An Empirical Study of AI related open-source projects

Lakshay Kalra

Course: Industrial Software Engineering

Lecturer: Dr.Sridhar Chimalakonda
Mentor:Venigalla Akhila Sri Manasa

Abstract

We have entered into the third decade of the 21st century where Artificial Intelligence designs almost every aspect of our life and most of its tasks are accomplished by using machine learning or deep learning algorithms such as autonomous cars, medical assistance by AI agents. Many of these algorithms/models are available to machine learning enthusiasts/developers without worrying about license and cost as open-source which provides the flexibility to understand, use and modify exclusive of reinventing the wheel. But today we have several such resources which creates an ambiguity of what and which project to choose?
This study aims to give answers to these questions by doing quantitative research.

1 Keywords

Machine Learning, Deep Learning, Repository, Issues, Commits, Topics.

2 Introduction

Artificial Intelligence is emerging to be a large and powerful body that has shown its application in various fields ranging from health care, education to Data security. Nowadays, it is almost impossible to name a place where it is not used.AI is a big banner under which there are two major domains namely

machine learning and deep learning which undertake almost every known aspect of it.
With the help of deep learning algorithms, automation has risen to a new level and looks promising in shaping a better future. Many of the real-world problems are being solved. On the other hand, machine learning algorithms are widely used to counter some other problems such as pattern recognition, computer vision, image processing, etc. It has also made understanding, manipulation, and working on large data-sets easy.Earlier when machine learning was introduced it was not able to gain any popularity as it requires large computation skills to solve problems but machines were not capable of doing it. With the improvement in technology over the years, we are capable of implementing these methods but it still demands large amounts of time and effort. The Open Source model helps us to overcome these shortcomings.
The machine learning algorithms built by masters and practitioners of this field can be used again and build something on top of it to save time and effort which has been wasted on creating something from scratch. It enables the community to use, modify and redistribute the code based on their requirements. This model has a positive impact on the developers' community as it paced up the development by reducing time and efforts in doing the work which has been already done. There are many classes of open source software available from different sources out of which we are interested in machine learning and deep learning-related projects from GitHub. We will be extracting required information from them which will help us in making decisions on questions discussed later in this study.

3 Experiments And Data-set Information

Q1.What are the types of projects available in our region of interest?

Many a time when a developer starts working on a project he/she might get confused by several repositories under one domain. For example, when we give 'Machine Learning' as the query to search, we get many repositories as result but which one becomes a messy task for the user as sometimes the name of the project is unable to give much information about it and going through every repository to find the right one is also not very convenient. So we thought of getting some more information along with the repository name to save time and effort. In this context, we decided to get categories/topics which are related to it.

Q2.What are the popular machine learning projects among users/software engineers?

As the main focus of this study is to make it easier for a user/software developer to choose the best-suited repository for his/her project. After analyzing the information at hand we came across many parameters such as

- stars, which can be interpreted as how many people liked this repository.
- forks, which gives us the number of users who found this repository useful and working with it.
- Watchers, number of people who visited it.

Now we wanted to use them to give some weightage to each repository, we call it Popularity Index. In the later section we explain how we come to a single number from these there different parameters.

Q3.What can be said about their Issue Resolution Rate?

There are some points that a person should keep in mind before he/she begins working on any project. One of them is about the responsiveness of a repository. Sometimes while working on a project developer can come across some problem with the repository content, it could be improper documentation, some piece of code is not functional, etc. These kinds of problems are reported to authors of the repository and are commonly called issues. These issues are resolved by authors and other contributors of the source. At any given time, there will be some closed issues(which are resolved) and open issues. We tried to classify them by a ratio, we call it Issue Resolution Rate(ISR). This will give us information about how faster issues are being resolved by the user/developers of this repository.

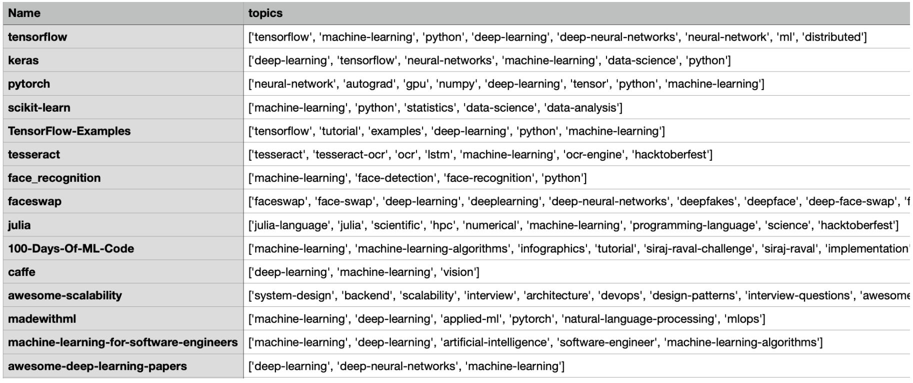
Q4.How frequently they are getting updated?
There is a common problem which many developers face while working with open source projects especially in the case of software that keeps on updating or APIs which get depreciated and no information is provided to users by authors/owners of the repository. This makes it very difficult for a user to work as result, he/she has to move from this project to another and hence configuring up the whole environment again. Active participation of authors also plays an important role in making any project successful as only they can commit any kind of changes to the repository and update it timely.
It is easy to understand it quantitatively so we tried to give a number to every repository showing its monthly frequency. After getting this we can tell that the repository we are going to choose is under the proper observation of its author or not.

Data-set: A very popular platform for open source projects, GitHub is taken as the data source. One of the major reasons for choosing it as a source of data is because of its ease of usability and its popularity also plays an important role to rely on it. Nearly 750 repositories are taken as data points to conduct this study from GitHub. This data is scraped using GitHub API and scripts were written in python language.

4 Methodology and Results

The methods used to answer the research questions are very simple yet effective to get the desired results. Different method is used to find a solution for every problem. We rely on mathematical analysis of scraped data to obtain the desired results.
Q1.What are the types of projects available in our region of interest?
In the interest of finding the type of projects, we choose readme files to begin with as they contain information about a project and it is written by the owners/authors of the project. Mostly textual information is provided in readme

files but not always. So we tried to apply some topic modeling techniques such as LDA to get the topics related to each repository. Results obtained were nowhere close to what we expected, in some projects we observed the absence of readme files or not in textual format. But while manually observing the data we came across one important feature which is commonly called tags. These tags are generally the topics given to a repository by its authors. This information easily gives information about the type of repository.
Most of the deep learning repositories are based on python language and deal with neural network-based frameworks. They provide code and implementation details so that one can use and modify it according to their requirement and further re-distribution can be done easily.Machine learning-related repositories are those which provide information about the libraries and their usage. Some repositories also aim to teach the basics and application of machine learning. This openness and sharing of thoughts are acting as a helping hand in pacing up the development process.



Name	topics
tensorflow	['tensorflow', 'machine-learning', 'python', 'deep-learning', 'deep-neural-networks', 'neural-network', 'ml', 'distributed']
keras	['deep-learning', 'tensorflow', 'neural-networks', 'machine-learning', 'data-science', 'python']
pytorch	['neural-network', 'autograd', 'gpu', 'numpy', 'deep-learning', 'tensor', 'python', 'machine-learning']
scikit-learn	['machine-learning', 'python', 'statistics', 'data-science', 'data-analysis']
TensorFlow-Examples	['tensorflow', 'tutorial', 'examples', 'deep-learning', 'python', 'machine-learning']
tesseract	['tesseract', 'tesseract-ocr', 'ocr', 'lstm', 'machine-learning', 'ocr-engine', 'hacktoberfest']
face_recognition	['machine-learning', 'face-detection', 'face-recognition', 'python']
faceswap	['faceswap', 'face-swap', 'deep-learning', 'deeplearning', 'deep-neural-networks', 'deepfakes', 'deepface', 'deep-face-swap', 'f
julia	['julia-language', 'julia', 'scientific', 'hpc', 'numerical', 'machine-learning', 'programming-language', 'science', 'hacktoberfest']
100-Days-Of-ML-Code	['machine-learning', 'machine-learning-algorithms', 'infographics', 'tutorial', 'siraj-raval-challenge', 'siraj-raval', 'implementation'
caffe	['deep-learning', 'machine-learning', 'vision']
awesome-scalability	['system-design', 'backend', 'scalability', 'interview', 'architecture', 'devops', 'design-patterns', 'interview-questions', 'awesome
madewithml	['machine-learning', 'deep-learning', 'applied-ml', 'pytorch', 'natural-language-processing', 'mlops']
machine-learning-for-software-engineers	['machine-learning', 'deep-learning', 'artificial-intelligence', 'software-engineer', 'machine-learning-algorithms']
awesome-deep-learning-papers	['deep-learning', 'deep-neural-networks', 'machine-learning']

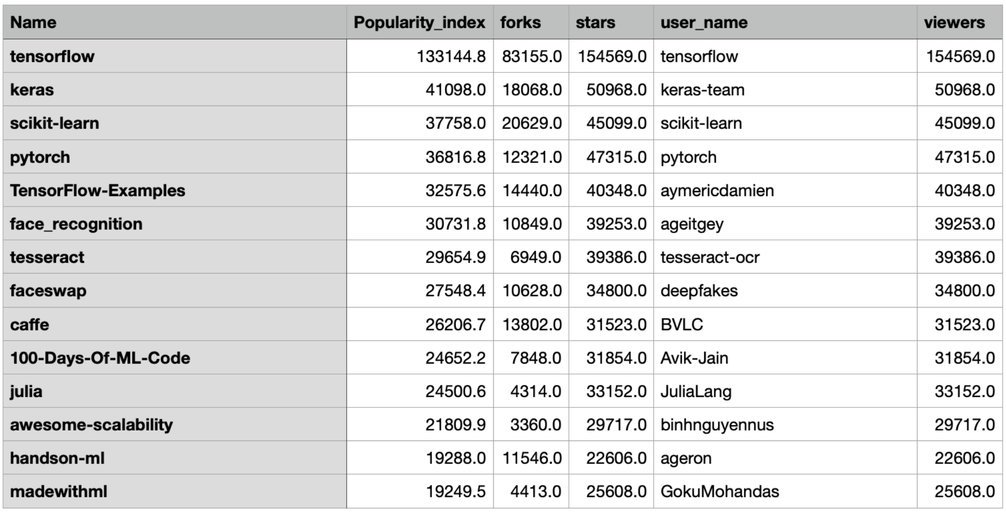
Fig1 shows some repositories with their repository tags as topics.

Q2.What are the popular machine learning projects among users/software engineers?

When we first searched for keyword-machine learning on GitHub we got thousands of projects which made us ponder which one to choose? This is the problem faced by many researchers and software engineers when they start working on any machine learning or deep learning project. From here we get the motivation to give some value by which it is easy to judge a repository.
We name it Popularity Index, in which we different weights to stars, forks, and views.
The formula used to calculate PI(Popularity index) is :

$$PI = 0.4(number_of_stars)+0.3(number_of_forks)+0.3(number_of_views)$$

This observation helped us to know which is more repository is having more footfall and likeness among the developers/users.



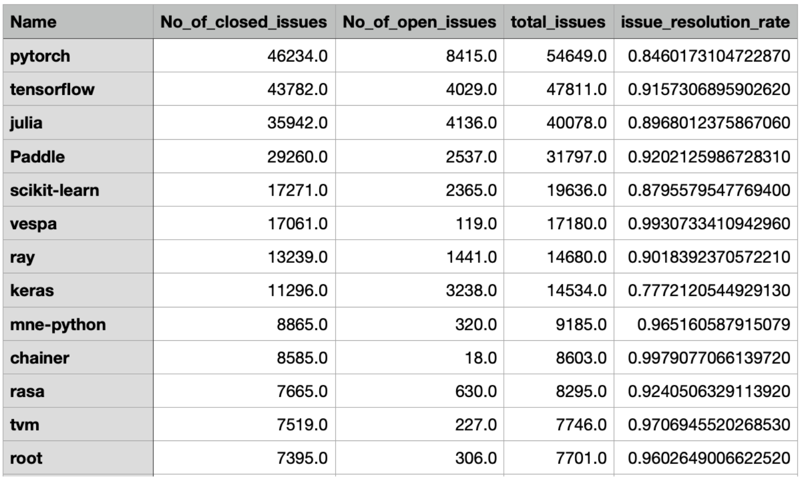
Name	Popularity_index	forks	stars	user_name	viewers
tensorflow	133144.8	83155.0	154569.0	tensorflow	154569.0
keras	41098.0	18068.0	50968.0	keras-team	50968.0
scikit-learn	37758.0	20629.0	45099.0	scikit-learn	45099.0
pytorch	36816.8	12321.0	47315.0	pytorch	47315.0
TensorFlow-Examples	32575.6	14440.0	40348.0	aymericdamien	40348.0
face_recognition	30731.8	10849.0	39253.0	ageitgey	39253.0
tesseract	29654.9	6949.0	39386.0	tesseract-ocr	39386.0
faceswap	27548.4	10628.0	34800.0	deepfakes	34800.0
caffe	26206.7	13802.0	31523.0	BVLC	31523.0
100-Days-Of-ML-Code	24652.2	7848.0	31854.0	Avik-Jain	31854.0
julia	24500.6	4314.0	33152.0	JuliaLang	33152.0
awesome-scalability	21809.9	3360.0	29717.0	binhnguyennus	29717.0
handson-ml	19288.0	11546.0	22606.0	ageron	22606.0
madewithml	19249.5	4413.0	25608.0	GokuMohandas	25608.0

In fig2 the top 10 repositories sorted according to their popularity index are shown.

Q3.What can be said about their issue resolution rate? After getting the popularity index, the next question which comes to mind is how frequently our issues will be resolved?
Now let us first explain about issues, these are generally the problems faced by the user to implement any particular feature of the project, sometimes there is some bug with some part of code or documentation that is not clear which makes it difficult for developers to work with that project. So he/she can post an issue that can be solved by authors or any other developer.This is the beauty of open-source that the issue posted will be seen by everyone and anyone can contribute resulting in speeding up the development.
So we came up with Issue Resolution Rate(ISR). We gathered information related to issues i.e. number of open issues and closed issues and then we need to normalize the results so we use this formula-

$ISR = numberofclosedissues/totalnumberofissues$

This also shows that how actively people are contributing to a particular project.



Name	No_of_closed_issues	No_of_open_issues	total_issues	issue_resolution_rate
pytorch	46234.0	8415.0	54649.0	0.8460173104722870
tensorflow	43782.0	4029.0	47811.0	0.9157306895902620
julia	35942.0	4136.0	40078.0	0.8968012375867060
Paddle	29260.0	2537.0	31797.0	0.9202125986728310
scikit-learn	17271.0	2365.0	19636.0	0.8795579547769400
vespa	17061.0	119.0	17180.0	0.9930733410942960
ray	13239.0	1441.0	14680.0	0.9018392370572210
keras	11296.0	3238.0	14534.0	0.7772120544929130
mne-python	8865.0	320.0	9185.0	0.965160587915079
chainer	8585.0	18.0	8603.0	0.9979077066139720
rasa	7665.0	630.0	8295.0	0.9240506329113920
tvm	7519.0	227.0	7746.0	0.9706945520268530
root	7395.0	306.0	7701.0	0.9602649006622520

In the fig3, the top 10 repositories with a high-resolution rate are shown.

Q4.How frequently repositories are getting updated?

We also wanted to know how active the authors/developers of a particular project are, which can be scraped by checking the commit history. This commit history is fetched with the help of a URL. As commit can only be done by repository owners/authors so it is valid proof of their participation. For that, we scraped monthly frequency for each repository from their creation date. This feature was not explored earlier by any of the study and it can act as reminder for owners to update their repository if its frequency is going low.

5 Related Work

Many researchers across the world have published about the benefits of using open source models[5] such as

- Integrated Management
- High-Quality Software
- Simple license Management
- and many more.

and machine learning is also being studied at every corner of the world but very little is being published about them together. So this study acts as a bridge between them.As we are moving forward in time we have developed machines with high computation capabilities and many other resources which have made analysing data possible in less time. Open-source software have created a positive impact in several fields so its need is well explained in [1] so we tried to explain how an amateur developer can step in this world by making the right decisons. In [2] the author only on specific packages of deep learning and gave a detailed quantitative analysis. Like what are the current practices which are being explored but it is limited to the reusability of such packages but we focus on why one should choose a particular project and what are the perks of choosing it. Some studies focus on languages used for building these machine learning frameworks and resulted in 'C++ being most commonly used to build it and 'Python' to implement it. [4]shows many developers are not using open-source as contibutors or as user of any project and they all face issues but what is they keep on waiting for an issue to be resolved. There can be many reason behind this problem absence or in-active behaviour of authors/owners could be one so this study aimed to throw some light on the accountability of projects.

6 Conclusion

Through this study, we provided information that can be useful to choose a machine learning repository according to one's need. And to support this decision we have some important factors such as the Popularity index which can be used by GitHub developers to increase the credibility of a particular repository, Issue resolution rate which can also help repository owners to keep track of their repository and improve it in the future if low issue resolution rate is encountered otherwise can still work for up-gradation and monthly frequency which can act as a reminder for repository owner to update it timely. Furthermore, this study can be extended in the future for all classes to give more insights and on several other sources which provide open-source software to remove the ambiguity faced by users/developers.

7 References

1. The Need for Open Source Software in Machine Learning-https://www.jmlr.org/papers/volume8/sonnenburg07a/sonnenburg07a.pdf

2. Empirical Study on the Software Engineering Practices in Open Source- ML Package Repositories-https://arxiv.org/pdf/2012.01403.pdf

3. Open Source software in medical informatics—why, how and what ?

4. How do programmers ask and answer questions on the web?
5. An empirical study of open-source and closed-source software products.
6. Open Source Frameworks for Deep Learning and Machine Learning