I.J. Intelligent Systems and Applications, **2022, 1, 1-23**
Published Online February 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2022.01.01

Towards an Intelligent Machine Learning-based Business Approach

Mohamed Nazih Omri
MARS Research Laboratory LR17ES05, University of Sousse, Tunisia
E-mail: mohamednazih.omri@eniso.u-sousse.tn

Wafa Mribah
MARS Research Laboratory LR17ES05, University of Sousse, Tunisia
E-mail: wafa.meribah@gmail.com

Received: 07 October 2021; Revised: 11 November 2021; Accepted: 02 December 2021; Published: 08 February 2022

Abstract: With the constant increase of data induced by stakeholders throughout a product life cycle, companies tend to rely on project management tools for guidance. Business intelligence approaches that are project-oriented will help the team communicate better, plan their next steps, have an overview of the current project state, and take concrete actions prior to the provided forecasts. The spread of agile working mindsets is making these tools even more useful. It sets a basic understanding of how the project should be running so that the implementation is easy to follow on and easy to use.

In this paper, we offer a model that makes project management accessible from different software development tools and different data sources. Our model provides project data analysis to improve aspects: (i) collaboration which includes team communication, team dashboard. It also optimizes document sharing, deadlines, and status updates. (ii) planning: allows the tasks described by the software to be used and made visible. It will also involve tracking task time to display any barriers to work that some members might be facing without reporting them. (iii) forecasting to predict future results from behavioral data, which will allow concrete measures to be taken. And (iv) Documentation to involve reports that summarize all relevant project information, such as time spent on tasks and charts that study the status of the project. The experimental study carried out on the various data collections on our model and on the main models that we have studied in the literature, as well as the analysis of the results, which we obtained, clearly show the limits of these studied models and confirms the performance of our model as well as efficiency in terms of precision, recall and robustness.

Index Terms: Machine learning, Business Project Management, Agile, BI tool, Data-driven, Predictive.

1. Introduction

1.1. Context and issue

In the information age, everyone produces data. In fact, tracking and extracting behavioral data is now possible and easier. The rise of 'big data' led to the renaissance of new fields such as business intelligence (BI). Business intelligence is the derivative of decision support systems. It helps businesses gain a competitive edge by supporting and improving their decisions with relevant, insightful information. Business Intelligence tools in general whether they are AI powered or not and whether they aim for project management or not, makes it easier and faster to discover, understand, and act on opportunities. It assists all team members from developers to Project managers with actionable insights. We have decided to work with BI tools in a business environment to assist the tech companies in project management tasks. This involves helping stakeholders with planning, scheduling, and predicting in the purpose of delivering the right product at the right time.

Project management has become more and more necessary to the business. On the one hand, companies want to gain control over the on-going project, thus the traditional approach of estimating future work, relying on one team member for project overview is not enough. Let alone missing deadlines and having to justify that to stakeholders and investors. On the other, they want to avoid the potential misunderstandings and confusions that could happen within a team. Especially since nowadays, teams are multidisciplinary, so they come with different point of views. That is why project management approaches are there to boost the communication and create an agile working environment. Adding to that, it is a great way to display what is going on with your project, what are the risks that you can encounter in the future in order to take the appropriate actions before unwanted incidents occur.

1.2. Contributions

We suggest a solution that makes project management accessible from different software development tools and data sources. We analyze the project data for the purpose of improving the following aspects:

- Collaboration: This part includes team communication, team dashboard where teams are provided with an overview about whether they are meeting the goal or not. It also optimizes sharing of documents, timelines, and status updates to notify everyone of important information such as per se how much work is done and how far are they in the project.
- Planning and scheduling: it can be difficult for your team to stay within schedule due to the lack of a specific guideline. Thus, we will aim to utilize the software outlines tasks and make them visible. This will also involve task time tracking to display work barriers that some members could be facing but not reporting.
- Forecasting: predicting future outcomes from behavioral data is very useful for taking concrete actions. This will support the team into making the right decisions for the benefit of the project. It can showcase Whether the project is in risk of shortage of resources or time or whether there is an opportunity to perform better after certain changes.
- Documentation: The approach involves reports that summarize every useful insight of the project from statistics and averaging different project factors like time spent on tasks per a period to charts that investigates the state of the project.

1.2. Paper structure

The current article is structured as follows: section 2 reviews the field of business intelligence while presenting its common methodologies in an agile environment. In this section, we will also discuss the company in general and its current Project management practice. Finally, we will present the published related work of Business intelligence tools. Section 3 presents de main related works in the literature. In section 4 we present the motivation of our solution. Section 5 aims to define the suggested solution on both architectural and logical terms. We will dive deeper into the server side of the approach and its deployment while introducing the techniques that helped to accomplish that. Section 6 focuses on the predictive modeling based on the learning machine. In this section we will discuss random estimations and the reason behind machine learning implementation in order to answer two predictive problems. We will define the different algorithms used to achieve the proposed solution. In section 7, we present the experimental study and the results analysis. In section 8 we conclude the article, and we give some future works.

2. Related Work

At this stage, we have gained the basics about business intelligence and project management. We have seen that these two fields can be combined into one purpose in order to manage projects within the company. Some companies have already started the implementation process of BI applications. We can mention few projects under the same scope, but we will content on the most adjacent among them.

In [1] the authors proposed a study of analysis of agile supply chain enablers for an Indian manufacturing organisation. This study deploys, analytic hierarchy process (AHP), a popular multi-criteria decision making (MCDM) tool as a solution methodology, such that the decision problem breaks into a hierarchy of different levels constituting goal, criteria and alternatives. The results show that there are three enablers namely virtual enterprises, customer satisfaction and adaptability are among the top priority enablers; enabler collaborative relationship is the moderate priority enabler and remaining three enablers i.e., use of information technology, market sensitivity and flexibility are the lowest priority enablers. Tran and al. in [2] proposed a CT-based agile 3D printing system for pre-operative planning of orthopedic surgeries. This work addressed the increase in complexity and number of orthopedic surgeries especially in the aging population is a major challenge to orthopedic surgeons. The availability of physical model of patient-specific bones can benefit pre-operative planning of surgeries and lead to better surgical options and improved outcomes. This work describes a CTbased agile 3D printing system for manufacturing patient specific bones that can be used for preoperative planning and assist in the choice and application of orthopedic implants. In the approach proposed by [3] the authors presented a model titled 'Glencoe – a tool and a methodology to manage variability within the product development process'. In this work, the author addressed typical problems in this area and how the presented methodology handles them. The free to use web application Glencoe implements main parts of this methodology. It is shown how Glencoe can be employed in industrial practice. In [4] the authors proposed a knowledge management support in the engineering change process in small and medium-sized companies. The main contribution of this work is the step KM model that is integrated into the EC process. Failure modes and effects analysis (FMEA) and design history files are the documents used to manage the knowledge related to a specific product. A product's design history file should contain explanations of decisions. Supporting activities for applying KM should include a campaign to raise awareness, and special attention should be put on the transfer of tacit knowledge that can be stimulated by mixed teams of senior and up and coming engineers. The content of the acquired knowledge should be checked periodically, and the

analysis should be followed by corrective measures. In another work [5] the authors proposed a multipath methodology to promote ergonomics, safety and efficiency in agile factories. The research aims at providing a pragmatic approach to support the application of ergonomic risk management in practice. It defines a multipath methodology to investigate human factors impacting on safety by considering the specific workspace, the adopted tools, the overall production environment and the workers' activity. An industrial case study is described to illustrate the methodology and demonstrate the benefits for companies. Results suggest that the proposed multipath methodology allow to effectively assist analysts in the definition of crucial risk factors and selection of proper ergonomics assessment and measurement tools according to the specific context of application. We cite among others the work in [6] where the authors propose an architectural model of the integrated system for the management of web-based e-commerce business processes. In [7] the authors propose a Context for Operations as a basis for the Construction of Ontologies to Employment Processes. The authors of [8] provides a business process management environment for collaborative research that can be supported by Web 2.0. In [9] the authors proposed a comparative analysis of factor analysis model for pinpointing agile developers. To find out most important agile developers, a survey-based study was conducted with the aim to develop model for agility developers and measure its level in Indian manufacturing industries. It was hypotheses that sensitiveness to change, relationship with customer, relationship with supplier, flexibility and competency and responsiveness is not significant with agile manufacturing. The work proposed in [10] consists of a powerful drag-and-drop tool that serves for custom reports, charts, and dashboard gadgets creation. It allows the user to access multiple features:

- Account creation for different team members where each view is aimed for a specific user.
- Data importation to initiate the application. The data is scraped from SQL, CSV files, spreadsheets.
- Data visualization and analysis. This step allows for data exploration, and it helps the user to create reports related to his project.
- Report publication.

These characteristics and functionalities are displayed through EazyBI dashboard below:

- Type of issue: Bug, Story, Task.
- Priority: major, minor or critical.
- Weights or points depending on the issue's complexity level.
- Dates of creation, start, update and closing. Sometimes resolution date is added.
- A short summary or title.
- Description.
- Comments about the issue.

3. Motivation

Artificial intelligence precisely Machine Learning [11, 12] is leveling up the game by giving us access to tools that are sufficiently learned and insightful to effectively take charge of our project. The benefits for the user are immensely tangible:

- Task planning and stakeholder coordination will be fully supported by the approach. At no time will the user need to recompile the elements himself to reorganize them in time, whatever the changes. This will not prevent him from remaining sole master on board and can modify at leisure the proposals that will be made to him. Profit: a considerable time saving.
- Optimization of the organization as the approach will be able to take into account at the same time all the elements (tasks, efforts, constraints...) specific to the user as to his collaborators. If the user chooses to follow these recommendations, he will be sure to always take the right action at the right time! Profit: a very significant gain in productivity thanks to the optimization of resources and securing deliverable in compliance with the commitments made.

For these reasons, in our proposed approach we will adopt the agile model and precisely the SCRUM method for software development, management and monitoring of our activities throughout the project life cycle. Daily scrum meetings are planned, tasks are created and distributed within Jira dashboard.

Our work will focus on the jira software, so it is important to be aware of the details related to Jira board and its features. The scrum board will illustrate the definition of the different issues and their states, assigned points, descriptions and other details that we will mention later in section 4 when speaking about the predictive modeling.

4. Intelligent Machine Learning-based Proposed Approach

In this section, we will discuss the solution in depth and unveil the different components that formed Geigr.io (see figure 1): the AI powered project management approach. We are also going to present an in-depth conceptual study of the approach including the functional requirements, end-users as actors and the user stories destined for these actors. Our approach is a business intelligence tool addressing agile project management while distinguishing itself from the market and handles the past, present and future outcomings of a project and adapting to new ones.

4.1. Functional requirements

Our proposed approach revamps three important factors including decision-making, planning and improving the agile workflow. It is the answer to the serious lack of meaningful support for software teams and practitioners. The figure 1 below gives away an overview of the approach. Each side of the figure represents a technical need that Geigr.io is trying to solve.

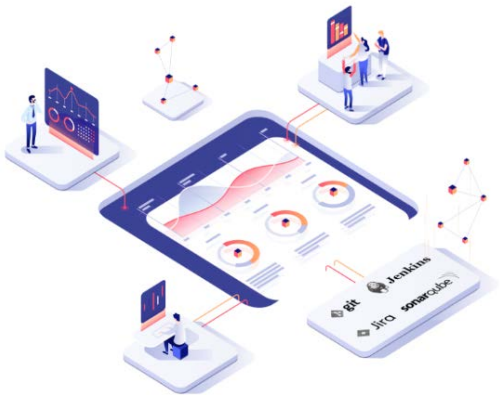

Fig.1. An overview of Geigr.io [13]

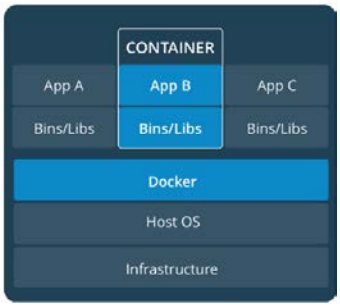

Fig.2. Docker Architecture [4]

The components visualized in the figure 2 above, represents four functionalities of Geigr:

- Untangling the project data: Geigr.io gathers Data from four main data sources. Data that usually goes misinterpreted or neglected. Its collection is done through Jira, Github, sonarqube, jenkins and gitlab-ci.
- Supporting decision-making process: Geigr.io offers predictive analytics that unveils risks and opportunities to boost performance thus it helps stakeholders take concrete steps.
- Scheduling and planning: Gegir.io presents an interactive dashboard that is accessible to multidisciplinary end-uses and help them to redeem control over their project workflow. It allows stakeholders to have a perception on the project delivery.
- Collaborating and communicating: Geigr.io is the middleman in the scrum meetings. It provides team members with unbiased and uninfluenced facts in order to enable better decision making.

4.2. Architecture

In this section, we will discuss the physical architecture of the approach. This allows to have a coherent view on the server-side of Geigr.io. Our solution is a containerized approach that scrapes data from different sources and instantiated and deployed on Hyperdev. So, before we move further into this section, we need to define a few terms. Docker is an open-source software platform for creating, deploying and managing virtualized approach containers on an operating system. The services of any approach that we want to deploy and its different libraries, configuration files,

dependencies and other components are grouped within a container similar to a virtual machine but have a significant advantage. While virtualization involves running many operating systems on a single system, the containers share the same operating system kernel and isolate the approach processes from the rest of the system.

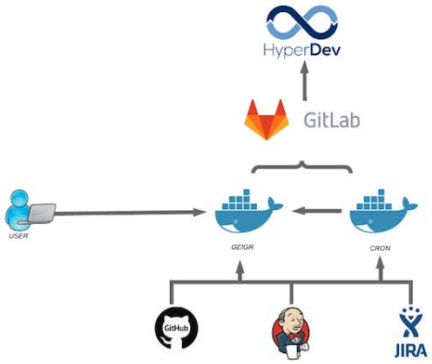

Fig.3. General architecture of the proposed Geigr.io

Docker has several advantages such as:

- Modularity: Docker's approach to containerization is based on the decomposition of approaches: the ability to repair or update a portion of an approach without having to disable the entire approach. In addition to this microservice-based approach, Docker allows you to share processes between different approaches almost as you would with a service-oriented architecture (SOA).
- Layers and control of image versions: Each Docker image file is composed of a series of layers. These layers are assembled in a single image. Each modification of the image creates a layer. Whenever a user executes a command, like run or copy, a new layer is created. Docker reuses these layers for the construction of new containers, speeding up the building process. Intermediate changes are shared between images, maximizing speed, size, and efficiency. That says layers overlay, says version control. With each change, a change log is updated to give you full control of your container's images.
- Restoration: The most interesting feature of layer layering is undoubtedly the restoration. Each image is composed of layers. Also, if the current iteration of an image does not suit you, you can restore the previous version. This feature promotes agile development and helps you implement CI/CD implementation practices at the tool level.
- Rapid deployment: Before, it took several days to set up new equipment, make it work, supply it and make it available. It was a complex and tedious process. Today, with Docker containers, you can do it all in just a few seconds. By creating a container for each process, you can quickly share similar processes with new approaches. In addition, since you do not need to restart the operating system to add or move a container, the deployment time is reduced further. Moreover, the speed of deployment is such that you can afford to easily and cost-effectively create and destroy your container data without any problem.

A. Geigr.io with Docker

Before we explain the procedure of implementing Docker. We need to familiarize with some key notes.

- Dockerfile: is a file that allows you to build an image (type "Docker") adapted and personified step by step. The goal is to create a personal image without having to use dozens of basic Docker commands. The dockerfile also allows you to keep track of your work. It also allows you to make quick adjustments because it is easily modifiable.
- Images: an image is an inert, immutable, file that is essentially a snapshot of a container. Images are created with the build command, and they will produce a container when run.
- Container: To use a metaphor for programming, if an image is a class, a container is an instance of a class. Containers are, the reason Docker gained such a popularity. They are lightweight and portable encapsulations of an environment in which to run approaches.
- Registry: a repository that groups together the images of the tech community.

We have briefly mentioned that we are running two processes. One is the approach itself and the other is the Cron job that scrapes data from different sources, to be analyzed within the approach. With docker, each process is usually fit within a container that includes everything needed to run the Geigr image. So, in order to implement Geigr with Docker we need to follow few steps which are illustrated in the figure 4 below:

- First, we need to create a container for the application. A snippet of this container is showcased in figure 4 below.

```
FROM thinkr/rfull
COPY demoshiny_*.tar.gz  /demoshiny.tar.gz
RUN R -e "install.packages('demoshiny.tar.gz', repos = NULL, type = 'source')"
COPY Rprofile.site /usr/local/lib/R/etc
EXPOSE 80
CMD ["R", "-e demoshiny::shiny_mon_app()"]
```

Fig.4. Geigr.io containerized.

- Second, we need to create a container that runs a background process to execute a scheduler, we call it CRON job.
- Then, we will to write our Dockerfile file which contains the necessary commands to be able to containerize the application.
- We will build the image.

B. Geigr and docker-compose

Since the docker file has two containers with each responsible for a single process, we need a tool that orchestrate both of them and that is when docker-compose comes in handy. Docker-compose is Docker official tool for managing, defining, orchestrating, linking, using, starting and stopping a set of containers. The purpose of docker-compose is to be able to do all these actions in a simple and fast way. The principle is to create a YAML configuration file: "docker-compose.yml".

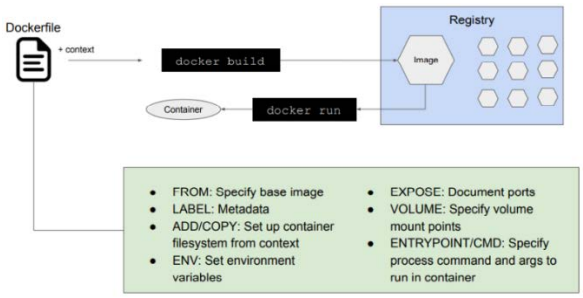

Fig.5. Geigr.io containerization with Docker.

This one will define the containers as one could do with the classic Docker commands.

C. Geigr and Gitlab-ci

Gitlab is an open-source platform for hosting and managing projects. From its dashboard, Gitlab provides a comprehensive view of current issues and tasks. It is a collaborative tool for developers in the realization of applications. In addition to these features, Gitlab also manages the integration and continuous deployment of the project by adding the.gitlabci.yml configuration file to the Git repository root directory. This article describes the key concepts of continuous integration in Gitlab that are, the configuration of the gitlab-ci.yml file, the jobs, the pipelines and the Runners.

Gitlab-ci has key features:

- Configuration of. gitlab-ci.yml : this configuration file is written in YAML. It configures the Gitlab project to use a Runner; thus each commit or push triggers the execution of the continuous integration pipelines.The data concerning the continuous integration (pipelines, jobs, environments, etc.) are displayed in the "CI / CD" menu of the vertical gitlab menu.
- Jobs: The YAML file defines a set of jobs (or tasks) that will have as mandatory constraints to be executed in a stage and always contain the script statement.
- A pipeline: is a group of jobs that are executed in stages.All jobs in the same step are run in parallel if there are enough simultaneous Runners available. The pipeline moves to the next step when the current jobs finish running without errors. If there is at least one job that fails, the pipeline stops executing.

Geigr.io is pushed to a deployment environment “hyperdev” through gitlab-ci.

D. Geigr and HyperDev

Geigr is running on HyperDev mainly for testing and deployment purposes. HyperDev is a continuous development platform which allows developers to quickly and seamlessly stand-up web applications without standing

up a server stack, so It's a fast way to get running code up on the internet without dealing with any administrative techniques that are not related to your code.

Briefly put, thanks to HyperDev, developers can develop, deploy and test their applications. The main characteristics of HyperDev are:

- HyperDev CI/CD: HyperDev provides teams with all the tools necessary for proper CI/CD pipelines. It doesn't only provide the tools, but they will also be configured and ready for use, including an example pipeline.
- HyperDev Dashboard: The HyperDev Dashboard makes it easy to developers to run their own development setup. All functionality is readily available through a well-defined API.
- Continuous Delivery and DevOps: Often teams are also responsible for maintaining the acceptance and/or production environment by means of CI/CD pipelines. Teams need to not only develop the software but also have the capabilities to ensure the software can be put into production in a controlled fashion.
- Build on Docker: HyperDev is built on top of Docker - the leading tool in containerization. The platform uses standards, such as docker-compose, to ensure a smooth transition to production and integration with all other services.
- Secure Cloud HyperDev: it runs a private cloud per tenant on top of virtual servers located in The Netherlands. All services (CI/CD tooling and applications) are accessible through a VPN. This way, the safety of data will be ensured. Consequently, Geigr.io extracts data from different data sources and is containerized via Docker. Since, we are running a cron-as-container in the background, it is built with docker-compose. Finally, we have used Gitlab-ci to deploy geigr on hyperDev. The approach is also utilized on the cloud.

4.3. Data Sources

Geigr is a data-driven platforms that collects data from different sources including Jira, github, jenkins and sonarqube. We will define in this subsection, all the data sources that haven't been introduced yet.

Jenkins is a continuous integration software tool. It is an open-source software, developed using the Java programming language. It allows you to test and report changes made on a large code base in real time. By using this software, developers can detect and solve problems in a codebase and quickly. As a result, new build tests can be automated, making it easier to incorporate changes into a project on an ongoing basis. Jenkin's goal is to accelerate software development through automation. Jenkins allows the integration of all stages of the development cycle.

Jira is an agile tool for development teams that allows us to plan, track and manage all agile software development projects, from agile tables to reports.

GitHub is a web-based versioning and collaboration platform for software developers. Git is used to store the source code of a project and track the complete history of any changes made to that code. It allows developers to collaborate more effectively on a project by providing tools to manage potentially conflicting changes from multiple developers.

Gitlab is a platform for hosting and managing web projects from A to Z. Presented as the platform of modern developers, it offers the possibility to manage its Git repositories and thus better understand the version management of your source code. Originally known for its ability to manage source code versions, Gitlab has developed in recent years to become a key tool for web project management.

Now, we are keen about the different parts that shaped the Geigr.io approach. As a consequence, we are aware that Geigr.io is a containerized tool, deployed on hyperDev and triggered through a plugin and other data sources. The data collected is interpreted in a meaningful way into multiple charts, graphs and statistics. However, the predictive modeling part is still blur until this point. That's why prediction models will be the focal point of the next section.

5. Predictive Modeling based on Machine Learning

In this section, we will discuss random estimations and the reason behind machine learning implementation [13, 14] in order to answer two predictive problems. One is about issue resolution time and the other is about velocity. Moreover, we will elaborate on how we managed to solve both of these research questions by defining the selected approaches and measure their accuracy.

Planning is a crucial factor in decision-making procedures and without a support system, it can be difficult and time consuming. Relying on estimators for scheduling is considered a wrong move simply because estimators are biased and can be influenced by people opinions and perspectives. Thus, for our solution, we went with predictive modeling to foresee behavioral performance outcome within the team.

5.1. Issue resolution time prediction

Companies have been estimating when the task in hand will be resolved for the purpose of achieving higher performance but as we have discussed earlier that approach can be highly biased and thus it will affect the outcome.

The goal for predicting resolution time is to yield a glimpse to the team's future conducts and prevent risks of postponement.

Delays are caused by not finishing the issues assigned per sprint on time, so the team has to shift them to the next sprint which consequently results in delays of product delivery.

A. Issue features explanation

Before we start our discussion, we first need to describe what we are resolving. Our unknown target is the resolution time, and our predictors involve every attribute that may affect our target variable whether implicitly or explicitly. In this section we will define these attributes and the related predictors.

A quick flashback on section 2 where we have discussed the use of Jira API for getting a Json object containing all the fields describing an issue. These attributes are explained in detail in table 1 below and they are fit to all jira issues type whether it was a bug, a story or a task:

The attributes mentioned above are often referred to as features under the spectrum of machine learning. Before choosing the model or let's say in simpler terms before choosing the algorithm that answers to the inquiry of this part of research, we need to select the features that are influencing this inquiry.

Table 1. Jira issue features.

Attribute	Type	Value
ID/key	Text	Automatically assigned to distinguish between different issues.
Title	Text	Represents a summary of issue.
Description	Text	Gives a detailed information about the issue.
Priority	ENUM	Blocker, Critical, High, Immediate, Low, Normal.
Type	ENUM	Bug, Epic, Gw-issue, Improvement, Incident, Investigation, New Feature, Project, Story, Sub-task, Task, Technical task.
Creation date	Datetime object	Indicates when the issue was created on jira dashboard.
Update date	Datetime object	Indicates when the issue was updated on jira dashboard.
Resolution Date	Datetime object	Indicates when the issue was marked done.
Time spent	Text	Indicates how many days were spent on an issue.
Reporter	Text	Name of the person who created the issue.
Comments	Text	The multiple remarks passed to jira from developers.

Our first research question is as follows: given the user's history data, how do we know when a specific issue will hit the status resolved? What are the impacts of certain factors over resolving an issue. To answer this question, we need to explore our data.

B. Data exploration and correlation coefficient

We will measure the time spent on each issue. Time spent on an issue is the difference between the creation date and the resolution date. These two attributes are given by the user. Next, we will go through the features one by one and measure the correlation between each attribute describing an issue and the time spent on that issue.

In statistics, correlation measures the degree of linear re lationship between two random variables. If two variables are correlated, it doesn't mean that one has caused the other. In short, correlation is a score called "correlation coefficient" calculated between two variables that gives away how much the first variable depends on the other one. Correlation coefficient noted 'r' varies between [-1..1] and it is explained as follows:

- If r is closer to 0 then there is no relationship between the two variables.
- If r is closer to 1 then there is a strong positive relationship between the two variables. In case one gets larger, the other one gets larger too.
- f r is closer to -1 then there is a strong negative relationship between the two variables. In case one gets larger, the one gets smaller.

The assessment of feature correlations will depend on the spearman rank method because our data points are both continuous and categorical. Moreover, the mathematical equation used to perform the spearman rank r(s) is given by the next equation:

$$r_i(s) = 1 - \frac{6\sum D^2}{n(n^2-1)} \quad (1)$$

D = difference between ranks of corresponding variables. n = number of observations.

C. Correlation between the reporter and resolution time

We inspected whether the person who reported the issue has any influence on the pace of its resolution while taking into account the reporter's reputation and seniority. We started by plotting the relationship as shown in the graph in figure 6.

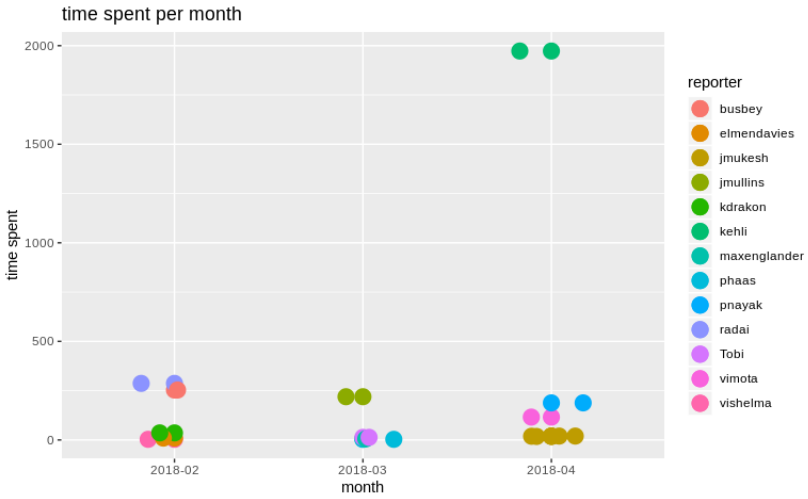

Fig.6. Correlation between reporter and resolution time

Through this graph we notice that the time spent of certain reporters is either two long (matter of weeks) or it is too short. Thus, we can assume that the reporter influences the pace of resolution. To prove this further, we have calculated the correlation coefficient r which is equal to: Rs = 0.432 In our business case, the value presented by the coefficient R is enough to deduce a relationship between both features.

D. Correlation between the transitions and the resolution time

Transitions are the different stages an issue proceed before hitting the status resolved. Transitions can be from in progress to resolved or open to resolved directly. We will showcase in graph in figure 7 the last five months observations since the end of our data points to verify whether there is an obvious association between both variables.

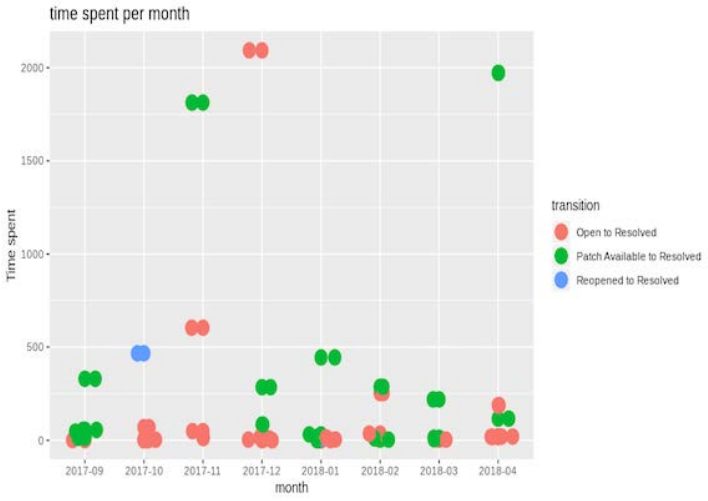

Fig.7. Correlation between transition and resolution time.

From the graph in figure 7 above, we are unable to observe a relationship between the transitions and the resolution time. In this case, we tend to rely on the coefficient r. The pearson coefficient induced a value of −0,5325474 indicates a negative relation between both variables.

Another feature to consider is the issue's priority, we have produced a plot graph, figure 8, that displays the time spent on an issue in regard to its priority.

The feature priority is obviously affecting the duration spent on an issue. Issues with critical or blocker priority are taking less time than the other as shown in the graph, figure 9, and trivial are taking little time to be resolved.

E. Correlation between resolution time and issue type

The type is an important feature to be aware of, but does it affect the resolution time? As usual, we will first plot it to see if there is a need to calculate the correlation coefficient.

The graph represented by the figure 10 shows that new features, tasks and wishes are resolved faster than bugs. This is persuaded by the correlation coefficient as r = 0,38 which in our case is enough to work out a link between both time spent and issue type.

```
> res <- calculate_correlations(transitions$days_since_open, transitions$transition)

> res
[1] -0.5325474
```

Fig.8. Output console.

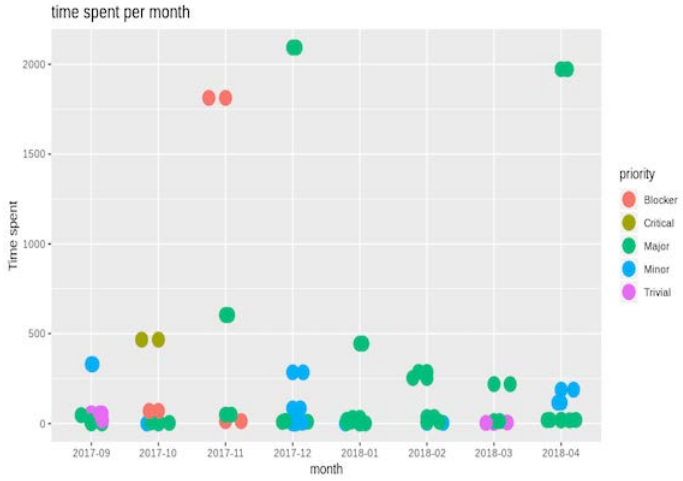

Fig.9. Correlation between resolution time and priority.

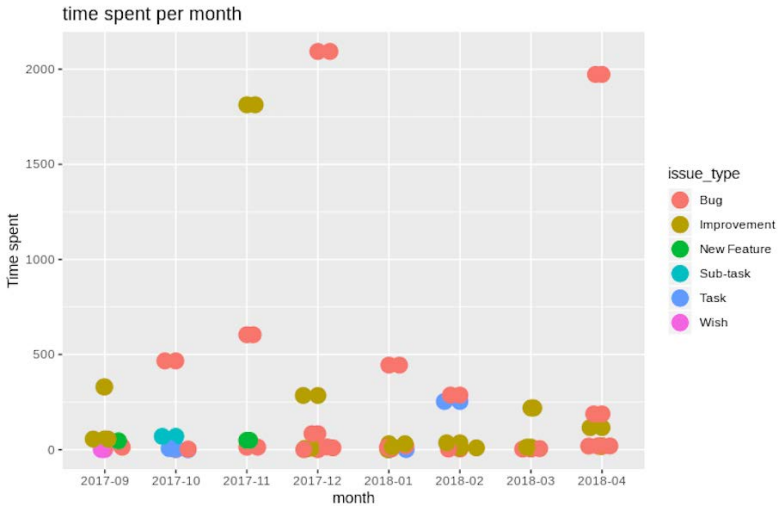

Fig.10. Correlation between resolution time and issue type.

F. Correlation between titles and descriptions

Assuming that two issues with similar descriptions are resolved in the same amount of time. This method will be explained in the next section and proved in section 5.

5.2. Selected models

By now, we went through all the features affecting the predictions. Text, priority, transition, type, and reporter will be the input of our models. But since text and other attributes cannot be combined into one model. We will try different approaches and assess them until we figure out which one gives the most accurate results.

A. Text based model

This model will be based on text similarity. The groundwork is to compare between the title or description of the issue to resolve and the rest of the titles or descriptions. When a match is found and by match, we mean the most similar text then we assume that the time spent on both issues will be equivalent. Text similarity mostly depends on the number of analogous words, and it is fulfilled by following three steps:

Many works proposed the uses of the similarity measure, in [15] the authors proposed a method titled "Measure of similarity between fuzzy concepts for identification of fuzzy user's requests in fuzzy semantic networks" that can measure the distance between two concepts in a semantic network. Also, in [16] the authors proposed another method titled "Linguistic Variables Definition by Membership Function and Measure of Similarity" to measure the degree of similarity between two concept's variables. Another model titled "Possibilistic Pertinence Feedback and Semantic Networks for Goal's Extraction" have been also proposed by [17] to extract pertinent data from possibilistic networks based on the similarity measure.

- First, we need to perform data preprocessing on the textual data.
- Second, we need to define a similarity matrix using tf.idf technique [18, 19]. A similarity matrix consists of tokens where we assign a value of importance to each one and this is applicable to the entire corpus or titles.
- Then we go over our corpus, and we calculate a cosine distance matrix for each sentence. This matrix will elucidate the degree of similitude between a specific title and all the rest.
- K-nearest neighbor seizes the top K similar titles to our query and calculates the mean.

As a result, the time spent on resolving the query issue corresponds to the output of the KNN algorithm.

B. Data preprocessing of textual data

Before we start the modeling process, it is crucial to remove noise from text in order to achieve better accuracy scores. Data preprocessing involved these steps: (i) Lowercasing the text, (ii) Removing numbers from text, (iii) Removing all punctuation from text, (iv) Removing excessive whitespaces, (v) Removing stop words, (vi) Remove symbols and keep only alphanumeric characters and (vii) Applying Stemming to words.

For further explanation of the new terms stated above:

- Stop words are words that are repeated frequently in linguistic sense such as: the, as, a, etc.
- Stemming is reducing a word to their most basic originated syllables. For example: Playing —> Play, Player —> Play, Played —> Play.

C. Term frequency-inverse document frequency

tf.idf is a numeric metric that reflects how important a word is to a corpus. The *tf.idf* value increases proportionally to the word occurrence. 83% of text-based recommender systems in digital libraries use *tf.idf* weighting scheme. This value is based on two main concepts: term frequency and inverse document frequency as given by the next equation.

$$tf.idf(t,d,D) = tf(t,d) * idf(t,d) \quad (2)$$

Term frequency *tf*: it computes how often the word occurs in the document. For large documents, augmented frequency technique fits better than the simpler ones.

$$tf(t,d) = 0.5 + 0.5 * \frac{f(t,d)}{\max\{f'(t,d): f' \in d\}} \quad (3)$$

With:

f(t,d) : is the raw frequency of the term in the document, and $\max\{f'(t,d): f' \in d\}$: represents the raw frequency of the most occurring term in the document.

We can notice that, on one hand, *tf* increases the words that have been frequently occurring and, on the other hand, it also increases stop words that are irrelevant to classify documents that explains the surge towards *idf*. Inverse document frequency idf measures give the degree of how much information the word provides across all the documents, the rarest the word is the most meaningful.

$$idf(t,d) = log\frac{N}{|\{d \in D: t \in d\}|} \quad (4)$$

With:

N: is the total number of documents in the corpus $N=|D|$, and $|\{d \in D: t \in d\}|$: is the number of documents where the term *t* appears.

D. Cosine similarity

Cosine similarity, as defined in [20], induces a number in the [0, 1] interval that stipulate the cosine of the angle between two non-null vectors A and B. The formula of cosine similarity is derived from the euclidean dot product:

$$A.B = \|A\|.\|B\| \cos\theta \tag{5}$$

After further demonstration of the multiplicative product, we conclude that Cosine distance is given by the next equation:

$$\cos\theta = \frac{A.B}{\|A\|.\|B\|} \tag{6}$$

E. K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a machine learning algorithm widely used for its ease of interpretation, low computation costs and reasonable predictive power. Following the quote "Birds of a feather flock together", KNN reckons that each similar variables are in close proximity. With that being said, Let's go into details while looking at the data points defined in the figure 11 below where titles are assigned.

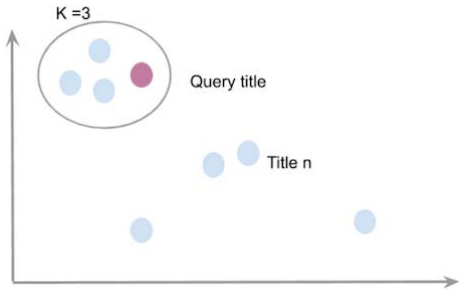

Fig.11. K-Nearest Neighbor (k =3).

We notice that *K* is a parameter indicating the number of neighbors to be averaged in time of prediction. The algorithm relies on cosine similarity to measure the three closest variables to the query and calculates the mean of their time spent.

Text based model is the first process we tried to predict resolution time; results will be discussed in the following section.

F. Meta-info based model

Most of machine learning tasks nowadays are classification tasks due to the need of decision-making. Given many classes or categories, classification algorithms aim to predict to which specific class a certain variable belongs.

We can look at the problem in hand as a classification assignment. Given many attributes, our goal is to classify the issue's resolution time into *N* time intervals or *N* classes.

These classes would be based on the average days spent on an issue. For our use case, we have concluded five classes of inclusion then we have selected decision tree and random forest models to fil the prediction task.

G. Decision tree model

Similar to how a tree-like algorithm operates, decision trees are a deterministic tool to classify an input. They rely on event outcomes, possible consequences and transition values.

It is a very common among machine learning engineers, it supports them to perform an accurate strategy to make a decision.

A decision tree consists of three types of nodes:

- Decision nodes – typically represented by squares
- Chance nodes – typically represented by circles
- End nodes – typically represented by triangles

To transit from one node to the other, Decision trees follow a probabilistic pattern. Depending on the probability of the input at a certain level, the transition is made to the lower level. If there is no lower levels then the output of the decision tree is chosen among the leafs.

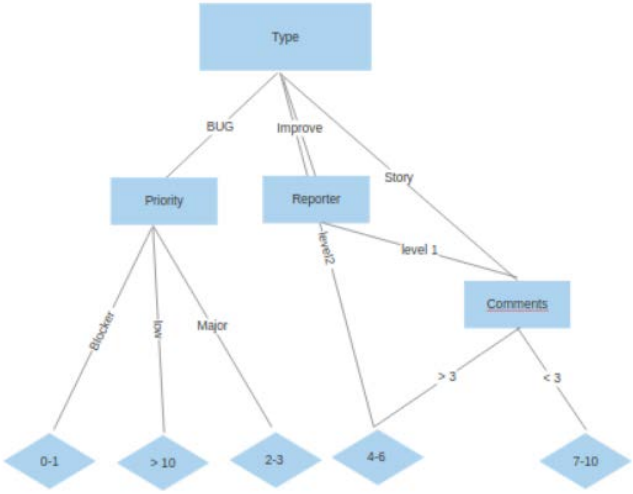

Fig.12. Decision tree for Resolution time prediction.

Time resolution decision tree is based on the issue feature. We noticed that there is categorical and numerical input. To develop a decision tree, we have to convert categorical variables or what as we talked about it earlier text type variable into levels. Each level will hold a type or a class of value and perform some normalization processes to make them less biased and more understandable to the machine.

H. Random forest model

While a decision tree is a fair classification algorithm, yet it misses a lot of possible combinations that multiple decision trees can cover and that possibly sums the notion of random forest.

Apart from being derived from decision trees, random forest is an algorithm that has a good prediction power.

Furthermore, Random Forest forms multiple decision trees at training time and outputs a class where the variable of interest belongs.

Random forest approach mainly performs with bagging which demonstrates into the following:

Given a training data set X ranges from [1..n] and labels Y[1..n], bagging of B steps or epochs creates a subsample of the training set and fits it into a collection of trees. Prediction is made by averaging the vote from each tree.

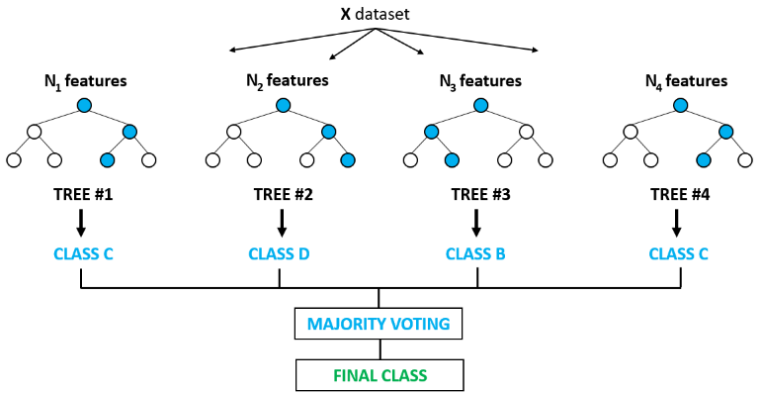

Fig.13. Random forest illustration.

As shown in the illustrative figure 13, random forest is a clustering algorithm that gathers results from different decision trees and classify the input based on majority decision.

In our case, we have hundreds of possible combinations of the feature:

- If type is bug, reporter is level 1 and comments count is 4.
- If type is story, reporter is level 1 and priority is low.
- If type is bug, reporter is level 1, priority is major.

Based on how many theoretical cases we can convey from the features; random forest algorithm tends to be of more effective that one simple decision tree.

We have implemented RF for our solution and the results will be discussed in the upcoming section.

5.3. Velocity forecast

We have triggered earlier the definition of velocity. To summarize the notion of velocity, is how much work can a team accomplish within a sprint. In scrum, work accomplished whether it was a task, story, new feature, improvement or a bug is measured by its level of complexity. Let's pretend, that you and your team defined a sprint of two weeks period, and the backlog of that sprint is twelve issues. For each issue, you and your team has defined a weight that measures how compound that issue might be. Let's say that you assign 10 points if it is really hard and 2 if the issue is trivial.

The summation of these points is called velocity and to forecast the current sprint velocity, we are mainly relying on the history data points. Moreover, for our forecast, we have adopted the principle of past teamwork can give up a glimpse about future teamwork.

The inspiration from the past to foresee the future is what we call in machine learning **auto-correlation**.

5.4. Auto-correlation

We have discussed at an earlier section that correlation is the linear relationship between two random variables. The only difference with what we stated, and autocorrelation is that the linear relationship is between the variable and its own lagged values.

$$Y(t) = A1.\mathrm{Y(t-1)} + \mathrm{A2.Y(t-2)} + \cdots + \mathrm{An.Y1} \tag{7}$$

For our use case, we need to prove that velocity is in fact an auto-correlated variables. For that purpose, we will convert the velocity variable into time series where each sprint point will be defined by a time slot. Figure 14 below, exhibits the velocity as a time series.

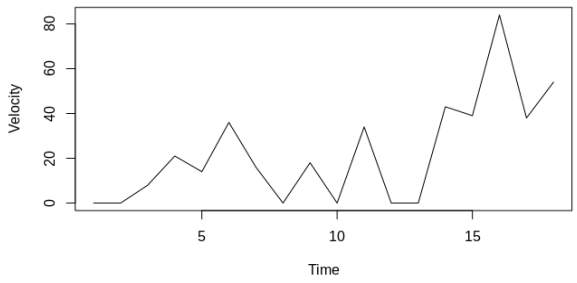

Fig.14. Velocity auto-correlation.

We can distinguish from figure 18 that there is a multiplication when displaying the velocity against time. Even Though, the time of the sprint is fixed in advance and keeps the same period of time, there is no predictive pattern on how it will turn out in the future also there is no trend or seasonality because we cannot follow up on a constant seasonal pattern. However, we can interpret a cycle every other four sprints, the velocity is alternating but almost always decreases on the fourth sprint.

Cyclic behavior is considered **stationary** and for that reason, we will elaborate on the research that has been made to deal with stationary time series.

5.5. Auto regression model

Auto regression is the act of forecasting the variable of interest using a linear combination of its lagged values. So, a mathematical formula from an autoregressive model would be:

$$y(t) = c + \alpha 1.y(t-1) + \alpha 2.y(t-2) + \alpha 3.y(t-3) + \cdots + \alpha n.y(t-n) + \varepsilon(t) \tag{8}$$

Autoregression models are really useful for handling large history data points and detecting patterns that we might have missed when performing data analysis.

Demonstrating the equation of order n above where there are n predictors or lagged values reveals a few characteristics about AR models:

When $\alpha(1..n) = 0$,then $y(t) = c +$ white noise.

When white noise = $\varepsilon(t)$ = 0 then the predictors will be overlapping.

When $\alpha(1..n) = 0$= 1 and c=0, then y(t)= random walk.

When $\alpha(1..n) =<$ 0 then y(t) tends to oscillate between negative and positive.

The noise has a zero variance, and it refers that the y(t-n) predictors are distributed with the same intensity at every frequency and all frequencies are visible. The higher the order p is, the more restrictions we need to assign the model in order to correct the accuracy.

5.6. Optimization technique for Autoregressive models

One of the optimisation techniques that we have used in order to deal with biased output is applying moving average smoothing on the residuals.

Moving average smoothing is a common exponential process in the gamut of forecasting and it is derived from autoregressive models. So instead of relying on lagged predictors, with MA smoothing we will be using Past white noises.

To define exponential smoothing, we can safely say that it consists of assigning weights to different observations while keeping in mind that recent observations are more important than the former ones. A further explanation of the procedure is: the further observations go back in time, the further the weights decay. There are multiple exponential smoothing methods but for our approach we will wield Holt-Winters seasonal method. The reason behind this is that we operate per sprint so there is a seasonal concept in the time series, maybe this seasonality is hindered due to the velocity, but it always persists in the overall forecast. The residuals which we're using for result optimization are mainly prediction errors. It unveils any seasonal patterns that we may have overlooked in the time of forecast. Many methods are used for exponential smoothing tasks starting from a simple exponential smoothing model to the sophisticated ones. For our use case we will contrive with Holt-winters model.

5.7. Holt-winters model

Holt-winters is extended from a former statistical method called Holt driven by the motivation of capturing seasonal patterns of a time series.Holt-winters encapsulates two main approaches either additive or multiplicative.

- Additive approach is used when the time series remain constant on average throughout time.
- Multiplicative approach is used when the time series increases uniformly through time.

To Improve on residuals, it is more fit to our aim to use the additive parameter. Holt-winters additive method uses three main equations:

Level equation L(t) which expresses the level of certainty in the prediction: usually it varies from 25% to 90%:

$$L(t) = \alpha * (Y(t) - S(t-m) + (1-\alpha) * (L(t-1) + b(t-1))$$

Trend equation b(t) to capture any repetitive patterns:

$$B(t) = \beta * (L(t) - L(t-1)) + (1-\beta) * b(t-1) \quad (9)$$

Seasonal equation S(t) is a weighted average between the current season and the last m period.

$$S(t) = \lambda * (Y(t) - L(t)) + (1-\lambda) * S(t-m) \quad (10)$$

The resulting forecast equation $\hat{y}$(t+m) that implements the levels, trends and seasonality for a period m is as follows:

$$\hat{Y}(T+M) = L(T) + M * B(T) + S(T+M-S) \quad (11)$$

5.8. Evaluation metrics

Evaluation metrics are a set of error calculation that validates the accuracy of the model. We often use statistical method that measures margin errors between real values and predicted one. For each of our use cases, we assessed the related algorithms differently.

Text similarity model is assessed using cross validation technique. **Cross-validation** is a usual practice in machine learning that consists of out-of-sample error calculation. For instance, we had splited the data into one large batch of training and one small batch for testing.

After multiple epochs of training the model on the first batch, we later use the test set to validate the model's ability to predict on data that it has never been seen before. We then calculate how many data points derived from the unknown data and predicted incorrectly. Whereas Random Forest algorithm is commonly validated by an **OBB** since it is a method that relies on subsampling or bagging. Given a data point Xi from a training set, we estimate the error by calculating the mean of the prediction of *Xi* on the tree that doesn't contain the subsample.

Last but not least, autoregressive models are validated by loss functions. We have anticipated two main estimators:

- MAPE: It is a metric used in regression and autoregression approaches. It is calculated via:

$$\text{MAPE} = \frac{100\%}{N}\sum_{t=1}^{n} \left|\frac{A(T)-F(T)}{A(T)}\right| \quad (12)$$

With:
A(t) = actual value,
F(t) = predicted value.

- MAE: is also a practice to compare the observations to the future outcomes. Yet, it presents a clearer vision about the horizontal distance between the historical variables XI and the predicted variables Yi:

$$\text{MAE} = \frac{\sum_{t=1}^{n} |YI-XI|}{N} \quad (13)$$

In this section, we have expanded our knowledge about the predictive models implemented in the application. The realization of both uses cases allowed us to explore different techniques such as text similarity, random forest, and autoregressive models. We then evaluated each one of them and carried out the most accurate between them.

6. Experimental Study and Results Analysis

This section focuses on the implementation of Geigr.io. We dive deeper into the work environment including the programming language, the development tools and libraries. Afterwards, we will mention the data collection process and most importantly the different views that shaped the application. Then, we will mention the evaluation results of prediction models to finally conclude with an overlook of the website.

6.1. Work environment and development tools

Geigr was created using R programming language for backend development combined with embedded web components built with Shiny.

R is an integrated environment for data manipulation, calculation and chart preparation. However, it's not just an "other" statistical environment (such as SAS), but also a complete and autonomous programming language.

The R is a language mainly inspired by S and Scheme and is interpreted, that is, it requires another program - the interpreter - for its commands are executed. In contrast, language programs compiled, such as C or C++, are first converted to machine code by the compiler and then directly executed by the computer. This means that when you write code in R, it is not possible to plead waiting for the end of the compilation phase.

RStudio teams have developed Shiny, a package that allows you to create dynamic web applications. It allows for standalone applications as well as CSS, HTML and Javascript extensions. Shiny package has two major components: a UI class which serves as client-side interface and a Server class for server-side tasks.

After careful consideration, the choice of these technologies wasn't random but not limited to several advantages:

- R language is based on the notion of vectors, which simplifies mathematical calculations and considerably reduces the use of iterative structures (for loops, while).
- R has no typing or mandatory declaration of variables.
- R come with short programs, usually just a few lines of code.
- R allows for short development time due to its effective libraries.
- It supports statistical machine learning algorithms.
- It comes with interactive data visualization tools such as plotly and ggplot.
- It is a powerful tool for data analysis due to a solid inference of dataframes.

RStudio is a newly released tool that fills a gap in the collection of tools associated with R: it is a functional, free, multi-platform IDE environment.

An IDE is not a graphical interface in the sense of SPSS or Modalisa, which would allow the software to be used through menus and dialogs: it is an environment that facilitates the input, the execution of code, visualization of results.

RStudio is cross-platform, so you can download it and run it on Windows, Mac OS X or Linux.

6.2. Libraries and frameworks

Plotly is a library for creating interactive charts and graphs. Not only providing an easy-to-implement plots such as line, bar and pie plots but also forecast charts. As well as, embedded with ggplot graphs to make them more interactive. This library is built upon the javascript package "plotly.js" and is available for R, python and javascript programming languages.

Forecast is a library that offers statistical prediction models for univariate time series including exponential smoothing, autoregressive models.

6.3. Data collection

Geigr.io garners data from different sources such as Jira, gitlab, Jenkins, Sonarqube. We will discuss how the day is gathered and what measures are we talking to protect its privacy. Data that has been scraped will be passed anonymously due to hashing.

The hash procedure is the transformation of a character string into a value or a fixed length key, usually shorter, representing the original string. The hash is used in particular to index and retrieve the elements of a database. It is faster to find the item based on the reduced hash key rather than using the original value. This function is also used in many encryption algorithms.

For the next phase, the data will be passed encrypted from a java environment to a middleware that will serve for decrypting it and sent it to the application that runs in R.

The return step for the deciphered data will be from the application itself to the middleware and back to display the Geigr index in a web component on Jira dashboard.

One of the advantages of Geigr.io, is that it gathers data from different sources so the user can make the best of what was yielded during the project.

We have talked about Jira, but for the rest we are imparting an Api for each source in order to read data and convert it to a convenient format.

6.4. Geigr.io views

The approach focuses on four different paths. These paths demonstrate the use of Geigr.io which was built to achieve several purposes.

These features are conveyed into four main views: Decide, plan, Collaborate and Geigr index.

The user can download a full report of what was stated on the platform plus more statistics that we couldn't add for compact complexity reasons. Yet, the button next to the project selectors serves to execute this functionality. A PDF document is downloaded locally on the client-side.

A. Decide view

This outlook of this fragment, is made of three major sections:

- The first gives away an overview of the product. It flaunts the forecasted delivery date with the level of certainty set on 80% and the budget invested in the project. The figure 16 above, shows the first section of this view tested with client data.
- The second section is for quality control. There are two charts that define both the data quality by either tracking Geigr index fluctuation or the performance undulation of the team. Performance undulation is measured by the quantity of the work that has been done compared to the bugs that has been induced. The figure 15 displays this particular point.

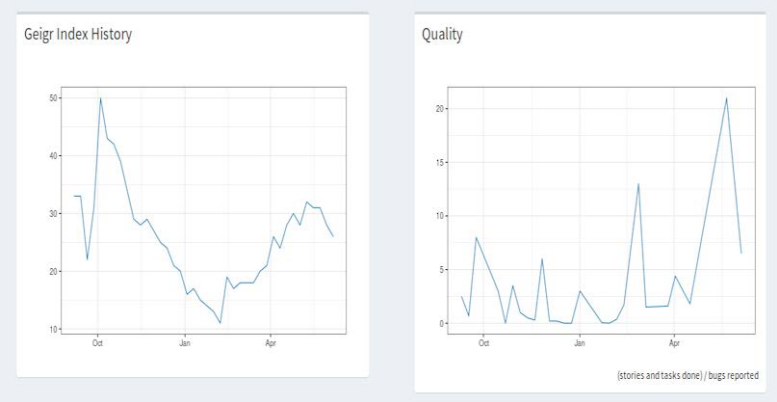

Fig.15. Decide view, Quality control.

- The third section on decide is the tracking of three cardinal aspects of any project: Time, Budget and Risk. We try to display all statistics that informs the client on these three levels. We can distinguish through figure 16 that we aren't just displaying historic data but also what we predict concerning the flow of the project. All for decision-making support.

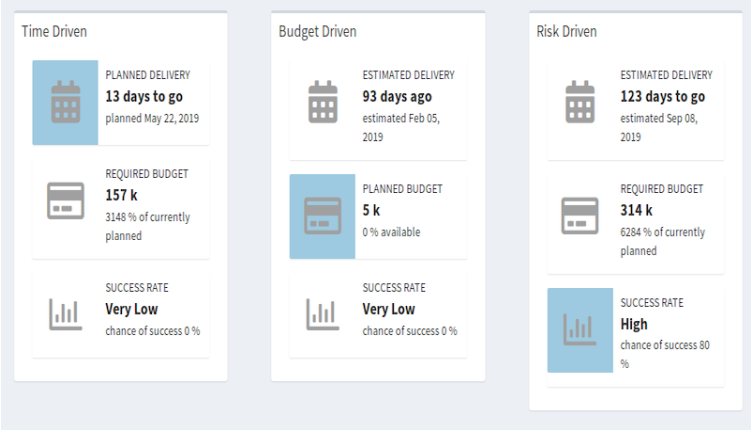

Fig.16. Decide view, project estimations.

B. Plan view

This fragment grants a more in-depth outlook of the project. It has two main charts. Each chart is the result of a single predictive model which we have discussed in the previous section.

- On the one hand, there is the velocity chart which exhibits the performance of the team per sprint. Each data point conveys the quantity of work that the team managed to complete within two-weeks period. As the figure 17 shows, we managed to output both the past velocity and the predicted one.
- On the other hand, precisely on the right side of figure 17 we have included a graph that shows all the issues created during the project and their resolution time. If there are unfinished issues, then we output the predicted resolution time.

Both plots are interactive which means that we can select the time slots to when we want to review our inputs, also, we can only select the category of data we are interested in.

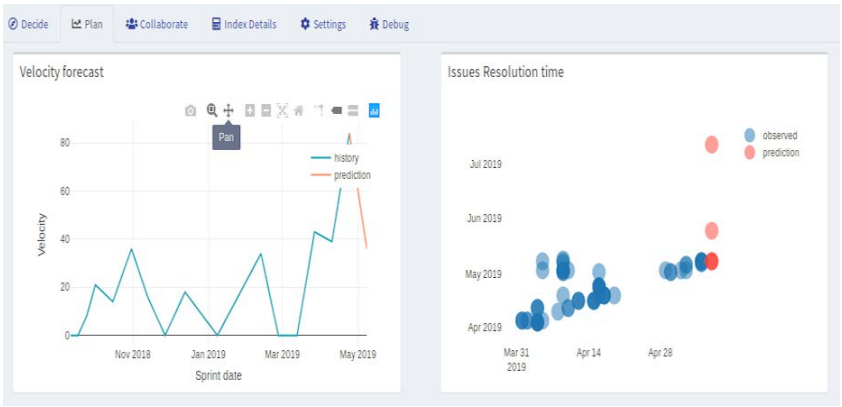

Fig.17. Plan view, Performance charts.

The rest of the view is dedicated for sprint tracking. It includes details about the past sprint and the current one. It also showcases how much work was done in terms of issues, tasks and points.

C. Collaborate view

The focal point of this section is the performance per person. Each member, how much does he contribute to the advancement of the project. As shown in figure 18 the left component mainly shows the details about points, time spent and averages of how much added value per team within a week of work.

Moreover, the right side is more about the input of each member individually measured by the number of commits in the last two weeks.

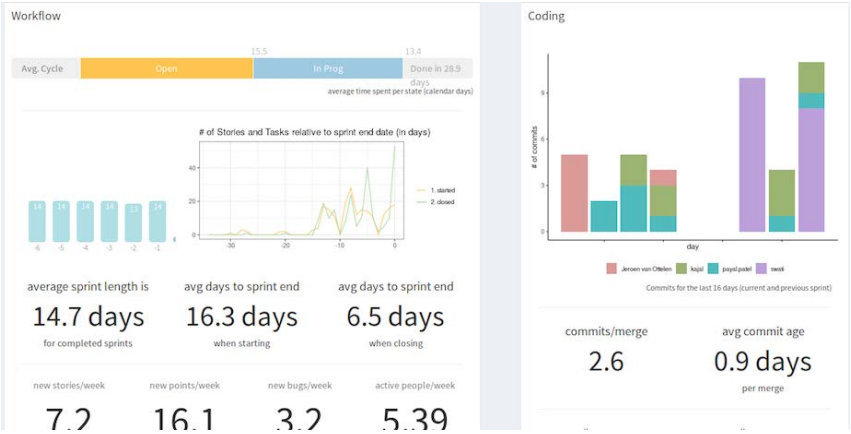

Fig.18. Collaboration view, averaging statistics.

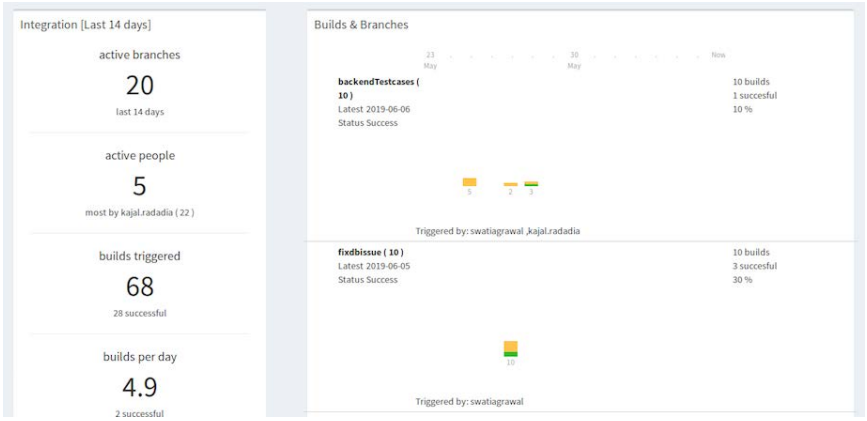

Fig.19. Collaboration view, Integration data.

On the bottom of this segment and as stated in figure 19, we are able to preview CI/CD information like build and status of the build which are derived from jenkins data and embedded to our approach via the Api. We can also see averages of data extracted from git, we can see active branches and delete those that hasn't been effective for a long time.

The correlation between the quality of data and the forecast exists and it is represented by a percentage called Geigr index. Geigr index is calculated through the following equation:

$$Geigr\_Index(\%) = Mean(\%Estimation, \%completeness, \%stability) \quad (14)$$

The three metrics defining the Geigr index are:

- Estimation outlier represents the pace of issue resolution. It is calculated through a median between the ones that were resolved fast and those that took a long time. In this case, the pace of an issue is defined by the number of points divided by the days that the issue spent in the 'IN_PROGRESS" status.
- Completeness reveals how many stories were assigned points along with an assignee and at the same time they marked done or in progress. Completeness levels varies between 30%-70%. This is due to the lack of commitments in the agile methodology practice.
- Stability metric is an indication of the constant decrease or increase in the velocity within a team. Precisely, the number of points done per week.

From the figure 20 above, we can confirm the metrics above and gain the ability to select the columns of data we want to review. This allows the user to verify the persistence of outliers or anomalies.

D. Settings view

- Alongside with configurations, we have included a chart of daily completed points and the forecasted ones within 13-days period with regard to the different settings and the confidence levels of forecasts. The confidence levels range from 25% to 90%.
- By now, we have gained enough knowledge to realize that the Data is loaded automatically to Geigr.io. Data

collected from four data sources: Gitlab, Jenkins, Jira and Sonarqube. However, what if we the user had his own chunk of files that we want to analyse?

- Then this view is meant for that plus the clients that relies on excel data files instead of agile software. Through this view, as shown in figure 20 we also provide the possibility to take a snapshot from your data per user request.

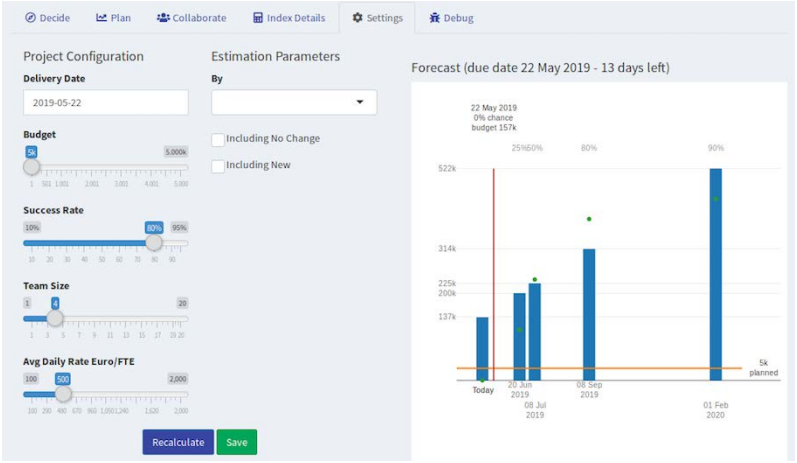

Fig.20. Settings view.

6.4. Predictive models evaluation

The consequence of knowing when the issue will be resolved is that a user will be able to take his measurements in advance and act upon it. As an example, if all the issues created for the current sprint will be done before the due date, then there is room for more issues thus an increase of the team productivity. This will directly influence the overall performance.

However, if the model foresees that there is a risk of not finishing the sprint on time then the product owner can add more resources in order to avoid this delay.

For this use case, we have already discussed the selected models. We have approached this research question from two perspectives:

- Text similarity [20-22] assuming that two issues similar in description and title will take approximately a close amount of time to be resolved. For this approach, we have used k-nearest neighbor. Before implementing k-nearest neighbor, we have gone through text preprocessing where we have stemmed and prepared the textual data. We have also built the TFIDF matrix then calculated cosine similarities. To validate the model, we are depending on cross-validation techniques, so we measured the error between predicted values and the actual ones. The figure 25 shows a sample of two similar issues according to the KNN algorithm along with their resolution time. This figure shows two similar tasks both related to the back end and both for the user interface. Also, the two issues have close resolution time.

```
wafa@wafa-ThinkPad-T430:~/sprint_tracker/Apis$ python3 text_similarity.py
 The actual title :  Backend - getallactiverecruiters / user
time spent the issue =  13.0
score of similarity =  0.11461382047224782
Most similar title in the corpus :   Backend // getbypartyid / user
time spent of similar issue =  10.0
```

Fig.21. Text Similarity Sample.

- Feature selection is relying on the input features to determine the category of the output. We have selected multiple attributes that are assigned to the historical issues in order to classify the input. We settled on type, priority, reporter, and transition status to dictate the class to which the issue belongs based on resolution time. For example, Class 1: 0-1 days, the issue will take 0 to a maximum of 1 day to be resolved.

The table 2 below, we are comparing the effect of features on the resulting predictions. The outcome increases with the increase of features which is reasonable in the sense of the more information you give the model, the more accuracy you will get.

From the table 3 above, we distinguish that before text preprocessing, Meta-data algorithm outperforms text-based

data by 8,046 on mean absolute error MAE scale. Improving textual data input resulted in higher accuracy of k-nearest neighbor algorithm. Consequently, we selected the model established through text-similarity over meta-information algorithm.

6.5. Velocity forecast

The use case of velocity anticipation is done through feeding the model historical data in order to predict the velocity of the two upcoming sprints. To perform this task, we have tested Holt-winters smoothing algorithm along with autoregression model.

Prior to detecting autocorrelations between the velocity variable at moment t and its lagged values, we ought to experiment with autoregressive models. We did not stop there but we also performed a smoothing of the moving average on the residuals. For further explanation, we iterated over the errors induced from autoregressive algorithm and corrected them using an equation that satisfies the rules of Moving average MA smoothing.

The figure 22 below, shows the MSE calculations along with a snippet comparing the actual values to the predicted ones.

```
predicted=7.250643, expected=2.000000
predicted=-1.045836, expected=4.000000
predicted=4.341746, expected=5.000000
predicted=6.092539, expected=6.000000
predicted=8.125890, expected=6.000000
predicted=1.635763, expected=6.000000
predicted=8.013019, expected=22.000000
predicted=12.281194, expected=21.000000
predicted=15.332082, expected=2.000000
predicted=6.024290, expected=2.000000
predicted=2.329722, expected=104.000000
predicted=48.567240, expected=1.000000
predicted=38.425972, expected=1.000000
predicted=-15.492368, expected=1.000000
predicted=11.157368, expected=20.000000
Test MSE: 178.545
```

Fig.22. Autoregressive model results.

Table 2. Random forest predictions with feature selection in terms of Mean Absolute Percentage Error (MAPE).

Features	(0-1)	(2-3)	(4-6)	(7-10)	(≥10)	Prediction rate
type, priority	0.16	0.23	0.23	0.13	0.127	12
type, priority, reporter	0.28	0.43	0.13	0.52	0.49	21,7
type, priority, reporter, transition	0. 384	0.428	0.469	0.526	0.302	42,7

Table 3. Comparison of text-based model and random forest in term of Mean Absolute Error (MAE).

Model	K-nearest neighbor	Random forest
Before text preprocessing	42,7	34,654
After text preprocessing	42,7	48,02

We have opted to look at the variable in hand as a univariate time series with seasonal variations. Those variations happen per sprint. The fittest model that meets these requirements is Holt-winters. We have evaluated the model using cross-validation and resulted are displays through the figure 23 below.

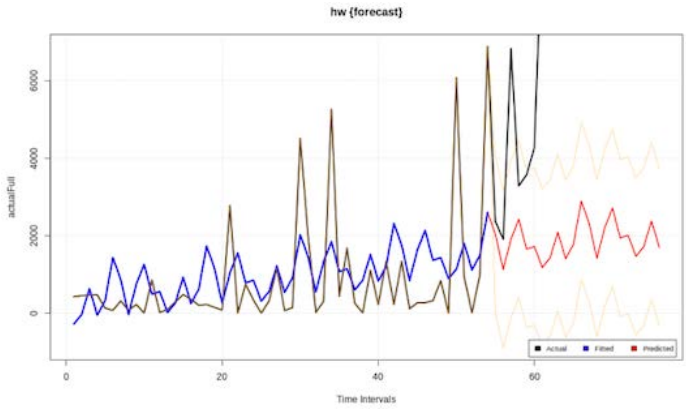

Fig.23. Holt-winters model, cross-validation plot.

6.6. Geigr.io website

The process of implementing the user data with Geigr.io is primarily established if the user himself has requested a demo via the application website https://geigr.io/.

The response time to the client request is usually very short. This website describes multiple characteristics of Geigr and leave the possibility for one-on-one meeting to reveal the multiple functionalities and technicalities of Geigr.io.

After an in-depth theoretical study of our solution, we have managed to create a proof of concept of Geigr.io platform. It is a self-sufficient business intelligence tool that combines data visualization, data analysis and predictive models to ensure the success of the project.

7. Conclusion and Future Works

In this paper, we presented our AI-powered solution to support agile practitioners. In fact, Geigr is a business intelligence tool that helps the client to schedule, plan and predict the team behavioral performance for the sake of advising stakeholders to take actions and act before hand. We initiated our study by familiarizing with business intelligence approaches, then we elaborated on state of the art. In this phase, we mentioned some existent projects under the same spectrum of Geigr. We have also stated that these approaches are not heterogeneous enough to gather all the data that the user might need and predict upon it. We continued with a conceptual study of the approach, starting from data sources to a deeper understanding of server-side aspects of Geigr. The experimental study and the analysis of the results showed the feasibility of our solution as well as its performance with respect to different standard metrics namely, for example, MAPE, MSE, MAE, and the resolution time.

Our future work will therefore be structured around three directions. The first direction is to conduct a more in-depth comparative study between the main approaches studied in the literature in order to give more amplification to academics and practitioners on how to manage PA from text analysis, in social networks. The second direction consists in Predicting resolution time: we have explored both text-based and meta-information-based machine learning algorithms. After validating both of them, we have noticed that the first approach outperforms the second and thus we decided to implement the text-based model. As a third direction, we plan to forecast velocity, after a thorough data analysis, we have concluded that the amount of work accomplished by the team is an autoregression task and that velocity is a stationary variable that has a determined frequency. We have experimented both Holt-winters and autoregression models and implemented Holt-winters into Geigr for accuracy reasons.

References

[1] Bharat Singh Patel, Atul Kumar Tiwari, Manish Kumar, Cherian Samuel and Goutam Sutar. Analysis of agile supply chain enablers for an indian manufacturing organisation.International Journal of Agile Systems and Management, 13(1):48–59, 2020.

[2] Ton Tran, Wen Cheng Liu, Bernard K. Chen, Andrey Molotnikov, Xinhua Wu, and Jerome Pun. Ct-based agile 3d printing system for pre-operative planning of orthopedic surgeries. International Journal of Agile Systems and Management, 13(1):1–27, 2020.

[3] Christian Bettinger, Georg Rock, and Anna Schmitt. Glencoe: a tool and a methodology to manage variability within the product development process. International Journal of Agile Systems and Management, 212(4):332–353, 2019.

[4] J. Tavčar, J. Benedičič, and R. Žavbi. Knowledge management support in the engineering change process in small and medium-sized companies. International Journal of Agile Systems and Management, 12(4):354–33381, 2019.

[5] Silvia Ceccacci, Marco Matteucci, Margherita Peruzzini, and Maura Mengoni. Multipath methodology to promote ergonomics, safety and efficiency in agile factories.International Journal of Agile Systems and Management, 12(4):407–436, 2019.

[6] Oleg Pursky, Dmytro Mazoha, "Architecture Model of Integrated Web-based E-trading Business Process Management System", International Journal of Information Engineering and Electronic Business, Vol.10, No.2, pp. 1-8, 2018.

[7] Iryna Zavuschak, Yevhen Burov, "The Context of Operations as the basis for the Construction of Ontologies of Employment Processes", International Journal of Modern Education and Computer Science, Vol.9, No.11, pp. 13-24, 2017.

[8] Asli Sencer, Meltem Ozturan, Hande Kimiloglu,"A Web 2.0 Supported Business Process Management Environment for Collaborative Research", International Journal of Information Engineering and Electronic Business, vol.7, no.1, pp.8-17, 2015.

[9] Alok Khatri, D. Garg, and G.S. Dangayach. A comparative analysis of factor analysis model for pinpointing agile developers.International Journal of Agile Systems and Management, 12(2):91–107, 2019.

[10] Alf Abuhajleh. Create dashboards, 2020.

[11] S. B Kotsiantis, I.D. Zaharakis, and P.E. Pintelas. Machine learning: a review of classification and combining techniques.Artificial Intelligence Review, 26(3):159–190, 2006.

[12] Xuegong Zhang.Using class-center vectors to build support vector machines. Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop, Cat. (No.98TH8468):3–11, 2019.

[13] Leo Breiman. Random forests.Machine Learning, 45:5–32, 2001.1936.

[14] J. R. Quinlan. C4.5: programs for machine learning. CA, USA: organ Kaufmann Publishers Inc.,2616:235–240, 1993.

[15] Mohamed Nazih Omri and NourEddine Chouigui. Linguistic variables definition by membership function and measure of similarity.Proceedings of the 14th International Conference on Systems Science, 14(2):264–14273, 2001.

[16] Mohamed Nazih Omri and NourEddine Chouigui. Measure of similarity between fuzzy concepts for identification of fuzzy user’s requests in fuzzy semantic networks.International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,

9(6):743–748, 2001.
[17] Mohamed Nazih Omri. Possibilistic pertinence feedback and semantic networks for goal's extraction.Asian Journal of Information Technology, 3(4):258–265, 2004.
[18] A. Poulston, Z. Waseem, and M. Stevenson. Using tf-idf n-gram and word embedding cluster ensembles for author profiling. ICLEF 2017 Working Notes, 1866(urn: nbn:de:0074-1866-8), 2017.
[19] Nils Schaetti. Unine at clef 2017: Tf-idf and deep learning for author profiling. CLEF 2017 Working Notes, 281866(urn: nbn:de:0074-1866-8), 2017.
[20] Yasunao Takano, Yusuke Iijima, Kou Kobayashi, Hiroshi Sakuta, Hiroki Sakaji, Masaki Kohana, and Akio Kobayashi. Improving document similarity calculation using cosine-similarity graphs.International Conference on Advanced Information Networking and Applications, 926(Springer):512–522, 2019.
[21] M. Lailil and B. Baharum. Document clustering using concept space and cosine similarity measurement.11th International Conference in Computer Technology and Development, pages 58–62, 2009.
[22] Monther Sendi, Mohamed Nazih Omri, and Mourade ElAbed. Possibilistic interest discovery from uncertain information in social networks. Intelligent Data Analysis, 21(6):1425–1442, 2017.

Authors' Profiles

Mohamed Nazih Omri received his Ph.D. in Computer Science from University of Jussieu, Paris, France, in 1994. He is a professor in computer science at the University of Sousse, Tunisia. From January 2011, he is a member of MARS (Modeling of Automated Reasoning Systems) Research Laboratory. His group conducts research on Information Retrieval, Data Base, Knowledge Base, and Web Services. He supervised more than 20 Ph.D. and Msc students in different fields of computer science. He is a reviewer of many international journals such as Information Fusion journal, Psihologija Journal, and many International Conferences such as AMIA, ICNC-FSKD, AMAI, etc.

Wafa Mribah did her higher studies at the National Engineering School of Sousse in Tunisia where she obtained her engineering degree in Computer Science option Applied Computer Science. Currently she is an engineer in Product Manager at LeasePlan in Netherlands.

How to cite this paper: Mohamed Nazih Omri, Wafa Mribah, "Towards an Intelligent Machine Learning-based Business Approach", International Journal of Intelligent Systems and Applications(IJISA), Vol.14, No.1, pp.1-23, 2022. DOI: 10.5815/ijisa.2022.01.01