, email

Description automatically generated

Figure Decision Tree with K-fold on unseen data

The results shows that there is significant 18% improve in the accuracy of decision tree when k-fold is applied.

## Random Forest Regressor (on testing data with k-fold)

After applying decision tree with k-fold on seen and unseen data. Random forest is also applied on the data to analyze if there is any change in the accuracy of the testing data.Graphical user interface, text, application

Description automatically generated

Figure Random Forest (also contain k-fold results)

It can be observed that the accuracy of random forest is 88%. On the other hand when cross validation is applied the accuracy is increased by 1% reaching at 89% on the testing dataset.

## Random Forest Regressor (on unseen data with k-fold)

After applying random forest on training data and testing their results, the model has also been tested on the unseen data. The results of the random forest with and without k-fold can be seen below:

Graphical user interface, text, application, email

Description automatically generated

Figure Random Forest Regressor (includes cross-validation)

It can be observed that the accuracy of random forest on unseen data is 66%. On the other hand when cross validation is applied the accuracy is increased by 2% reaching at 68% on the unseen dataset.

# Model Training on In Proceeding

## Decision Tree Regressor

A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

**Graphical user interface, text, application, email

Description automatically generated**

Figure Decision Tree Regressor with and without k-fold

As advised in the project, the testing and training data have been obtained from the data before 2016. In this model testing and training data has been portioned with 80%-20% ratio for testing and training. The results show that the accuracy of in proceeding with decision tree is 74%. On the other hand when k-fold is applied the accuracy is reduced to 71%.

## Decision Tree Regressor (Unseen Data)

Previously, the predictions were made on the data before 2016. Now, the models are being tested on unseen data (after 2016).

Graphical user interface, text, application, email

Description automatically generated

Figure Decision tree on unseen data

The accuracy of decision tree on unseen data can be seen which is only 70%. Keeping in mind that the growth of dblp dataset is exponential. After applying k-fold on unseen data the accuracy remains same at 70%. Which shows that different techniques apply on different types of data.

## Random Forest Regressor (on testing data with k-fold)

After applying decision tree with k-fold on seen and unseen data. Random forest is also applied on the data to analyze if there is any change in the accuracy of the testing data.Graphical user interface, text, application

Description automatically generated

Figure Random Forest (also contain k-fold results)

It can be observed that the accuracy of random forest is 82%. On the other hand when cross validation is applied the accuracy is decreased by 9% reaching at 71% on the testing dataset.

## Random Forest Regressor (on unseen data with k-fold)

After applying random forest on training data and testing their results, the model has also been tested on the unseen data. The results of the random forest with and without k-fold can be seen below:

Graphical user interface, text, application, email

Description automatically generated

Figure Random Forest Regressor (includes cross-validation)

It can be observed that the accuracy of random forest on unseen data is 74%. On the other hand when cross validation is applied the accuracy is decreased by 7% reaching at 67% on the unseen dataset.

# Comparison of techniques

## Accuracies of Articles

It can be observed that the accuracies are improved after implementing k-fold validation.

|  |  |  |
| --- | --- | --- |
| Models | Without k-fold (%) | With k-fold (%) |
| Decision tree | 86.92 | 89.76 |
| Random Forest | 88.40 | 89.76 |

Table Testing data Accuracies

It can be observed that after applying k-fold validation on the unseen data the accuracies jumps from 72.5% to 90% and 66.6% to 68.9% in decision tree and random forest respectively.

|  |  |  |
| --- | --- | --- |
| Models | Without k-fold (%) | With k-fold (%) |
| Decision tree | 72.51 | 90.82 |
| Random Forest | 66.65 | 68.09 |

Table Unseen data Accuracies

## Accuracies of In Proceedings

It can be observed that after applying k-fold validation on the testing data the accuracies decrease from 74.25% to 71.59% and 74.24% to 67.93% in decision tree and random forest respectively.

|  |  |  |
| --- | --- | --- |
| Models | Without k-fold (%) | With k-fold (%) |
| Decision tree | 74.25 | 71.59 |
| Random Forest | 74.24 | 67.93 |

Table Testing data Accuracies

It can be observed that after applying k-fold validation on the unseen data the accuracies remains same in decision tree at 70.96% decrease from 82.63% to 71.59% in decision tree and random forest respectively.

|  |  |  |
| --- | --- | --- |
| Models | Without k-fold (%) | With k-fold (%) |
| Decision tree | 70.96 | 70.82 |
| Random Forest | 82.63 | 71.59 |

Table Unseen data Accuracies

# Combined Data

Now the datasets of articles and in proceedings have been combined to see any difference. Following are the results that have been found from the models. It can be observed that after applying k-fold validation on the testing data the accuracy decreases from 91.4% to 88.86% and increase from 85.32% to 88.86% in decision tree and random forest respectively.

|  |  |  |
| --- | --- | --- |
| Models | Without k-fold (%) | With k-fold (%) |
| Decision tree | 91.42 | 88.86 |
| Random Forest | 85.32 | 88.86 |

Table Combined data accuracy on testing data

It can be observed that after applying k-fold validation on the testing data the accuracy increase from 72.55% to 89.94% and 66.08% to 75.81% in decision tree and random forest respectively.

|  |  |  |
| --- | --- | --- |
| Models | Without k-fold (%) | With k-fold (%) |
| Decision tree | 72.55 | 89.94 |
| Random Forest | 65.08 | 75.81 |

Table Combined Data Accuracy on unseen data

# Conclusion

In the end decision tree performed better than linear regression and random forest. Furthermore, it has been found that hot-encode and dummy encoding did not performed well with the algorithms.

# References

<https://github.com/IsaacChanghau/DBLPParser>

https://dblp.org/