

An Open Source Platform for the Complete Life Cycle of AI and ML

Haleema Sadiya , Deepthi S , Harshitha Gowda KL ,Pranav T P, Darshan M R , Girish L
*Department of Computer Science & Engineering, Visveswaraya Technological University,
Channabasaveshwara Institute Of Technology, Gubbi, Tumakuru,
Karnataka, India*

haleemasadiya1211@gmail.com , pranavpante47@gmail.com.

Abstract- Machine learning is used in enterprise systems for automation, but it can be difficult to integrate many machine learning dependencies and utilities, due to the absence of a common architecture. In certain cases, machine learning frameworks are advanced for systems that are struggling for particular business area usage cases, resulting in redundant effort and incredible reuse. Acumos is a free and open-source framework. Acumos can conveniently bundle machine learning models into scalable containerized microservices that can be shared through the platform's catalogue and implemented into a range of business applications. Via Acumos app, providing a case study of packaging sentiment analysis and classification machine learning models, enabling allocation with others. We show that when adapting machine learning models to business applications, the Acumos framework reduces the technological burden on application developers.

I. INTRODUCTION

Acumos may be a stage that accelerates AI model creation, preparation, and implementation. Its aim is to extend the reach of AI-based software across a wide variety of industrial and commercial problems in order to accomplish a large number of applications. Acumos can drive close a data-centric mechanism in this approach for generating applications based on machine learning by way of the core paradigm. This stage is looking for qualified data scientists to build more adaptive AI models and relieve them of the burden of custom production of fully integrated solutions. Acumos is used by software engineers to transform the software creation process from a code-writing and editing practice to a classroom-like code preparation process in which templates are created and classified based on their abilities to effectively interpret datasets. The best model for the job will then be chosen and combined into a full application [3].

Acumos isn't knotted to any precise modeling language or toolkit and it is not partial to anyone run-time infrastructure or cloud service. Acumos generates an open-source mechanism for packaging, sharing, licensing and deploying AI models within the variety of portable, containerized microservices and publishes them in an exceedingly shared, secure catalog. Data scientists can use Acumos to create abstract AI models using their preferred or utmost suitable software, which can then be adapted to a variety of data formats using data adaptation libraries and turned into applications using a streamlined

chaining method. These models are intended for use by IT practitioners who do not have a data science experience or training in the various AI toolkits used by data scientists and who can incorporate the models into realistic applications. Acumos is designed to support a wide variety of tools and technologies for the development of machine learning models, including both open source and exclusive toolkits. Models can be conveniently onboarded and enfolded as containerized microservices that can interact with a wide variety of components. Acumos provides a Marketplace for data-driven decision making and artificial intelligence tech models that is independent of any toolkit. It allows users to securely exchange AI microservices as well as information on how they function, such as ratings, popularity figures, and user-provided feedback, in order to introduce crowd sourcing to software growth. The interface allows concept developers and apps to integrate in order to simplify user reviews, exception management, and product upgrades.

Acumos Design Studio is often used to connect many models, along with data translation software, filters, and output adapters, to create a complete end-to-end system that can be implemented in any run-time environment. The Acumos catalogue will be able to look for details on the licensing and execution conditions of both recycled AI products and fully optimized solutions, making product collection a breeze.

Acumos' Data Broker allows you to collect data from external sources, use it to mentor or tune models, and save it so that future models can be retrained [3].

Since the Acumos platform's source code is available under an OSI-approved open-source license, any feature can be readily tailored to new software toolkits, private data sources and data streams, and a personalized run-time environment.

Remaining section of this paper is well organized as follows: section two defines the problem statement; section three defines the goal of Acumos; section four explains the purpose and mission of Acumos platform; section five explains the literature survey of Acumos AI; section six describes the Acumos platform architecture; section seven describes the Acumos market place and four stages of AI development supported by Acumos; section eight explains Acumos design and its process; section nine explains Acumos applications; section ten explains characteristics of microservices; section eleven explains AI models; section twelve concludes the paper;

II. PROBLEM STATEMENT

Previously we've such a large amount of popular AI frameworks that are there to create AI apps. But inappropriately, the combination of AI apps developed using these frameworks isn't a straightforward task for beginners, as it's totally attached the cloud-based environment. Only the progressive programmers will do this. So ACUMOS AI consists of Design Studio supported Linux which helps you to integrate these frameworks with one another and it provides comfort form of deployment for beginners.[13]

ACUMOS' key goal is to make AI and machine learning available to a wide range of people by providing a marketplace of reusable solutions and references from a variety of toolkits and languages [1]. It makes an ordinary developer who is not a skilled in this field can easily able to create their models and applications.

III. PURPOSE AND MISSION OF ACUMOS PLATFORM

Acumos is using an open-source partnership platform to achieve openness and provide an industry-wide ecosystem for developing, preparing, implementing, and delivering machine learning solutions. Acumos splits the production flow into four separate phases in order to simplify the process and make it accessible to a wider community of developers. Models are first onboarded to an Acumos-compliant platform and bundled as independent microservices, each with a component blueprint connecting to the microservice API and dependencies. Second, the model is bundled into a training program that can be installed in an appropriate training environment and used to practice and cache data for potential retraining. Third, a reference training model known as Predictor is released into a catalogue that can be circulated around a network so that other developers can discover, discuss, evaluate, and build a complete solution by combining the Predictor with other modules that use the Predictor to make decisions but also have all of the more traditional skills needed to work on them. [15] Finally, the whole solution is packaged into a Docker container that can be deployed to a suitable runtime environment for execution. Although some of the components in this bundled system are used to view data and incorporate features, others are used to move new data records to the Acumos platform, where they can be connected to cached datasets and used for additional training during the continuous learning phase [1].

IV. LITERATURE SURVEY

The field of artificial intelligence (AI) is described as the study of "intelligent agents" in computer science. Typically, the term "AI" refers to a computer that simulates tasks that humans do in conjunction with other human brains, such as learning and problem solving [5].

There has been a massive explosion of tech that combines artificial intelligence elements in the last few years. Machine learning, natural language processing, image processing, and

data mining are only a few of the AI subfields that have caught the attention of today's engineering giants. Machine Learning

is heavily used in Google's predictive search bar, Gmail's spam blocker, and Netflix's display recommendations. Apple's Siri and Google's voice all use Natural Language Processing. Face recognition tagging apps on Facebook and Google's self-driving vehicles all require image processing. Because of the vast volumes of data gathered every day, data mining has become a slang word for the computing industry. Companies like Facebook and Google gather massive quantities of data from users every second and are looking for a way to view it. In today's technology-driven culture, Artificial Intelligence has already proved to be a valuable new weapon.

The "top down" approach, which began with the upper-level functions and applied them, and the "bottom up" approach, which detected at the neuron level and agitated to establish higher-level functions, are the two major methods that have been created for the standard AI system [2].

In the last decade, artificial intelligence has come a long way. However, developing powerful AI also necessitates an excessive amount of effort. Giving a computer common sense or insight may be a crucial factor in getting it to think. While understanding how to translate input to output appears to be important, good AI needs a computer that understands why output is connected to input. It's also important to continue to improve techniques for identifying human feelings and behaviors. This is frequently a multi-disciplinary topic that can require advances in Psychology, Linguistics, Machine Learning, Natural Language Processing, and Image Processing to better understand how humans behave in order to identify emotions and investigate human gestures and body language [5].

Open-Source AI:

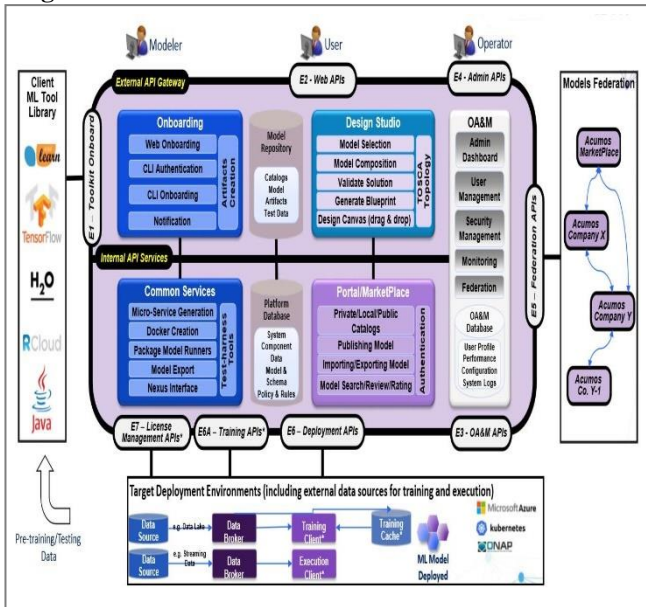
The AI sector is being largely dominated by open-source software that puts together many partners and organizations, much as other software applications. In Open-Source AI Ventures, Perspectives, and Patterns, Ibrahim Haddad writes [8], "It is increasingly common to work out AI as open-source projects. This is often partly thanks to its roots in academia, which has historically been a gentle source of open-source proof-of-concept projects. It's also deep-rooted in the cost to make a platform and also the realization that the precise value is within the models, training data, and also the apps. By open sourcing a platform, the host has any chance to employee others (possibly unpaid) contributors, mainly if other organizations is incentivized to integrate the platform into their private products. Like any technology where talent premiums are high, the network effects of open source are very strong."

V. ACUMOSPLATFORM ARCHITECTURE

Below figure shows the architecture of the Acumos platform. It contains a function for onboarding models from ML tools (TensorFlow, etc.), a set of mutual services for microservice and Docker creation (developers can create and export production-ready AI applications as Docker files), the marketplace function (discussed below), and supportive operations and admin capability.[1] The design studio offers a graphical interface for combining several templates, data translation methods, filters, and output adapters into a complete end-to-end system that can be implemented in any runtime

environment to speed up AI applications.

Figure 1: Acumos Platform Architecture



Source: Acumos

VI. ACUMOS MARKET PLACE

Acumos leveraged the community development concept not only to develop its private code base, but likewise to create a marketplace of models. Copying the "app store" concept, Acumos provides a marketplace where academics and commercial data scientists can vend specialized ML models, customized to address specific industrial use cases. CSPs (and other companies) can experiment with the crowdsourced models on their personal datasets and, if they prove useful, pay to use them in production. By diminishing the customization and contextualization needed to use off-the-shelf pre-trained models, users can get insights more quickly and economically. Some models are available in the marketplace (contributed by different participating organizations) that are generic enough to be downloaded and re-trained for a precise context. These models coverup areas such as image classification, face detection, face privacy filtering, image mood classification, topic modeling, cross-selling and customer segmentation.

Acumos AI Development Process:

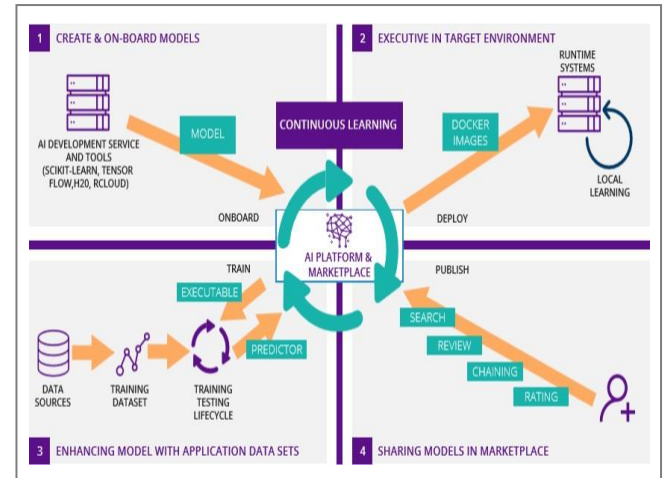
- 1 Models are typically imported from a data science toolkit (TensorFlow, for example) and bundled as microservices with a component blueprint detailing the API and dependencies[1].
2. The model is then bundled into a training programme that can be deployed to a training area without needing any modifications from a developer. Custom training customers, data access, and data caching capabilities make putting together a targeted training framework for each machine learning model simple [7].
3. Then, to transform a simple model into a Predictor that has been learned to perform a particular function, Acumos provides the training and testing interface. The Predictor is added to a list

and made available to the Acumos community. Through using the Predictor and chaining it with other components using the

Design Studio, other developers will evaluate it and construct a complete solution [1].

4. Lastly, the entire solution is compiled into a Docker image file that can be used in a Docker environment with the help of a container management tool like Kubernetes [7]. Image files can be distributed to Azure, AWS, and other common cloud providers, as well as any enterprise data centre or any real-time environment that supports Docker or other possible lifecycle management software [4]

Figure 2: Four Stages of AI Development Supported by Acumos



VII. ACUMOS DESIGN AND PROCESS

Acumos' overall architectural style is extensive and includes many elements. Acumos' mission is to build an inclusive environment of three stakeholders. Figure 3 represents the key contact flow between these stakeholders. A product contributor adds models to the site by uploading (on-boarding) them. The templates are transferred to the end user, who then uses them in their own applications. The Acumos Platform supports the platform and enforces sharing concept requirements. It should be remembered that a contributor can still be an end user; the network can be used to store and distribute templates and microservices for personal use as well as for sharing with others [4].

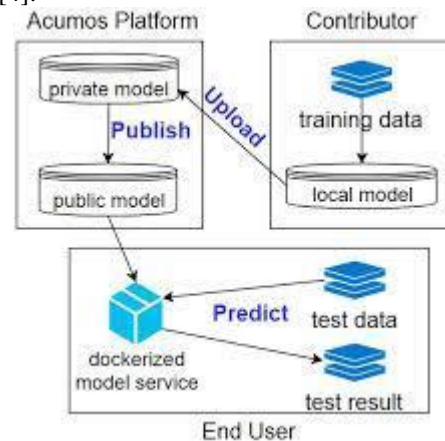


Fig. 3. Stakeholders and their relationship in the Acumos ecosystem

The main flow is divided into three stages:

- **Uploading:** This process starts when a contributor (e.g., a modeler or data scientist) uploads a locally trained model to Acumos. A contributor will create and practice his or her ML model using a variety of existing ML libraries. Contributors can upload their templates using Acumos' Python, R, and Java APIs. Since data scientists are always unwilling to adjust their preferred ML environment, the option to use their preferred vocabulary and ML toolkit is advantageous. As a result, a contributor will build templates in his or her chosen setting [4].

- **Publishing:** The templates are saved in a private location where only the contributor has access since they've been posted. The contributor will then choose whether and how he or she wishes to share his or her model. Meanwhile, since the concept is encapsulated and functions as a 'black box,' the contributor is expected to define the model's documentation, such as the feature definition, input and output formats, and model category, before publishing it. Other users will use the metadata to browse the templates and find the ones that are right for them [4].

- **Predicting:** The Acumos framework will bundle the uploaded model as a microservice in a Docker image that is ready to be deployed and used (prediction, classification, etc.). As opposed to virtual machines, Docker6 offers container virtualization and has faster capacity, agility, and portability. The main explanation is that hardware is being virtualized to run multiple Operating Systems (OS) instances of virtual machines, while Docker allows you to run multiple workloads on a single instance. Customers will download the Dockerized service and deploy it to the cloud or some other local hardware that supports Docker. Users can send feedback to this operating microservice and receive its output through a RESTful API as soon as it is deployed [4].

VIII. APPLICATIONS

1. Gaming: - The most well-known use of AI is Video Game AI, which has been used for quite some time, in fact, since the very first video games. However, the explosive increase in the sophistication and efficiency of AI in recent decades has led to video game characters learning our attitudes, adapting to prompts, and reacting in unpredictable ways. In 2014, the game 'Middle Earth: Shadow of Mordor' was published, and it is widely regarded as one of the best examples of NPC (Non-Player Character) identities, past interaction experiences, and variable goals. Some shooting games, such as 'Far Cry' and 'Call of Duty,' use AI as well, with enemies that have the unique ability to analyze their surroundings in order to find treasure or execute acts that can aid in their survival; to improve their chances of victory. In terms of AI, video games are relatively easy, but due to the high market demand, a significant amount of time and money is spent each year to perfect this AI [2].

2. Natural Language Processing: - Conversation and human expression are both complex and subjective. Mouse and keyboards, as well as a specific and simple series of verbal instructions, are the existing traditional means of computer communication. This varies from how humans communicate

due to the degree of variability of human communication; the term "blood" in "red hair" is not the same as "red" in "red apple." This basic issue of correctly expressing ideas with objects, or sentences, is impeding Natural Language Processing's development. If these barriers are solved, natural language processing systems may be able to prompt learned biases, translate languages at human interpreter speeds, and understand the distinction between a red apple and red hair, and put orders like "hand me the purple thing down there" into effect [2].

3. Image Processing and Vision System: - Human-centered architecture aims to step away from the current model, in which a computer merely obeys commands provided through keyboard, mouse, or simple verbal commands. This change would necessitate a greater capacity to process images and view data. The bulk of current facial recognition software employs 2D spatial processing, which searches for abstract forms and edges in the face [2]. However, evidence suggests that studying both facial expression and body language yields the most precise behavioral judgments of human activity.

4. Self – Driving Cars: - No one has ever seen someone doing their nails, tying their belt, or doing something else while driving, but smart cars or self-driving cars are getting closer to reality; the two most recent approaches that have been seen in the news recently are Google's self-driving car initiative and Tesla's autopilot feature. Google has created an algorithm that could help self-driving vehicles learn to drive in the same way that humans do, but from training. The purpose of designing this algorithm was for the car to finally be able to look down the road and make choices based on what it sees, while still assisting it in learning. Although Tesla's "autopilot feature" is still in its early stages, it is already being used on the road, suggesting that these innovations are inevitably on their way in [2].

5. Fraud Detection: - You must have received a letter or a mail from time to time asking whether you've made a certain payment with your credit card or not? Many banks send these kinds of emails to see if there has been any violation on your account and to ensure that you have acknowledged the purchase before turning the money over to another entity. Artificial Intelligence is the infrastructure that is being used to keep an eye on this form of fraud. Computers deal with a vast variety of dishonest and non-deceptive transactions, so they've been asked to try to detect signs that a sale falls into one of two groups. After a sufficient amount of training, this type of AI would be able to detect a fraudulent transaction based on the signs and signals learned through the training process [2]. And many more application are there.

Acumos has a packaging feature that allows you to wrap individual modules, such as templates, adapters, and public data sources, into microservices, making it simple to create composite solutions that share data, even though each component is written in a different language or uses a different toolkit or SDK.

Since Acumos microservices are designed to adopt a shared communications model and operate in separate containers, they are often interoperable [1].

A microservice is a part that follows a particular architectural style for constructing business functions out of tiny, recyclable

units. (Figure 2).

A discrete framework can be made up of a loosely coupled series of such microservices, each of which has a particular feature or task in the overall use case, is designed without any external dependencies on other microservices, and has a well-defined API and an integrated and resilient lifecycle.[16]

As a result, each microservice can be restarted and replaced independently of other microservices or some other hybrid solution created from these microservices in the event of a malfunction. The aim is to condense a single function and enable modularity without complicating the programme it serves. Acumos' microservices are made up of Docker containers, making them easy to deploy and maintain across a multitude of target environments. Microservices that are highly complex can't be easily incorporated or implemented.

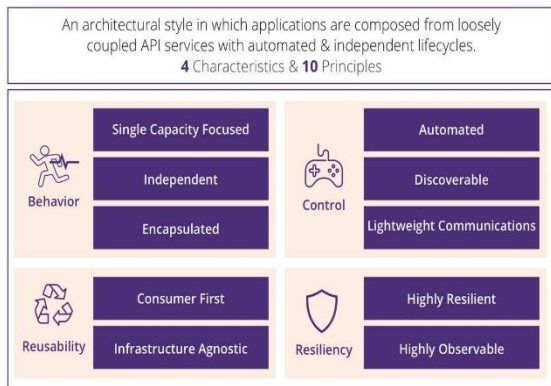


Figure 2. The characteristics of microservice

IX. ACUMOS DESIGN STUDIO AND INTEGRATION OF AI MODELS

Machine learning as a driver of AI is a very useful tool for solving certain problems, but it does not fully substitute traditional programming. Machine learning, on the other hand, is a method for coping with a particular type of problem. The general classification problem is one kind of problem where machine learning works well.

In fact, most photographs do not focus on a single subject. We might list all of the faces in a picture if a classifier would learn to detect them. So far, only a small number of issues have been addressed. For e.g., group all of Uncle Fred's photos in my collection – that could be all we need to know. However, the challenge is much easier in most practical usage cases [2].

ATT- FACE PRIVACY FILTER GUIDE

This model comprises the capability to generate two sub models: one for face detection and one for face suppression through pixilation. A model example for face detection from images within Acumos [20].

Background: This model examines static images to detect frontal faces. It uses a frontal face cascade from

the [OpenCV](#) image processing library. Model load time is enhanced by creating and maintaining the fixed cascade in memory while operating. Demonstrating the capability of custom classes and necessary member variables, the cascade is serialized with the model as a string asset which is deserialized

and loaded from disk upon startup.



Usage: Input to the model is an array of at least one tuples of image binary data and a binary mime type. The position of the image within the array is used in the output signature as a zero-based index. For example, if 3 images were sent, the output probabilities would have 0, 1, and a pair of as index values. The output of this model could be repeated array of the detected regions for every face in each input image. In order that image data may be cascaded to any other models, the original image and mime type are embedded with the special region code - 1 within the output.

Performance: By way of this model wraps a previous cascade, no formal testing evaluation was achieved. Yet, experimental usage indicates the subsequent highlights. [20]

- Faces that are also minor can easily be missed.
- Frontal faces accomplish best, with some tolerance of about 5-10 degrees off-plane rotation.
- Detection is equally sensitive to rotation in plane, so try to not let subject faces rotate over 15 degrees.
- Dark or low contrast images usually don't perform well for detection.

TOPIC MODELLING MODEL

An unproven approach to find the topics in the documents. It attains frequent patterns of words or topics in textual material.

Benefits: Automate the process of extracting topics/trends from a text corpus. Topic modeling can deliver a useful view of a vast collection/Corpus in terms of the collection as a whole, the individual documents, and the relationships between the documents.

Input: The sample documents that are used to train the model is about some health and documents speaking about the family. E.g.: doc1 = "Sugar is bad to consume. My sister adores to have but not my father"

Output: The Output will be the topics with topic weight of each term

['0.168*health+0.083*sugar+0.072*bad'] interprets the document is talking about health.

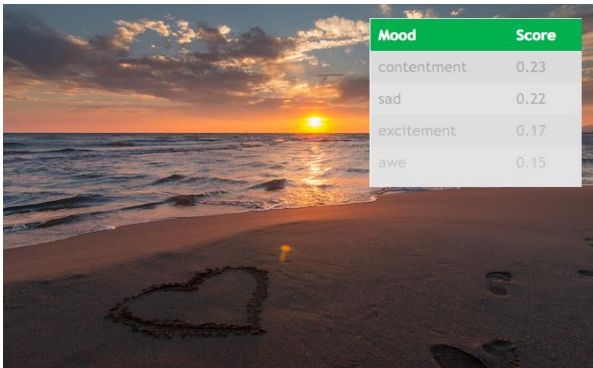
ATT- IMAGE MOOD CLASSIFIER

A model example for the compound image classification for emotional impression and metadata tagging.

Background: This model trusts on a first pass with image classification features and then studies a secondary classification layer. To use this model within Acumos, one would need to use a model connector accessible through the design studio. Using classification outputs from image classifier, guess the mood of an image into classes (here Amusement, Awe, Contentment, Excitement, Anger, Disgust, Fear, and Sad).

Usage: Input to the model is an array of several classification scores for each image. Definitely, an image index, a class name, and the probability of that class are expected as inputs. Outputs of the model may contain a similar format excluding that they will be created in the domain of the mood classifier classes: image index, mood class name, and probability score. The probabilities are normalized to sum to 1.0 over all values so that they can be used as relative confidence scores.

Performance: The model has very few training modes, but there are many algorithms (random forest) that seem like to perform the best between nonneural-net computations. The training and evaluation are finished only though [scikit-learn](#) to demonstrate the utility of simple pipeline wrapping for an Acumos model.[20]

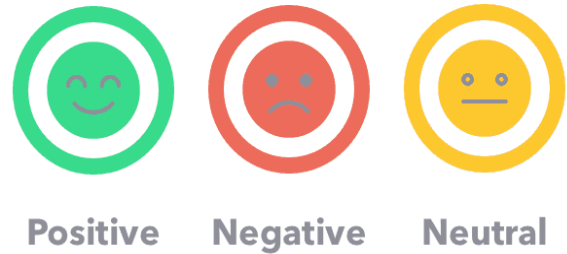


SENTIMENT FOR ONLINE USER GENERATE CONTENT

In the web 2.0, a huge part of online data is generated by users, such as movie reviews, Amazon product reviews, hotel reviews. According to a recent survey, 93% of users feel User-Generated Content (UGC) are supportive when making a purchasing decision. Likewise, the sellers regard the sentiment over UGC as a good indicator for the product or service feedback. Therefore, more and more companies built their own sentiment analysis model to monitor their brand reputation. Though, the models were built with different qualities. Some are good, and some are bad. and there is a lot of repeated work [18][19].

The problem is how to develop some good models and share with others. We constructed a sentiment analysis model for user generated text comments. It ensembles three deep learning models (R-CNN, residual LSTM and Attention LSTM). The model is pre-trained over 5 million IMDB movie reviews. [17]

Sentiment Analysis



X. CONCLUSION

Acumos main objective is to make AI more accessible for real-world business applications. Data scientist can underwrite ML models that ordinary developers can easily use to create their personal applications. By making the platform open source, Acumos will knock into the development capabilities of many participants.

Acumos enables data scientists to print AI models without requiring them to create custom, fully developed solutions that use those models. Acumos is independent of some particular runtime infrastructure or cloud provider. It promotes a number of hardware infrastructures to ensure that the technologies implemented are as useful as possible. Acumos compatible systems are both compact and adaptable.

Acumos offers a framework for packaging, exchanging, licensing, and deploying AI models in the form of interoperable, scalable, containerized microservices. It has a publishing framework for sharing and securing catalogues, as well as a deployment mechanism for any appropriate runtime infrastructure. Since it ignores the training process, the traditional software development process – code, validate, debug, retest, deploy – is insufficient for developing ML systems. For machine learning, teaching the programme model is an essential part of the production process. Until code can be completely debugged, it must be written and qualified, making the overall production process more difficult.

This shifts the focus of software development to a new generation of training-oriented technologies that deal with data collection and storage. Data scientists will continue to programme the models, but the majority of the analysis will be accomplished by training the models with large volumes of data. Rather than the basic ML algorithm used, the training method and related data could be the deciding factor in the overall effectiveness of the solution. Acumos would allow developers to test various algorithms on model datasets before making a final decision.

REFERENCES

- [1] AT&T and Linux Foundation, "Acumos ai: An open source ai machine learning platform." [Online]. Available: <https://www.acumos.org/>
- [2] Available: www.ijert.org
- [3] Available: docs.acumos.org
- [4] C. Wang, S. Zhao, A. Kalra, C. Borcea, and Y. Chen, "Webpage depth viewability prediction using deep sequential neural networks," IEEE Transactions on Knowledge and Data Engineering, 2018.
- [5] Submitted to university of Nothumbria at Newcastle
- [6] Available: www.scilit.net
- [7] Gangadhar. "Crop Yield and Rainfall Prediction in Tumakuru District using Machine Learning." (2019).
- [8] Techrights.org X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in Proceedings of the 28th international conference on machine learning (ICML-11), 2011, pp. 513–520
- [9] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," arXiv preprint arXiv:1502.02791, 2015.
- [10] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2010.
- [11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit-learn: Machine learning in python," Journal of machine learning research, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [12] Girish, L., & Deepthi, T. K. (2018). "Efficient Monitoring Of Time Series Data Using Dynamic Alerting". i-manager's Journal on Computer Science, 6(2), 1-6. <https://doi.org/10.26634/jcom.6.2.14870>
- [13] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen et al., "Mllib: Machine learning in apache spark," The Journal of Machine Learning Research, vol. 17, no. 1, pp. 1235–1241, 2016.
- [14] M. Hofmann and R. Klinkenberg, RapidMiner: Data mining use cases and business analytics applications. CRC Press, 2013.
- [15] C. Olston, N. Fiedel, K. Gorovoy, J. Harmsen, L. Lao, F. Li, V. Rajashekhar, S. Ramesh, and J. Soyke, "Tensorflow-serving: Flexible, highperformance ml serving," arXiv preprint arXiv:1712.06139, 2017.
- [16] D. Crankshaw, X. Wang, G. Zhou, M. J. Franklin, J. E. Gonzalez, and I. Stoica, "Clipper: A low-latency online prediction serving system." in NSDI, 2017, pp. 613–627.
- [17] L. G. (2019). Anomaly Detection in NFV Using Tree-Based Unsupervised Learning Method. International Journal of Science, Technology, Engineering and Management - A VTU Publication, 1(2),27-31. Retrieved from <http://ijesm.vtu.ac.in/index.php/IJESM/article/view/232>
- [18] J. N. Van Rijn, B. Bischl, L. Torgo, B. Gao, V. Umaashankar, S. Fischer, P. Winter, B. Wiswedel, M. R. Berthold, and J. Vanschoren, "Openml: A collaborative science platform," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2013, pp. 645–649.
- [19] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [20] Acumos-An open-source AI machine learning platform. Marketplace.