

The Republic of North Macedonia's Research Ranking Platform for Academic Staff and Universities

Arbnor Rushiti¹, Artan Luma¹, Ylber Januzaj², Azir Aliu¹, Halil Snopçe¹,
Anis Sefidanoski³, Mitko Trpkoski⁴

¹ Faculty of Contemporary Sciences and Technologies, SEEU University, Tetovo, Republic of North Macedonia

² University "Haxhi Zeka", Peja, Kosovo

³ Faculty of Applied IT, Machine Intelligence and Robotics, University of Information Science and Technology (UIST) "St. Paul the Apostle", Ohrid, Republic of North Macedonia

⁴ Faculty of Information and Communication Technologies, University "St. Kliment Ohrid-ski", Bitola, Republic of North Macedonia

Abstract – In the present-day realm of higher education, the integration of open data and machine learning has become a crucial means of improving the evaluation and ranking of research publications by university professors. This research endeavours to address the pressing need for an effective computerized paradigm that combines the collection, analysis, and use of public data in higher education. The main objective is to provide recommendations in the form of informative reports for interested institutions and to rank university professors' scientific works on an open data platform. This research not only contributes to the advancement of educational evaluation methodologies but also responds to the growing importance of open data and machine learning in modern higher education. By automating the assessment of research publications, universities can better track and enhance their academic impact, ultimately fostering a culture of excellence in research and innovation.

Keywords – higher education, ranking, big data, machine learning, open data.

DOI: 10.18421/SAR71-01

<https://doi.org/10.18421/SAR71-01>

Corresponding author: Arbnor Rushiti,
Faculty of Contemporary Sciences and Technologies, SEEU
University, Tetovo, Republic of North Macedonia

Email: ar26730@seeu.edu.mk

Received: 08 January 2024.

Revised: 14 February 2024.

Accepted: 19 February 2024.

Published: 27 March 2024.



© 2024 Arbnor Rushiti et al; published by
UIKTEN. This work is licensed under the CC BY-NC 4.0

The article is published with Open Access at
<https://www.sarjournal.com>

1. Introduction

Based on the preliminary research, the idea was born to develop a ranking system which would present the results for third parties of interest regarding the publications of university professors. Developing a machine learning-based system for university ranking based on the impact of research papers published by academic staff is a challenging IT problem [1].

1.1. The Life-cycle of the System for Machine Learning University Ranking

Here are the 7 general steps we are going to follow in this thesis to approach the problem:

Data collection: Collect data on research publications from different universities. This data should include the name of the university, the name of the author, and the title of the paper, the publication venue, the publication year, and the citation count.

Feature engineering: Take pertinent features from the data that can help in ranking universities. Features may include the quantity of citations and publications, the citation per paper, the h-index, the number of co-authors and the publication venue's impact factor.

Data preprocessing: Purify the data by eliminating duplicates, fixing mistakes, and adding missing values.

Model selection: Select an appropriate predictive machine learning model of the impact of research papers and rank universities based on this impact. Some possible neural networks, decision trees, random forests, and linear regression are examples of models.

Training the model: Use the collected data to train the selected model. Divide the data into training and testing sets, and then assess the model's performance using cross-validation.

Following is the Python code for data preprocessing, i.e. splitting into appropriate chunks for further training:

```
import pandas as pd
# read CSV file into a pandas DataFrame
df = pd.read_csv('/home/Rank/dataset.csv')
df['Year'] = df['Year'].astype('category')
df['Author'] = df['Author'].astype('string')
print(df.info())
print(df['Year'].cat.categories)
test_length = 29 * 10
train_length = df.shape[0] - test_length
df_train = df.head(train_length)['Paper']
df_test = df.tail(test_length)['Paper']
# write filtered DataFrames to new CSV files
df_train.to_csv('/home/Rank/dataset_train.csv',index=False)
df_test.to_csv('/home/Rank/dataset_test.csv',index=False)
```

Validation: Examine the model's predictions against the actual impact of research articles to validate it.

Ranking universities: Once the model is validated, use it to rank universities based on the impact of their research papers.

The first step is to collect data on research publications from different universities. There are several sources of such data, including academic search engines, scholarly databases, and university websites. Making sure the data is precise, comprehensive, and trustworthy is crucial.

The second step, Feature engineering is the process of identifying pertinent traits from the gathered data that may be used to rank colleges. Some of the features that can be used in this context are:

Number of publications: This feature counts the number of research papers published by the academic staff of a university.

Number of citations: This feature counts the total number of citations received by the research papers published by the academic staff of a university.

Citation per paper: This function determines the typical quantity of citations that each research paper published by the academic staff of a university.

H-index: This feature is a metric that considers the quantity of publications in addition to the quantity of citations.

It measures the impact of an academic according to the quantity of publications they have authored and the quantity of citations those publications have obtained.

Impact factor of the publication venue: This feature takes into account the quality of the publication venue where the research paper was published. The average number of citations that papers published in that venue obtain is measured by the impact factor.

Number of co-authors: This feature measures the number of co-authors who contributed to a research paper. It can be an indicator of collaboration and the level of research activity in a university.

The third step, Data preprocessing, involves cleaning the gathered data by eliminating duplicates, fixing mistakes, and adding missing values. This is an important step for ensuring that the data is accurate and reliable. Some common techniques used in data preprocessing include:

Data cleaning serves to remove duplicates, correcting errors, and removing irrelevant data. Transforming the data into a format that the machine learning model can use is known as data transformation. As an illustration, one instance of feature engineering involves the transformation of textual data into numerical representations. Data normalization on the other hand involves scaling the data to ensure that each feature has a similar range of values.

Model selection is the fourth and most important step. Model selection involves selecting an appropriate machine learning model capable of predicting impact of research papers and rank universities based on this impact. Some possible models include:

Linear regression: This model is used to predict a continuous target variable based on a set of predictor variables.

Training the model is the action which transforms the design created by the previous phase into a working system [2]. This is similar to the development of the software engineering life cycle. Once the machine learning after the model has been chosen, the gathered data have to be used to train the model. The model is trained on the training set of data after the data is divided into training and testing sets. Cross-validation is used to assess the model's performance.

We import Tensorflow, Pandas and Numpy to train the model.

Our algorithm is the following:

```
import tensorflow as tf
import pandas as pd
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,
Dropout, Bidirectional, LSTM, Conv1D,
ConvLSTM1D, MaxPooling1D,
BatchNormalization, Activation, Flatten, Reshape
from tensorflow.keras.regularizers import l1, l2
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import
LearningRateScheduler
YEAR_COUNT = 29
def trainModel(EPOCHS, BATCH_SIZE):
    df=
pd.read_csv('/home/Rank/dataset_train.csv')
    trainX = df[((df.index + 1) % YEAR_COUNT)
!= 0]
    trainY = df.iloc[(YEAR_COUNT -
1)::YEAR_COUNT, :]

trainX = np.array(trainX)
trainX=trainX.reshape(trainY.shape[0],
trainX.shape[0]//trainY.shape[0], 1)
def lr_schedule(epoch):
    lr = 0.0004
    # if epoch > EPOCHS//4:
    #     lr = 0.0002
    # elif epoch > EPOCHS//2:
    #     lr = 0.0001
    # return lr
    model = Sequential([
        LSTM(
            10
        ),
        Dense(1, activation = 'relu')
    ])
    optimizer = Adam(learning_rate=0.0004)
    model.compile(
        optimizer = optimizer,
        loss = tf.keras.losses.MeanAbsoluteError(),
        metrics = [tf.keras.metrics.MeanSquaredError()]
    )
    model.compile(
        optimizer = optimizer,
        loss = tf.keras.losses.BinaryCrossentropy(),
        metrics = [tf.keras.metrics.AUC()]
    )
    model.build((1, YEAR_COUNT, 1))
    model.summary()
    checkpointer=
tf.keras.callbacks.ModelCheckpoint(
    filepath = '/home/Rank/rank.keras',
    save_weights_only = False,
    save_freq = 'epoch',
    save_best_only = False
```

```
)
    history = tf.keras.callbacks.History()
    lr_scheduler=
LearningRateScheduler(lr_schedule)
    model.fit(
        trainX,
        trainY,
        batch_size = BATCH_SIZE,
        epochs = EPOCHS,
        callbacks = [checker, history],
        validation_split = 0.15
    )
    trainModel(500, 3)
```

The sixth step involves the process of validation. To guarantee that the model is accurate and dependable, validation is required after training. This is done by comparing the predictions of the model with the actual impact of research papers.

Last but not least, we got the ranking of universities. Once the model is validated, it can be used to rank universities based on the impact of their research papers. The universities are ranked based on their predicted impact score, which is calculated using the selected features and the trained machine learning model [3].

2. State of the art

In recent years, the incorporation of Artificial Intelligence (AI) into higher education has led to significant advancements in the way we evaluate and rank educational institutions. AI-powered ranking systems have emerged as a promising approach to provide more accurate, data-driven, and fair assessments of universities and colleges. This research paper delves into the development, challenges, and implications of AI-powered ranking systems in higher education. It explores how these systems leverage AI techniques to enhance ranking accuracy and fairness while addressing potential ethical and bias concerns. This research proposal provides a structured framework for conducting a comprehensive study on enhancing higher education ranking systems through the integration of Artificial Intelligence. Researchers can use this outline to develop a detailed research plan and project.

The first real project of artificial intelligence dates back to 1943, when two researchers, McCulloch and Pitts, brought to light the first artificial neuron [4]. This system employed a model of artificial neurons, in which the status of these neurons changed from off to on in the presence of stimuli caused by a sufficient number of surrounding neurons. McCulloch-Pitts paradigm then came to show that any mathematical function can be represented by some neuron network, and that all logical connectives can be implemented by a simple neural structure.

The first formally recognized neural network was instead created seven years later, in 1950, by two students from Harvard University named Marvin Minsky and Dean Edmonds. These early works, however, did not arouse great interest in the public, who did not yet understand how technology could be applied to solve real problems. The studies of Alan Turing, mathematician, logician, cryptographer and British philosopher who in 1950 brought the debate to a philosophical level hitherto unexplored, were much more sensational. The question Turing sought to address was whether a machine can exhibit the capability to think. The question, as at the time, even nowadays is able to create discussions, conflicting views and cause a certain anxiety among the people. To try to answer this question, Turing invented a behavioral test called the imitation game and also remembered as the Turing test. This test allows one to measure whether or not a machine is able to think, if it can do it autonomously, and above all if it is able to do it exactly like a human being. The test is carried out like this: there are three people, indicated by ease as a, b and c. A is a man, b a woman and c is the third individual who does not care about the sex and has the role of interrogator. Player c has the task of guessing, by asking a series of questions, the sex of a and b. A, however, will have to try to deceive c, while b will have to try to help him solve the question. The players are in three different rooms without the possibility to get in touch with each other and the answers to the questions have to be provided with a typewriter, to avoid that the handwriting and the voice can help us find the solution. The game takes place in two steps; the first time the game is played as described in the previous lines, while the second time the person replaces a machine. If c after this substitution does not notice anything then she should be considered as intelligent as a human being. According to this test the machine is therefore defined as intelligent if the frequency with which the interrogator correctly identifies the man and the woman is at least the same with which he correctly identifies the machine and the woman. Alan Turing predicted that by 2050 machines will be able to pass his test and therefore will be the date on which a machine can be considered intelligent.

3. Research Methodology

As we mentioned above, the crucial stage in creating a machine learning-based system for university ranking is choosing the model. The intention of model selection is selecting a model that can accurately predict the impact of research papers and rank universities based on this impact.

There are many variations machine learning models available, each with its own strengths and weaknesses. The nature of the data, the difficulty of the issue, and the required degree of accuracy all influence the model selection.

3.1. The Creation of the AI Model

The most well-liked [5] machine learning models that can be used for university ranking:

Linear regression: A popular and straightforward model for predicting a continuous target variable from a set of predictor variables is called linear regression. It assumes that there is a linear connection between the predictor variables as well as the intended variable. Linear regression is able to make predictions on the impact of research papers based on features such as the number of citations, the number of co-authors, and the publication venue.

Decision trees: Among the most widely used models for predicting a categorical goal differ based on a set of predictor variables. They construct a model of decisions and potential outcomes that resembles a tree. Decision trees is able to predict the impact of research papers depending on characteristics like as the number of publications, the impact factor of the publication venue, and the h-index [6].

Random forests: Decision trees are extended by random forests, which employ numerous decision trees to increase prediction accuracy. They create a large number of decision trees, each based on a random subset of the features. One can use random forests to predict the impact of research papers based on multiple features, including the quantity of publications, the number of citations, and the h-index.

Neural networks: Neural networks are a powerful model for predicting a desired variable determined by a set of input variables. They consist of multiple layers of interconnected nodes that process the input data. Neural networks can be used to predict the impact of research papers based on features such as the number of publications, the number of citations, and the h-index.

It is crucial to remember that there is not a single formula that works for all university rankings [7]. The selection of model depends on the specific requirements of the problem, like the kind of data, the number of features, and the desired level of accuracy. Thus, we choose the third and most powerful one, based on neural networks.

Neural network models differ from other machine learning models in several ways.

First, neural networks are designed to model complex, nonlinear connections between the goal variable and the predictor variables [8]. This means that they can handle data that is not easily represented by linear models or decision trees.

Second, neural networks are capable of learning from large volumes of information and are able to recognize patterns and relationships that may be difficult for humans to detect. This is because neural networks may eliminate the need for human feature engineering by automatically extracting pertinent characteristics from the data.

Third, neural networks are useful for many different kinds of tasks, such as speech and picture recognition, natural language processing, etc., and time-series forecasting. This versatility makes them a popular choice for many different types of problems [9].

Finally, neural networks can be very powerful and flexible, but also require careful tuning and training to ensure optimal performance. This can make them more difficult to work with than simpler models such as regression or decision trees. Overall, neural network models offer a powerful tool for building machine learning systems, but their complexity and computational requirements have to be carefully considered before selecting them as a modeling approach.

Neural network-based models are an effective modeling complex tool, non-linear relationships between predictor variables and the target variable. There are several types of neural network-based models to choose from, including feed-forward convolutional neural networks, recurrent neural networks, and neural networks. We have to remember that the neural network-based model is just one component of the overall system for university ranking. We will also need to consider how to integrate the model with other components such as data collection, data processing, and user interface design [10].

As a machine learning framework will use TensorFlow with Keras to create the neural network-based model for university ranking. In fact, Keras is a popular high-level API for building deep learning models in TensorFlow. It provides a user-friendly interface for creating and training neural networks, making it easier for developers to build complex models without requiring a deep understanding of the underlying implementation details. Keras allows us to define the architecture of our neural network using a set of pre-defined layers and activation functions. One can also customize the architecture by adding our own layers and defining our own activation functions.

Once we have defined the architecture of our neural network, we can compile it and train it on our data using a variety of optimization algorithms [11].

Keras also provides a range of utilities and tools for monitoring and evaluating the performance of our model during training. This includes metrics such as accuracy, loss, and validation accuracy, which can help us to identify and diagnose issues with our model. Overall, using TensorFlow with Keras can provide a powerful and flexible platform for building and training deep learning models, including neural networks for university ranking.

4. Proposed Model

Integrating the neural network-based model with other components such as data collection, data processing, and user interface design is essential to create a complete machine learning system for university ranking. To integrate the neural network model with data collection, we need to determine the sources of data that will be used to train and test the model. This may include academic databases such as Scopus or Web of Science, or university ranking websites such as Shanghai Top University List. We will also need to determine the criteria for selecting the data, such as the time period, the publication type, or the academic field.

4.1. Wrap-up: Integration of the Model with the Rest of the System

Once we have collected the data, we will need to preprocess it to get it ready for the neural network model to use. This may include tasks such as removing duplicates, standardizing the data, and converting categorical variables to numerical variables [12]. We may also need to perform feature engineering, pertinent features from the data are extracted so they may be fed into the neural network model. To integrate the neural network model with user interface design, we will need to create a user interface that allows users to interact with the system and view the results of the university ranking. This may include designing a web application or a mobile application that provides a graphical user interface for users to input their preferences and view the results. We may also need to incorporate visualizations or other interactive features to help users understand the results and compare universities. Once the neural network model has been trained and evaluated, we will need to deploy it in a production environment so that it can be used to rank universities in real-time.

This may involve setting up a server or cloud-based service that can handle the computational requirements of the model and provide a scalable and reliable infrastructure for users. Overall, integrating the neural network-based model with data collection, data processing, and user interface design requires careful planning and coordination to ensure that all components work together seamlessly.

Following is the code we wrote in Python to preprocess the academic paper data gathered from the public repositories of several universities from North Macedonia.

```
import pandas as pd
import numpy as np
df =
pd.read_csv('/home/rumi/Desktop/Rank/total.csv')
df = df.drop(columns = ['Nr', 'Keywords',
'Publica-tionLink', 'Publisher'])
typeList = ['conference', 'journal', 'proceeding',
'article']
for current_type in typeList:
    df['Type'] = df['Type'].apply(lambda x :
current_type if type(x) != float and current_type in
x.lower() else x)
    df['Type'] = df['Type'].apply(lambda x : 'journal'
if type(x) != float and 'journa' in x.lower() else x)
    df['Type'] = df['Type'].apply(lambda x : 'research'
if type(x) != float and
('research' in x.lower() or 'scientific' in x.lower()
or 'paper' in x.lower()) else x)
    df['Type'] = df['Type'].apply(lambda x : 'article' if
type(x) != float and ('review' in x.lower() or 'article'
in x.lower() or 'online' in x.lower()) else x)
    df['Type'] = df['Type'].apply(lambda x : 'program'
if type(x) != float and ('wor-shop' in x.lower() or
'programe' in x.lower()) else x)
SEEU_bad_spelling = ['South East European
University ', 'South East European University ',
'South East European University']
df['Affiliation'] = df['Affiliation'].apply(lambda x :
'South East European Uni-versity' if
SEEU_bad_spelling.count(x) > 0 else x)
df['Affiliation'] = df['Affiliation'].astype('category')
KlimentOhridski_bad_spelling = ['Ss. Cyril and
Metho-dius University', 'Sv.Kliment Ohridski', 'ss.
Cyril and Methodius University', 'Sv.Kliment
Ohridski']
df['Affiliation'] = df['Affiliation'].apply(lambda x :
'Sv. Kliment Ohridski' if KlimentOhrid-
ski_bad_spelling.count(x) > 0 else x)
df['Affiliation'] =
df['Affiliation'].astype('category')
#converting columns to proper types
df['Author'] = df['Author'].astype('string')
df['Academic Title'] = df['Academic Ti-
tle'].astype('category')
```

```
df['Year'] = df['Year'].apply(lambda x : '2006' if x
== '2006-2007' else x) #po-so nekoj pameten se
prase
df['Year'] = df['Year'].astype('category')
df['Type'] = df['Type'].astype('category')
df['Primary Author'] = df['Primary Au-
thor'].apply(lambda x : True if x == 'Yes' else False)
df['Primary Author'] = df['Primary Au-
thor'].astype('boolean')
df['Second Author'] = df['Second Au-
thor'].apply(lambda x : True if x == 'Yes' else False)
df['Second Author'] = df['Second Au-
thor'].astype('boolean')
df['Impact Factor'] = df['Impact
Factor'].apply(lambda x : True if x == 'Yes' else
False)
df['Impact Factor'] = df['Impact Fac-
tor'].astype('boolean')
dataset = df.groupby(['Author',
'Year'])['Paper'].count().sort_values()
dataset = dataset.reset_index()
dataset = dataset.sort_values(by = ['Author',
'Year'])
dataset = dataset.reset_index()
dataset = dataset.drop(columns=['index'])
da-ta-
set.to_csv('/home/rumi/Desktop/Rank/dataset_author
.csv', index = False)
print(dataset.info())
print(dataset)
#print(df['Author'].nunique())
#print(df['Year'].nunique())
#print(df.groupby(['Affiliation'])['Paper'].count().
sort_values()).
```

4.2. Wrap-up: Integration of the Model with the Rest of the System

When creating a machine learning model, validation is a crucial stage since it lets us assess the model's effectiveness and ensure that it can generalize well to new data. Some common approaches to model validation are the following:

Holdout validation: The supplied data is divided into a training set and a validation set in this method. The validation set is used to assess the model after it has been trained on the training set. This enables us to evaluate the model's performance using previously unseen data.

Cross-validation: In this approach, the available data is divided into multiple folds. The process is repeated k times, with the model being trained on k-1 folds and validated on the remaining fold. This enables us to evaluate the model's performance across several data subsets.

Leave-one-out validation: In this approach, the model has been trained using one every data point, and the performance is evaluated on the left-out data point. Every data point goes through this procedure once again, and the average is calculated. This allows us to evaluate the model's performance on data that is very similar to the data it was trained on.

Bootstrapping: In this approach, multiple training sets are sampled with replacement from the available data, and the model is trained on each of these sets. The performance is evaluated on the remaining data. This allows us to evaluate whether the model's performance is consistent across several data subsets.

In addition to these validation approaches, we can also use methods like early stopping, regularization, and changing of hyper parameters to improve the performance of the model and prevent overfitting.

The choice of validation approach for university ranking system depends on several of parameters, including the size of our dataset, the complexity of our neural network model, and the specific requirements of our project. If we have a large dataset and a simple neural network model, holdout validation may be a good choice. In this approach, we can divide our dataset into two subsets: a validation set (used to assess the model's performance) and a training set (used to train the model). The size of the validation set 80% of the data is utilized for training and 20% is used for validation in an 80/20 split, though this may vary depending on the size of our dataset.

If we have a smaller dataset or a more complex neural network model, cross-validation may be a better choice. In this approach, we can divide our dataset into k-folds and train the model k times, each fold serving as validation data once, and the remaining k-1 folds serving as training data. The results are then averaged to obtain an estimate of the model's performance [13].

Leave-one-out validation may be a good choice if we have a very small dataset or if we want to test the generalizability of the model to fresh data points that are very similar to the training data. Bootstrapping may be useful if we want to evaluate the performance stability of our model on various data subsets.

As we are going to deal with relatively small dataset in the pilot version of the system, due to the limited number of universities/papers in North Macedonia, we have decided to use cross-validation for model validation in our university ranking system. Here are some general steps we can follow:

Divide our dataset into k-folds, where k is typically between 5 and 10. Each fold should have approximately the same size.

Train the neural network model k times, with the remaining k-1 folds utilized for training, and each fold employed once as the validation data. This means that we will train k different models.

Analyze each model's performance using the validation set. This will give us k different measures of performance.

Calculate the average performance across the k models to determine how well the model performed across the whole dataset. This is often referred to as the cross-validated performance.

5. Results

Once we have selected the best model based on cross-validated performance, we can train it on the entire dataset and use it to forecast using fresh data. In addition, it is important to properly partition the dataset into training and validation sets for each fold in order to prevent overfitting. We can use further techniques such as stratified sampling or random shuffling to ensure that the data is partitioned randomly and evenly across the folds.

Preprocessing:

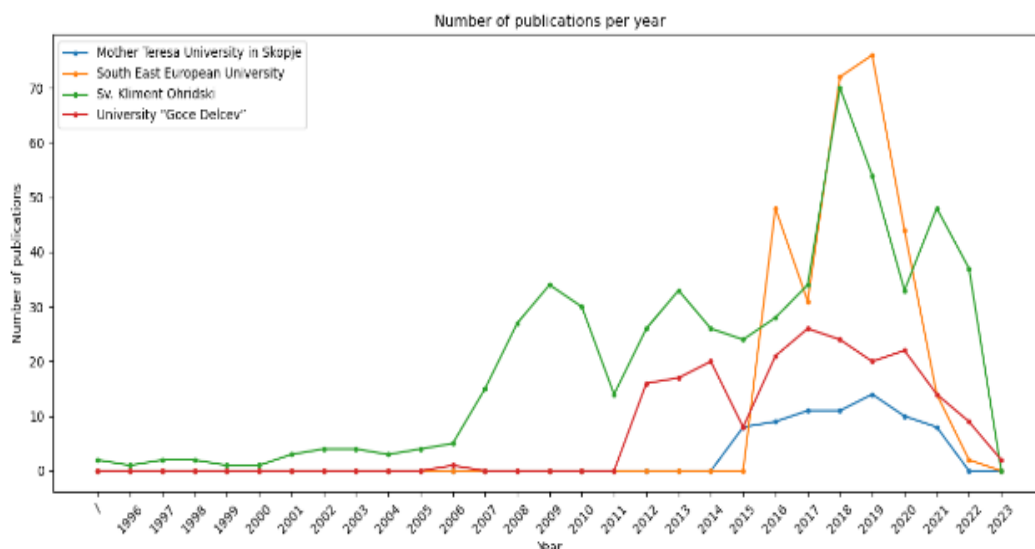
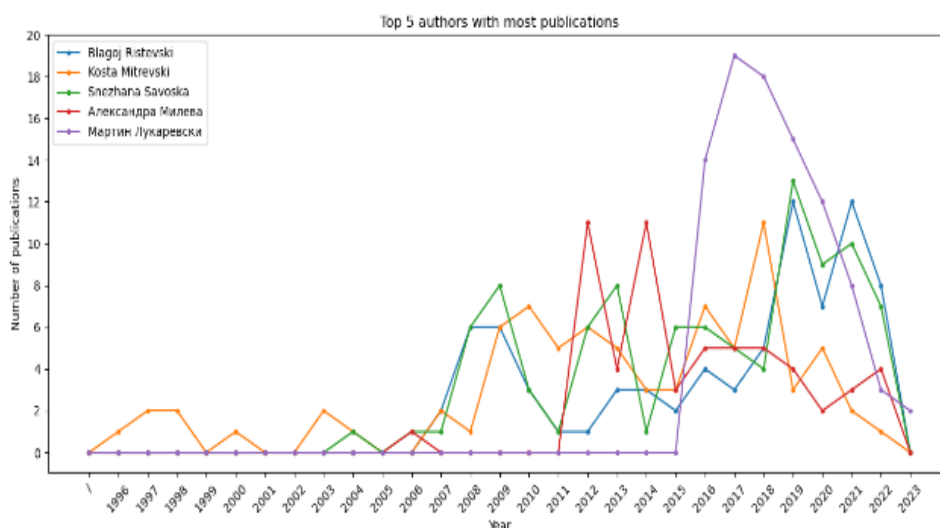
- Removing inconsistencies with naming of paper types e.g. 'Scientific paper' and 'Paper';
- Both classified as 'research'.
- Removing inconsistencies with naming of universities.
- Converting Primary Author, Second Author and Impact Factor to Boolean type.
- Converting Affiliation, Type, Academic Title and Year to categorical type.
- Converting Author to string type.
- This was done to lower memory consumption and speed up training.
- The dataset for the machine learning model contains 3 columns.
- [Author, Year, Number of publications]
- Test and train data split:
- Test data contains 290 entries (10 people * 29 years).
- The train data has a validation split of 15%.

Model:

- The model receives data from 1996 to 2022 for an individual and predicts
- Number of publications that they will release in 2023.
- Due to small amount of data the model only uses 1 hidden layer.
- LSTM (Long short term memory) with 10 units.

Visualization:

- In Figure 1 we have presented the number of publications per year for every university.
- In Figure 2 we have presented the number of publications per year for the top 5 individuals who have made the most publications.

Number of publications per year for every university.*Figure 1. Number of publications per year**Number of publications per year for the top 5 individuals who have made the most publications.**Figure 2. Top 5 authors with most publications***6. Conclusions and Recommendations**

An automated methodology for combining the gathering, analysis, and application of open data in higher education is presented in this work.

By using a novel and efficient automated model for integrating the gathering, analysis, and utilization of open data in higher education, universities may be enhanced to more effectively rate and review research articles, which is very beneficial in open data education.

In conclusion, the creation of a machine learning model for university ranking based on the impact of research papers published by academic staff involves several key steps. These include data collection, data processing, feature engineering, model selection, and validation. A neural network-based model was selected for this particular job because of its capacity to manage intricate data structures and nonlinear feature interactions. TensorFlow with Keras was suggested as a suitable tool for implementing the model.

For model validation, cross-validation is recommended as an effective method for assessing the model's performance on the entire dataset, while concurrently mitigating the risk of overfitting. Overall, the successful implementation of a machine learning model for university ranking can provide valuable insights and aid in decision-making processes. However, it is crucial to confirm that the model is validated using appropriate techniques and procedures, and that it is continually updated and refined as new data becomes available.

References:

- [1]. Umar, R., Yudhana, A., & Dernata, J. (2022). The Admission Decision Support System for Muhammadiyah Student Association Cadres Using the Profile Matching Method. *JUITA: Jurnal Informatika*, 10(1), 53-58.
- [2]. Kassen, M. (2017). Open data in Kazakhstan: Incentives, implementation and challenges. *Information Technology & People*, 30(2), 301-323. Doi:10.1108/ITP-10-2015-0243
- [3]. Mohler, G., Porter, M., Carter, J., & LaFree, G. (2020). Learning to rank spatio-temporal event hotspots. *Crime Science*, 9(1), 1-12
- [4]. Gleicher, M., Barve, A., Yu, X., & Heimerl, F. (2020). Boxer: Interactive comparison of classifier results. *Computer Graphics Forum: Journal of the European Association for Computer Graphics*, 39(3), 181-193. Doi:10.1111/cgf.13972
- [5]. Zhang, J., Wang, Y., Molino, P., Li, L., & Ebert, D. S. (2018). Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics*, 25(1), 364-373. Doi:10.1109/TVCG.2018.2864499
- [6]. Yeh, A., & Ngo, A. (2021). Bringing a ruler into the black box: uncovering feature impact from individual conditional expectation plots. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 34-48. Cham: Springer International Publishing.
- [7]. Zubaedah, R., & Prasetyo, H. (2022). Decision support system for high school entrance selection. *Brilliance: Research of Artificial Intelligence*, 2(2), 71-74. Doi:10.47709/brilliance.v2i2.1566
- [8]. Maack, R. G. C., Scheuermann, G., Hagen, H., Peñaloza, J. T. H., & Gillmann, C. (2023). Uncertainty-aware visual analytics: scope, opportunities, and challenges. *The Visual Computer*, 39(12), 6345-6366. Doi:10.1007/s00371-022-02733-6
- [9]. Gomez, O., Holter, S., Yuan, J., & Bertini, E. (2021, October). Advice: Aggregated visual counterfactual explanations for machine learning model validation. In *2021 IEEE Visualization Conference (VIS)*, 31-35. IEEE. Doi: 10.48550/arXiv.2109.05629
- [10]. Yuan, J., Chan, G. Y.-Y., Barr, B., Overton, K., Rees, K., Nonato, L. G., ... Silva, C. T. (2022). SUBPLEX: A visual analytics approach to understand local model explanations at the subpopulation level. *IEEE Computer Graphics and Applications*, 42(6), 24-36. Doi:10.1109/MCG.2022.3199727
- [11]. Heer, J. (2019). Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences of the United States of America*, 116(6), 1844-1850. Doi:10.1073/pnas.1807184115
- [12]. Dasgupta, A., Wang, H., O'Brien, N., & Burrows, S. (2020). Separating the wheat from the chaff: Comparative visual cues for transparent diagnostics of competing models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 1043-1053. Doi:10.1109/tvcg.2019.293454
- [13]. Esquivel, D. J. A., & Esquivel, D. J. A. (2021). A Machine Learning Based DSS in Predicting Undergraduate Freshmen Enrolment in a Philippine University. *arXiv preprint arXiv:2108.07690*.