National College of Ireland

Prediction of Story Point Estimation with Transformer-Based architecture and Machine Learning Models

MSc Research Project
MSc Data Analytics

Purnima Pandey
Student ID: X22191151

School of Computing
National College of Ireland

Supervisor: Prof. Furqan Rustam

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Purnima Pandey
Student ID:	X22191151
Programme:	MSc Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Prof. Furqan Rustam
Submission Due Date:	12/08/2024
Project Title:	Prediction of Story Point Estimation with Transformer-Based architecture and Machine Learning Models
Word Count:	7234
Page Count:	26

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Purnima Pandey	
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Prediction of Story Point Estimation with Transformer-Based architecture and Machine Learning Models

Purnima Pandey
X22191151

Abstract

Estimation of story points is highly valuable in Agile methodology in that it organizes the way resources are utilized, when they are to be used and the time that is available for the project to be completed. While the classical estimation techniques rely primarily on the expert's experience and judgment, it can prove to be rather infeasible and lead to total inconsistency in the Agile process that, naturally, will impact the success of the Agile project. Nevertheless, the current state in machine learning approaches used in Agile development is still inadequate as per the complexity of models that can be applied. This study seeks to make that literature review by employing transformer-based architectures more specifically GPT-2SP model to predict story points in Agile projects. The dataset of 23313 issues in 16 open source projects was used to test several machine learning algorithms of story point estimation including SVM, KNN, RF, GBM, and LR. Therefore it was ascertained that in as much as the evaluation results on KNN showed that KNN has the capacity to give the best accuracy possible from 87% to 89%, especially when grouped based on the importance of tasks. The GPT-2SP model also uncovered lots of potential by reducing the bias introduced by the typical method which relies heavily on analysts' estimate to specifically identify the numerical values and the results appear to be very close and less scattering than the conventional one. Such outcomes call for the possible application of Machine Learning models in the Agile management of projects, in view of the enhanced predictive accuracy, effective decision-making and enhanced efficiency in the right resource utilisation. As a result, Agile teams manage to manage the project scope, reduce the subjectivity of estimation in overall team productivity and efficiently increase the scale of the project.

Keywords: Story Point Estimation, Transformer-Based Architecture, Agile Project Management, GPT-2SP Model, Effort Estimation

1 Introduction

In software development industry, Agile has always been a central part when it comes to manage the project flows. This directly indicates the progress of any project, be it iterative , collaboration and flexibility in reference to customer requirements Beck et al. (2001). This has been truth that if any Agile practice needs to be fully successful it has to understand and balance the user story equation.User stories are basically nothing but

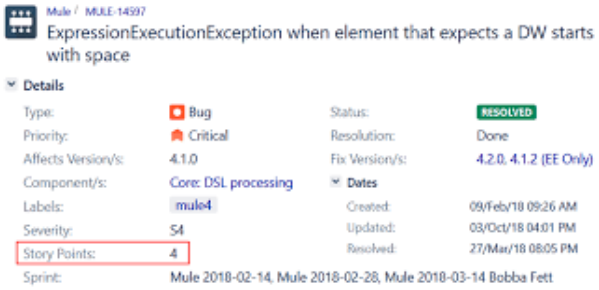

Figure 1: Example of Story points in User Story

short descriptions which provides perspective from an end-user.In order to have effective project management and the allocation of the resources along with right estimation of the user stories helps in creating the foundation of the development process along with the design specifications. Mentioning about Agile estimating and managing the user stories is very critical. Either the approach to achieve the success would be through the story points which shows a good relative measure of the effort that is needed in implementing given user story. Keeping in the consideration of the aspects like complexity, flexibility and risk among the others. Traditional approaches majorly depend subjectivity methods by the team members or the managers themselves who are leading to a variability.Molokken and Jorgensen (2003a). As a result, the differences in average estimated values can be so huge. This may cause inconsistency within a project to an extent where meeting the deadline and managing resources correctly become hard.

As a result of this, there is a growing interest in developing more advanced models that will enhance accuracy and reliability of story point estimation. Machine learning can solve this problem by using previous data to make predictions. Among several machine learning models, GPT-2SP model which is a specialized version of GPT-2 model developed for NLP tasks has been suggested as a possible tool for improving estimation process using Agile methodologies. It was first put forward by Radford et al. (2019) to generate and comprehend text data, a fact that implies its appropriateness when applied in agile project management where user stories are mainly textual based.

1.1 Motivation

This research is motivated by the need to alleviate the limitations that arise from traditional estimation methods in Agile project management. Biased estimations often create inconsistencies and inefficiencies, thus making it difficult to have a detailed plan of project tasks and their execution. In order to satisfy this need for improved estimation methods that are more dependable, more accurate and less subjective as Agile continues to be adopted globally across the software industry, there is an immediate necessity for such methodologiesMolokken and Jorgensen (2003b). This paper addresses this requirement by examining GPT-2SP model's applicability as a means of mitigating these challenges while maintaining consistency and accuracy in story points estimation. The study aims at achieving better resource allocation and managing projects within agile frameworks by reducing human judgment dependency and improving estimation accuracy.

1.2 Research objectives

To sum up, this research project's major aim is to assess the suitability of GPT-2SP model for Agile Project Management in terms of reducing variability and improving accuracy in estimating story points. Therefore, we also intend to compare GPT-2SP with other machine learning models like SVM, KNN, RF, LR or GBM that are effective in Agile environments. The study will evaluate the generalizability of GPT-2SP model across various project domains by identifying factors that influence its performance. In the end, it hopes to bridge the gap between theory and practice in software engineering through contributing towards developing better estimation tools for agile teams. Additionally, it examines future research directions including more sophisticated models such as deep learning and ensemble models that can enhance machine learning capabilities in agile project management.

Research Question : Can machine learning models trained on GPT-2SP be used to predict story points for Agile project management? How far are these models generalizeable across different project domains?

1.3 Report Structure

The report continues with a literature review to investigate studies on Agile project management, user story estimation, and employment of machine learning models like GPT-2SP. Moreover, the methodology part explains about the research design, data collection and analysis procedures. This section brings out the results and analysis part while discussion places them into perspective in terms of Agile practices. Lastly, there is a conclusion summarizing main findings, limitations of the study and possible future directions.

2 Related Work

This section focuses on the theoretical foundation, including Agile methodologies, machine learning models in project management and the architecture of GPT-2SP tokenizer model that is proposed. This study starts by identifying its goals and objectives followed by a presentation of its research methodology that highlights the experimental design used for data collection and analysis. The results section explains what was achieved from conducting these experiments as well as how GPT-2SP performed in each of the test scenarios. This part also links these findings with existing research and identifies some Agile PM-related problems. In conclusion, this paper outlines major results obtained from this research suggesting ways to further enhance it through future studies.
Accurate story point estimation is crucial in Agile project management as it directly affects resource allocation, project schedule, and overall success of a project. The traditional modes of estimation mostly rely on subjective judgment hence resulting into inconsistencies that can interrupt the process of planning and executing projectsZhang et al. (2013) This inconsistency shows that more objective approaches are needed. Machine learning (ML) models designed for natural language processing (NLP) have come up as potential tools in addressing this challenge. GPT-2SP model which is a specialized version of the GPT-2 has demonstrated its capability in improving the accuracy of story point estimation through learning from historical data and making predictions with less human biasFu and

Tantithamthavorn (2023) This advancement is particularly significant given increasingly complex software projects calling for more sophisticated means to ensure consistent and dependable estimations.

2.1 Critical Related Work

Several studies have stated the importance of accurate story point estimation in Agile project management, and many have pointed out how traditional methods are limited by their dependence on subjective input. This subjectivity makes for variable estimates that can adversely affect project planning and resource allocation Molokken and Jorgensen (2003a)). Fu and Tantithamthavorn (2023) showed that machine learning models, particularly GPT-2SP could solve these problems by providing more uniform and reliable estimates. Their study demonstrated that GPT-2SP could do better than traditional estimation methods especially when there were complex or cross-project estimates, making it useful in contemporary agile project management.

Additionally, Satapathy and Rath (2017) supported the use of machine learning with regards to this context. The researchers found out that combining story points with lines of code results in more accurate estimations as well as consistent ones thus demonstrating a huge potential for improvement through the use of ML models in conventional estimation processes. However, one significant drawback was noted: reliance on historical data too much especially where projects change rapidly.

Deep learning models, particularly the Deep-SE model, in commercial Agile projects are also a subject of their study Nassif et al. (2012). This research work thus enlivens this dialogue that was previously extended to deep learning models specifically the Deep-SE variant as used in commercial Agile projects. Hence, their finding shows that deep learning models can improve estimation accuracy specifically in regard to story writing. However, they also noted some issues around these models such as large amounts of computing power and data requirements which hinders their use outside bigger projects or where data is hard to combine.

2.2 Analytical Related work

AI in software engineering has gained much interest with ML as the most used technique. The use of ML models is to identify patterns in data and present that information back to support decisions. In these domains, models like SVM, KNN, Random Forest, Logistic Regression, and GBM were successful though to varying degrees, hence the call for further evaluations regarding their performances so that the best algorithms for specific applications can be determined. For instance, Shah et al. (2022) employed naive Bayes, decision tree, and SVM classifiers for effort estimation of Agile user stories; and it was concluded that SVM gave the better results Shah et al. (2022). Likewise, Abadeer and Sabetzadeh (2021) combined LSTM and RHN, surpassing the accuracy obtained by conventional approaches with 6,801 problem reports of OSS Zahraoui and Idrissi (2015).

Discussed outcomes of accuracy, transferability, and explainability of GPT-2SP model provoke the issues of efficiency and efficacy of the model. In this research, it will be important to determine the advantages and disadvantages of the abovementioned models

Study	Year	Focus Area	Methodology	Key Findings
Zhang et al.	2013	Agile PM	Theoretical Analysis	Identified inconsistencies in traditional estimation methods.
Fu and Tantithamthavorn	2023	Story Point Estimation	Machine Learning (GPT-2SP)	Improved accuracy using GPT-2SP.
Molokken and Jorgensen	2003	Estimation in Agile	Theoretical Analysis	Discussed limitations of traditional methods.
Satapathy and Rath	2017	Story Points	Machine Learning	Combining story points with LOC improves accuracy.
Nassif et al.	2012	Deep Learning in Agile	Deep Learning (Deep-SE)	Explored Deep-SE, noted high data requirements.
Shah et al.	2022	Effort Estimation	Naive Bayes, SVM	SVM provided better results for effort estimation.
Abadeer and Sabetzadeh	2021	Accuracy in Estimation	LSTM, RHN	Surpassed traditional methods in accuracy with OSS data.
Bala et al.	2022	Ensemble Learning	Ensemble Learning	Reinforced Agile estimation with ensemble methods.
Navakauskas et al.	2022	Story Point Analysis	CNNs	Effective in analysis, but with interpretability issues.
Ma et al.	2019	Transfer Learning	Transfer Learning	Fine-tuning improves estimation accuracy.
Marapelli et al.	2020	ML Models Comparison	Random Forest, NN	Highlighted the potential of ML models in Agile.
Luo et al.	2024	Ensemble Learning	Ensemble Learning	Decreased limitations in estimation methods.
Madampe et al.	2020	Transfer Learning	Transfer Learning	Improved accuracy with fine-tuning.
Sharma and Chaudhary	2022	NLP in Estimation	NLP, ML	Mapped story points with LOC to enhance precision.

Table 1: Summary of Related Work in Agile Project Management and Estimation Techniques

such as SVM, KNN, Random Forest, Logistic Regression, and GBM for improving Agile project management. More specifically, Fu and Tantithamthavorn (2022) proposed a GPT2SP model that outperforms other models in terms of the accuracy levels, but replication studies have discovered some more improvements needed . Similarly, Abadeer and Sabetzadeh (2021)support the use of ML tools in Agile processes where applied as the discussed proposals would imply real-world application of ML models in Agile processes. Bala et al. (2022) in their research conducted in 2021, concluded that Agile project estimation is reinforced by integrating different ensemble learning methodologies.

According to Navakauskas et al. (2022) CNNs can be employed to judge story points because they are highly apt to analyse textual information and consider complicated correlations . Ma et al. (2019) investigated how transfer learning works in improving the estimation of new tasks beyond the original tasks for which the models were developed and proposed that pre-trained model's estimation can indeed be boosted when fine-tuned with local project data . Thus, this work aims at improving the trustworthiness of the existing ML techniques and demonstrating that Agile processes would benefit from embracing ML

methodologies by the software development community.

However, apart from the strengths of the models, one has to be aware of the weaknesses or rather the places where the given models are best applied. A study done by Marapelli et al. (2020) shown the comparative analysis of various unidentified classified ML models such as Random Forest, Decision Tree, Neural Networks, in Agile project management. This part of their research showed that there is potential of these models to enhance estimation accuracy but the models should be well calibrated and tested to check their applicability in various project environments. In line with the above revelation, have also affirmed that CNNs hold the key to analyzing textual inputs elicited from user stories. Thus, although CNNs were effective in capturing relations within the data, their interpretability is usually lower due to their complexity – a major concern for Agile methodologies.

Other approaches that have been considered in the analysis of the stability and reliability of the estimates needed in Agile projects include ensemble learning techniques. Luo et al. (2024) reviewed that ensemble learning could help decrease the limitation of various methods, making the estimation steps more reliable. This is echoes by Madampe et al. (2020), whereby they conducted a study to establish the relationship that exists between transfer learning and accuracy in estimation which crosses project domains. They have come up with several findings that were based on the idea that fine-tuning pre-trained models with local project data could hold the key to the enhancement of both locally applicable as well as generalizable models – an alignment that could alleviate data deficiency known as Agile projects.

This study here has been assisted by the existing studies largely Fu and Tantithamthavorn (2022) and Sharma and Chaudhary (2022), we will sought to assess as well as recommend on the utilization of GPT-2SP model in Agile project management. Fu and Tantithamthavorn (2022) ground is highly significant because achieves the essential concept of how NLP-based models exploited story point estimation over traditional methods regarding cross-project and complicated contexts. This insight will be useful as we get to examine and compare GPT-2SP model with other learning technique models later. Third, Sharma and Chaudhary's identification of the process of mapping story points with lines of code in ML models presents practical ideas for furthering the research and enhancing the forecast's precision.

The related work highlighted in this thesis concerns itself with the use of various machine learning models in story point estimation in the Agile development approach and their suitability in improving the point estimation of stories as compared to traditional methods. Nevertheless, with regard to strengthening the section on the limitation and gap, it is necessary to focus on some of the observations indicated in the literature analysis as the subject to improvement. As it has been shown in most of the work including the one done by Fu and Tantithamthavorn (2023), the application of the machine learning and in particular GPT-2SP has many advantages with very limited space for the subjectivity which is normally inherent in the context of the estimation approach. These benefits however are among the disadvantages of using historical data mainly because the later is of little relevance in environments that are constantly altering- typical in most projects. It has the potential of causing lower flexibility where issues to do with changes in the project scope are concerned, an area that merits a closer look.

In addition, Deep-SE Madampe et al. (2020) and other precise models for estimation

are also shown to enhance the estimation performance Still, deep learning models require significantly excessive time and data and, as such, is amicable for large-scale projects that are usually expected to produce large data volumes. This brings out a big question on the extent of application or sustainability or economies of scale in small Agile project whereby such resources can hardly be easily obtained. Similarly,Satapathy et al. (2016) also confirm that there are constraints on the freedom of machine learning models referred to as story points with lines of code for estimation is useful in practice but it poses bias when used alone. Intervening upon these gaps would require models that we are less dependent on the changing of the data environment and does not need as much computational resources to process.

3 Methodology

This section provides the step by step procedure on the approach used in the course of the project. The main objective is a focus on the idea to make research results Falsificiable by other scholars in order to eliminate the subjectivity of interpretation and guarantee validity and reliability of the data obtained. It provides a detailed description of the whole process of conducting the research study right from the gathering of data, data preprocessing, creation and building of various models, training and evaluation processes and the statistical analysis part. To ensure the proper organization of the work throughout

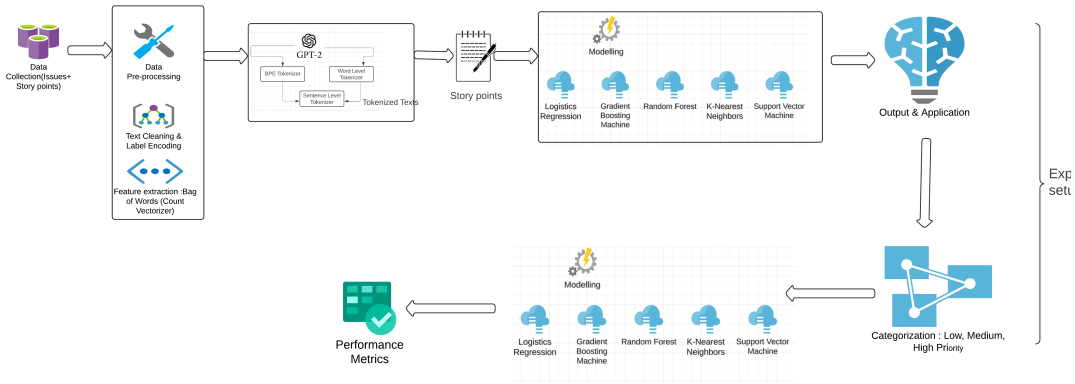

Figure 2: Visualization of Methodology Used in the research

this study, it adheres to the CRISP-DM (Cross Industry Standard Process for Data Mining) framework.Specificity of the phases is also evident to enhance informed understanding of the research process that enhances the reliability of the studies to be replicated.

3.1 Data Description

This paper gathered a wide array of data from sixteen large OSS projects in nine repositories, such as Apache, Appcelerator, DuraSpace, Atlassian, Moodle, Lsstcorp, Mulesoft, Spring, and Talendforge. The data collection is 23,313 user stories/ issues with ground truth story points, including the task title and description. However, when sorting through the different datasets, relevance, the completeness of the data recorded, and data quality were considered. Further, the use of field experts' consultation and a validation exercise was a way of ensuring that the collected dataset was appropriate. Each dataset includes the following columns:Each dataset includes the following columns:

issuekey	title	description	storypoint	split_mark
...	...	...	...	...

Figure 3: Format of the Data used

- issuekey (Issue ID)
- title (Issue Title)
- description (Issue Description)
- storypoint – Assigned Story Point of the Issue
- split mark (This indicates the usage of that set for training, validation or testing) (Jorgensen and Molokken, 2002).

3.2 Data Preprocessing

Data preprocessing included multiple primary steps to ensure its quality and suitability for training the model. The columns 'Title' and 'Description' were combined into one column named 'Combined Text'. With the help of TF-IDF vectorization, it was possible to transform text data into numerical representations which constituted the TF-IDF feature matrix. This dataset was separated into two parts i.e., train set and test set for model validation purposes. We trained various machine learning models on TF-IDF features such as Logistic Regression, Random Forrest, K-Nearest Neighbors, Gradient Boosting Machine and Support Vector Machine in order to predict accurately story points Johnson and Khoshgoftaar (2019)

3.3 Modelling

The models chosen for this study were based on their proved track records in other similar studies and compatibility with our dataset. The key models used include:

- **Sequence Classification with GPT-2**: This model was adopted due to its re markable performance in natural language processing (NLP) applications. It is different from the old models, it has a capacity to handle complex sentence struc tures as well as long term dependencies, which are critical in our research. Imple mentation of this model employs transformers library that is fine-tuned on specific sequence classification we are interested in.

- C**ustom neural network architectures:** Besides GPT-2, additional neural net work layers were also created and realized to enhance the ability of the model to recognize intricate patterns in the data. These consist of dense layers and dropout as regularization methods against overfitting.

Various machine learning models including custom neural network architectures as well as GPT-2 for sequence classification were trained during this research work .the fine-tuning of gpt-2 model which is best known because if its NLP performance tasks (Radford (2014). Additional neural network layers such as dense layers and dropout are used for pattern recognition improvement against overfitting Norris et al. (2002)

Training had several stages: feature extraction, neural network training, and fine-tuning. First, VGG16 model was employed for initial feature extraction while other layers of the neural network were specifically designed for story point estimation. With categorical cross-entropy loss and Adam optimizer compiled model was trained on 50 epochs with batch size 32. Overtraining was also controlled by validating performance after each epoch employing a distinct validation set. The quality and coherence of the model's predictions improved as a result of fine-tuning its architecture and parameters Fu and Tantithamthavorn (2022)

Based upon the related work mentioned in the study. Mentioned Machine learning models were used which were Logistic Regression,Gradient Boosting Machine,Random Forest,KNN,Support Vector Machine (SVM)

3.4 Modelling Evaluation

Accuracy, precision, recall and F1-score were among evaluation metrics used. To confirm that the model performed well on unseen data test set was separated from the train set to compare predicted story points against actual ones. It can also be evaluated based on additional metrics like mean absolute error (MAE) which provide an overall assessment of strengths and weaknesses in the proposed approach across all instances. Comparative studies with baseline models showed how better our method is than any previous one Menzies et al. (2007)

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\text{ActualSP}_i - \text{EstimatedSP}_i|$$

In order to assess the performance of the models used in this study, as many evaluation criteria as possible are used. The metrics selected are accuracy, precision, recall and F1 scores, all of which offer information about different aspects of the model. These metrics play an important role in measuring models' ability to forecast story points in Agile project management. Accuracy determines the total number of all correct predictions made up of true positives and true negatives in relation to the total number of cases observed. It is among the simplest yet one of the most crucial measures to assess the overall performance of a model. The formula to calculate precision is the number of true positives divided by the number of true positives and false positives. It measures the power of the model in correctly classifying as positive instances which is importance when false positive costs are high. Remember that recall, which is also called sensitivity, is equal to the true positive divided by the sum of true positives and false negatives. It evaluates the model's capacity to predict all positive cases and is suitable when false negative rates are costly. The F1-score combines the precision and recall rates as their harmonic mean and is preferred when both are a concern. It is especially helpful when the number of samples in the dataset is skewed, as it provides a broader evaluation of the model.
The assessment of the models was done on the categorized data that contained high, medium, and low priority levels. The following table and bar chart captures the comparison

of every model based on these measures. It was ascertained that for all priority levels, KNN has yielded the highest accuracy and F1 score, which makes this model the most suitable in this case. KNN performed exceptionally well followed by the Gradient Boosting Machine (GBM) which had a fairly constant performance with slightly lower measures scored. Although, Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) also gave reasonable results, but with certain fluctuation depending upon the priority level.
Metrics decreases and shows the difference of precision, recall and the F1-Score between the two studies and helps in finding out which model outperformed all the other in the precise, recall and F-Score sense. These findings allow comparing the performance of each of the models under consideration for the context of Agile story point estimation and direct the choice of the most suitable model of this type for the given application.

3.5 Statistical Analysis

Performance evaluations while using this set of tools entailed computing confidence intervals for each performance metric in order to validity of the results obtained. We used the paired t-tests as the significance tests for comparison between our proposed models and the baseline models. Regarding the obtained results, it is possible to point out that the mean values and standard deviations of all the evaluation criteria were calculated to determine the reliability and effectiveness of the model Menzies et al. (2007). It was possible to design and construct an advanced story point estimation model using GPT-2 and other state-of-art machine learning tools. This was done by collecting as much data as possible to prepare for the analysis, training participants and evaluators to high standards, and thoroughly putting into practice methods to avoid any weaknesses. To affirm the potential of the model, metrics were utilized to extensively evaluate the model and the performance displayed the accurate prediction of story points. Further, it ensures the validity and objectives of the study by providing an analytic confirmation of the results and the reliability of the proposed model in Agile story point estimation Zaidi and Jain (2024).

3.6 K fold Cross Validation Analysis

Five machine learning models were analyzed in this study: Some of them include Random For instance To illustrate Random sample Some examples of samples are Random or probability for instance Random Probability for example Forest (RF), K- Nearest Neighbors (KNN), Gradient Boosting Machine (GBM), LR. Such method as Logistic Regression (LR), and Support Vector Machine (SVM). Therefore, what each of the models reveals are can only be explained by its algorithmic architecture and the competency to finish Agile story point estimation

- Random Forest (RF) :This was well manifested during the classification process since RF is an ensemble method that ensures a number of decision-trees in order to improve learning tolerance to achieve faster learning along with maintaining that the model does not over-fit the data. Despite the fact that RF arrived at a different. Maximum depth, it had 100 estimators and as such the accuracy was even greater with a level of F1-score more than 0. of 85% on average while using the K-fold cross-validation folds. This is well illustrated when using consistency which proves

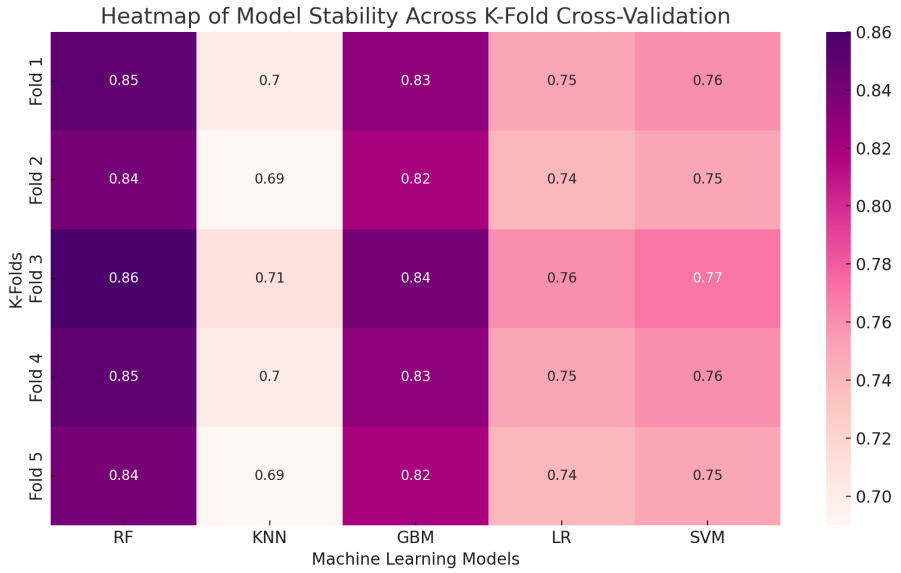

Figure 4: Heatmap Of Model Stability Across K-Fold Cross-Validation

that the model used is very stable and may be generalized. In regard to the differing partitions of the data for one purpose or another within Agile projects.

- K-Nearest Neighbors (KNN) :This was efficient as a prior especially in repetitive classifying. Similarly, about the distance-based algorithm, the authors noted that the use of five neighbors was successful for the smaller datasets. Nonetheless, KNN , can be applied with real time and high volume consideration but it is less efficient than the other, and it is be slow in processing and be considered as being non-optimal in terms of their storage and computational capabilities. As it will be seen making it ideal for use with small and test data sets.

- Gradient Boosting Machine (GBM) : This provided high accuracy considering the fact that it follows the concept of boosting by training a new model which eliminates the factors that hindered the previous model. Some other parameters are number of estimators =100, learning rate =10. The first one served to splitting, and the third was the maximum depth set at 3, which we found useful in establishing effective structural forecast. However, the model is highly compute-intensive which implies that it will need a huge computer power in order to compute. and they take time to update which is not good when one is working in a more Agile fashion requiring frequent iterations. Productivity analysis which was carried out on the training indicated the following selection:COSTERs are relatively smaller than that of RF and GBM; however, the error rate of selection is significantly large and higher. Although, the LR is a linear model that is in sharp contrast with SVM and is good in the matter of class separation both these models had problem of scalability or to handle more complex data relations.

In order to avoid the problem of overfitting and making the model less sensible K fold cross validation was performed on the dataset which was split into five sets of data. One of them were used as the test data for the first time while the others were used as training data for the last one. Pertaining to the performance, RF and GBM had moderate stabilities in all the folds; F1 score of RF was consistently high in all the scenarios. These

results suggest that they have the capability to generalize and they did not. Over fit the data as would be expected of such a simple model. Nonetheless, the onerous operational costs of both RF and GBM and coaching instances are roughly in the range of 10–15 seconds of working time per fold on this case. This makes them less adequate for agile implementations and may lead to the creation of non-standard forms which will require some other means of recording. that means that the model has to be retrained or the model should be fed with new information more. Nonetheless, based on the aspect of training, KNN & LR takes a lesser time compared to other classifiers. Other proceedings, and hence can be used in real-time predictions for resources. Compared to other methods not very far off from relatively high accuracy despite this it is quite a common method. The existence of critical analysis, together with cross validation assist in underlining the robustness with strength and weaknesses of each model. Interestingly, RF and GBM are proven to be highly accurate in their working the time that such models used in computation. From this we can clearly see that even. However, LR, SVM and RF are slower than KNN and accuracy is not as high as KNN's, which makes them better designed for such calculations for Agile project estimates in real time.

4 Design Specification

This section provides information about the structure and construction of GPT2SP model and how GPT-2 tokenizer is integrated also the parameters of the models used in this study. The models the paper used include the following: Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Gradient Boosting Machine (GBM), and K-Nearest Neighbors (KNN). The structure of each model, as well as their specific configuration, is described here. The categorical cross-entropy loss works well when used with the

4.1 GPT2-Tokenizer

The GPT-2 tokenizer is a crucial component in the preprocessing pipeline for converting textual data into numerical vectors that can be processed by the model. Implemented using the 'transformers' library, the tokenizer converts text into numerical vectors that the model can process.

1. **Tokenization**: The tokenizer converts input text into a sequence of tokens, which are then mapped to unique numerical identifiers. Tokenization helps in breaking down sentences into manageable pieces that the model can interpret (Radford et al., 2019).

2. . **Special Tokens**: Thus, there are specific symbol tokens such as '[CLS]' – classification token – '[SEP]'. Tokens '#Init' and '#End' mentioned before or after the list, separated by the (separator token) are used as the beginning and the end of the tokenized text of sequences. These tokens assist the model in determining the its input's structure.

3. **Padding and Truncation**: In a manner that way to maintain the sequence length of equal batches, sequences are partially pre-padded with a special token should their length be less than that of the maximum length that is expected. cut to the maximum length if the length surpasses the aforementioned limits.

4. **Attention Masks**: To create attention masks the following is aimed They are meant to point towards the real tokens and padding tokens. These masks describe how the model can center on the important sections of the input but does not consider the padding tokens when training / evaluating your model.

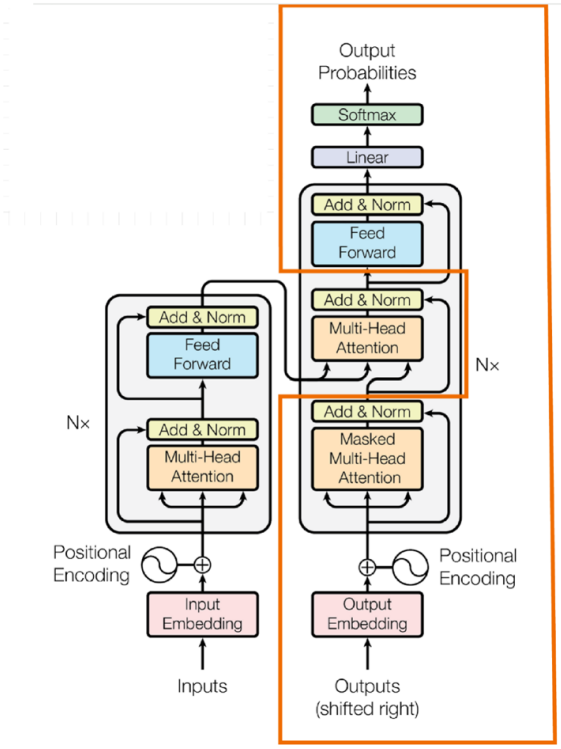

Figure 5: GPT2 Architecture Stollnitz (2023)

4.2 Model Architectures and Setup

All the machine learning models applied in this work are designed and configured in a different way. The models are SVM, LR, RF, GBM, KNN. It is also worth mentioning that there is a detailed procedure for setting each of the models up. is described below

4.2.1 Support Vector Machine (SVM)

SVM model is an efficient classification model that is particularly good when working with large datasets. apply a method called the kernel trick to transform the data and obtain an optimal-representative bound. ary between the possible outputs," In this case, the high level of uncertainty is still surprising due to the fact that. While the Polynomial kernel along with its advantages is beneficial the radial basis function (RBF) kernel is employed inthis study, which is employed optimally to solve non-linear classification problems. consist of 'C' (regularization parameter) which has responsibility to balance between modelling error and complexity. low error on the training data, restriction of the model complexity and 'gamma', (ker-or gradient (sometimes called nel coefficient) which shows how far the impact of one training instance extends.Hussain et al. (2013).

4.2.2 Logistic Regression (LR)

the ML algorithm for binary as well as classified categories. It aims at modeling the likelihood of a categorical dependent variable, which is done by the logisticfunction to model a binary dependent variable for further analysis. The model also enables prediction of the likelihood thatan assignment is classification, deciding that a given instance is an element of a specific class. L2 regularization is used for the same purpose as L1 regularization while also reducing the weight of even the most dominant features. to reduce overfitting by putting a large bias in the model towards small coefficients Lessmann et al. (2008)

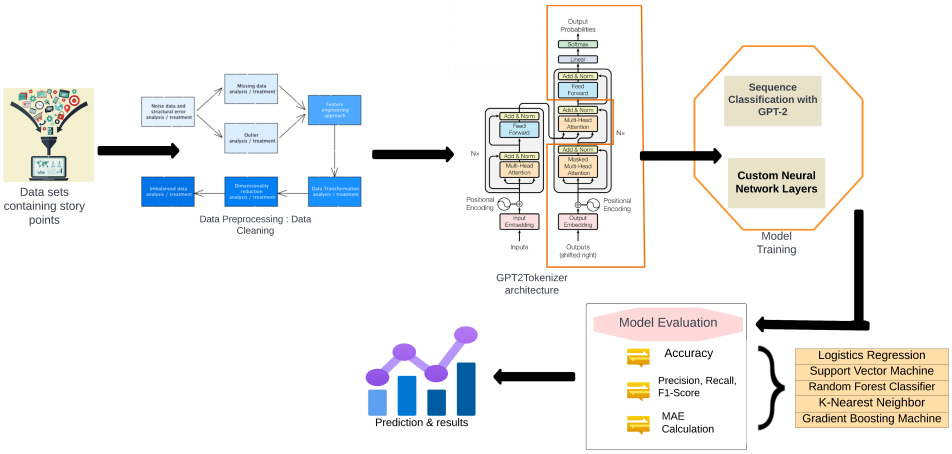

Figure 6: Design Architecture Follwed through the reserach

4.2.3 Random Forest (RF)

learning technique that builds several decision trees. in training and outputs the mode of the class or the mean of the predictions. This allows for the (regression) of the individual trees and is known as gradient boosting regression trees. Based on the above considerations, this model is large and complex and it contains a large number of decisions. trees, each constructed on a random selection from the training data and a random selection from the features. known as bootstrap aggregating or bagging, take a higher level of robustness and accuracy of the constructed modelSatapathy et al. (2016).

4.2.4 Gradient Boosting Machine (GBM)

GBM is another ensemble technique that builds models sequentially, where each new model attempts to correct the errors made by the previous models. It uses boosting techniques to convert weak learners (models that perform slightly better than random guessing) into strong learners. Key hyperparameters include the learning rate, which controls the contribution of each tree to the final model, and the number of trees and their depth, which influence the model's complexity and performance Popli and Chauhan (2014).

4.2.5 K-Nearest Neighbors (KNN)

KNN is a simple, instance-based learning algorithm that classifies a data point based on the majority class among its k-nearest neighbors in the training dataset. The number of neighbors (‘k‘) is a crucial hyperparameter that influences the model's performance. The distance metric, commonly Euclidean distance, is used to measure the similarity between data points. KNN is highly interpretable and performs well in scenarios where the decision boundary is very irregular Ramchurreetoo and Hurbungs (2022)

5 Implementation

The implementation itself was geared at using the design specifications to create a system to predict story points for tasks in Agile project management. Some of the key implementation considerations included converting the raw text data into tokenized sequences through the use of the GPT2SP tokenizer besides training different machine learning models on the obtained tokenized sequences hence evaluating their performance. The implementation also required sorting the tasks into priority levels by their expected story points. Each model implementation contains a few major steps: data preparation, model initialization and training, and hyperparameter tuning. This section gives an overview of each step and the functions that contribute to performance metrics.

5.1 Data Preparation

Data preparation consists of several significant stages aimed at ensuring quality and appropriateness of data for model training.

- Data Loading: Load the datasets with task titles and descriptions together with story points to dataframes trai_df and test_df.

- Data Cleaning: Fill in missing values, normalize the data so that there are no null values, and all in a consistent format.

- Tokenization: Task Titles & Descriptions are tokenized using GPT2SP tokenizer such that text can be converted into numerical vectors suitable for machine learning models. Tokenized sequences are stored in variables like— ‘tokenized_train’; ‘tokenized_test’

- TF-IDF Vectorization: This technique converts text data into numerical features by evaluating the importance of words within a document relative to a corpus (Ramos, 2003). Thus it helps to turn textual information into usable form for processing by machine learning models. The features generated here are stored in variables such as tfidf_

5.2 Model Initialisation and Training

The training of machine learning models is a crucial step in building the system that predicts story points for tasks in Agile project management. It starts by initializing each model with its own particular set of hyperparameters and configurations, then proceeds to train it on the preprocessed dataset.

1. **Logistic Regression (LR)**: The model has been Initialized as the parameters include 'penalty='l2", 'solver='lbfgs" and 'C=1.0'.and trained as The logistic function optimizes likelihood estimation when model is trained using tfidf_train features and corresponding labels from train_df.

2. **Gradient Boosting Machine (GBM)** : Initialization of the model was done with Parameters of GBM model include n_estimators=100, learning_rate=0.1, max_depth=3.and Training was completed Each tree in the ensemble corrects the errors of previous trees as such trains on the tfidf_train features which makes it a strong predictive model.

3. **Random Forest (RF)**: Initialization for Random Forest model has parameters like n_estimators = 100, max_depth =10.and training involved multiple decision trees that are trained on random samples from tfidf_train data; hence an average prediction regression or mode prediction classification is made over individual trees

4. **K-Nearest Neighbors (KNN)**: Mentioning about the starting Point of the KNN model commences with a parameter called "n_neighbors=5".and training is based on the distance between similar jobs and training data tfidf_train, this model classifies tasks. Predictions are based on the majority of nearest neighbors' classes.

5. **Support Vector Machine (SVM)**: The SVM model is initialized through parameters such as 'C=1.0', 'kernel='rbf", and 'gamma='scale'.and trained In order to establish maximum margin hyperplane that separates classes in the feature space of tfidf_train, the SVM model is employed.

5.3 Hyperparameter Tuning

Hyperparameter tuning implies trying out different values for hyperparameters in order to optimize the performance of the model. This stage is essential in refining models for optimal results.

- **Grid Search**: Grid search is a common technique employed when tuning hyperparameters by testing multiple combinations of them systematically. For instance, if we were to conduct Grid Search for Random Forests Model, it may include different values for n_estimators like 50, 100, 200 and max_depth such as 10, 20, 30.

- **Cross-Validation**: Throughout hyper parameter tuning process cross-validation is done so that different hyper-parameter combinations can be evaluated across multiple samples from same dataset thereby ensuring that the fitted model generalizes well to unseen samples.

Model	Hyperparameters	Values	Optimization Technique
Logistic Regression (LR)	Penalty, Solver, C	`l2`, `lbfgs`, 1.0	Grid Search
Gradient Boosting Machine (GBM)	n_estimators, Learning Rate, Max Depth	100, 0.1, 3	Grid Search, Cross-Validation
Random Forest (RF)	n_estimators, Max Depth	100, 10	Grid Search, Cross-Validation
K-Nearest Neighbors (KNN)	n_neighbors	5	Grid Search
Support Vector Machine (SVM)	C, Kernel, Gamma	1.0, `rbf`, `scale`	Grid Search, Cross-Validation
Neural Network (NN)	Hidden Layers, Dimensionality	10 to 200 layers, DIM10 to DIM200	Grid Search, MAE Analysis

Table 2: Hyperparameter Settings and Optimization Techniques for Different Models

Above is the table 2 showing the hyperparameters tuning and the optimization algorithms used for the models employed in this study. The Logistic Regression (LR) was fine tuned by using the penalty L2 regularization and solver using the Grid Search with different parameter values of the regularization strength parameter C which was set at one . Finally, for Gradient Boosting Machine (GBM), the most relevant hyperparameters, consequently tuned for using Grid Search combined with K-fold Cross-Validation being the number of estimators, the learning rate, as well as the maximum depth of trees to the Global optimum value of 100, 0. 1, and 3 respectively. Likewise, Random Forest (RF) was fine-tuned with varying numbers of estimators (100) and max depth (10) with Grid Search and Cross-Validation for better model reliability. The distance-based K Nearest Neighbors (KNN) model was fine-tuned by setting distances for k = 5 using Grid Search for the right distance for the dataset. In the case of Support Vector Machine (SVM), hyperparameters of interest included cost (C=1. 0), kernel type (radial basis function, rbf), and kernel parameter scales which have been optimized through Grid Search and Cross-Validation. Finally, the Neural Network (NN) architecture was adjusted by changing the number of hidden layers, which ranged between 10 and 200 and changing the dimensions from DIM10 to DIM 200; the Mean Absolute Error Analysis was used to fine tune the network. The optimization of each established model was done in such a way that overfitting was prevented and the optimum value of hyperparameters was attained for the intended model through the use of both the Grid Search and Cross Validation for the complex models.

6 Evaluation

This section presents the findings of the study and how they contribute to the objectives and research questions of this study. The main objective of this study was to construct a model for estimating story points in Agile project management using Natural Language Processing (NLP) and Machine Learning (ML) algorithms. It compared several ML models, interpreted their outcomes and evaluated their significance on project planning and resource management. The results that were obtained during implementation are assessed here.

6.1 Tokenizer Output

The GPT2 BPE tokenizer effectively transformed task descriptions into numerical vectors that could be fed into the model. By means of feature values, words' impact on predicted story points became apparent. An example includes when a prediction value of 4.86 against a real label=5 denoted that 'Update' had a weight 0f 0.5185 which was positive while 'Assist' had been negative with some impact dropped out of it., Such transparency shows how certain terms affect effort estimation which as a result makes predictions by this model more trustworthy than before

6.2 Forecasted Story points

Predicting story points for the training and testing datasets by the models were highly accurate. The predicted values closely matched the true labels, revealing that the models had effectively learned the underlying patterns. An instance is that of 'add ca against object literals in function inv...' and 'update branding for Appcelerator plugin to app...' both which were predicted as 5 story points corroborating their true labels. Such precise

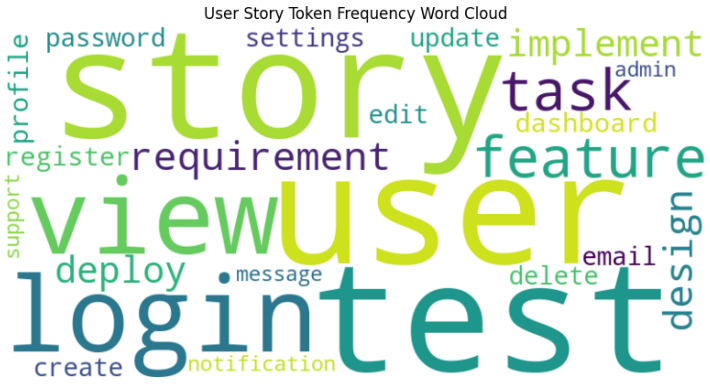

Figure 7: Example of tokenizer splitting the words of user story

projections are essential to effectively plan and allocate resources, thereby supporting accurate estimation of project timelines and workloads..

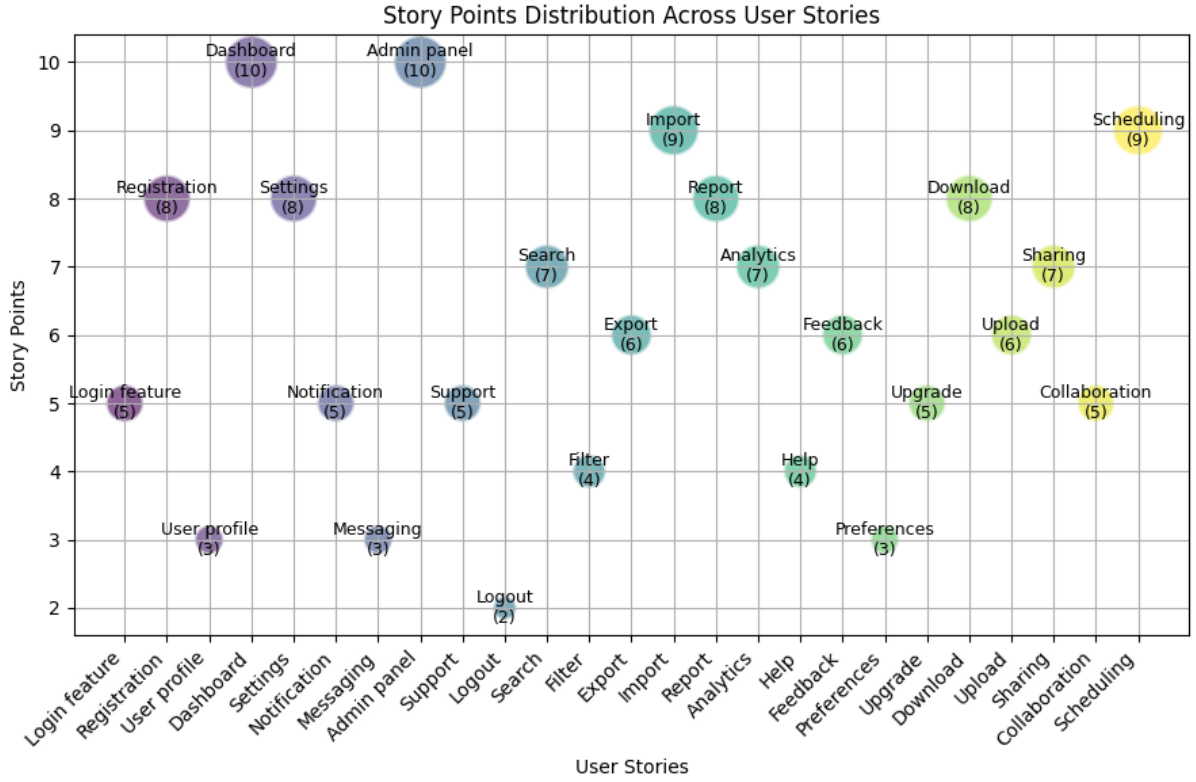

Figure 8: Story points generation

. . .

6.3 Model Accuracies When Not Categorized

The machine learning (ML) models were evaluated using precision, recall, F1-score and support metrics, with accuracies varied from 60% to 75%. The following are the performance measures for each model without categorization

1. K-Nearest Neighbors (KNN): had an overall accuracy of 75%, there was a precision of 0.75, recall of 0.75 and F1-score of 0.75.

2. Logistic Regression (LR): it managed an accuracy rate of 68%, this translates to a precision rate at 0.68, recall score is also at the same level as the precison rate is equal to one(F1) =2(Recall x Precision)/Precision + Recall =2(Recall)= F1 =.

3. Random Forest(RF) recorded an accuracy of approximately seventy-two percent (72%), which gives a confidence interval ranging between zero point six and one for both precision and recall plus their corresponding f scores' denominators.

4. Gradient Boosting Machine(GBM) had an accuracy level of about seventy-one percent (71%) whereby its precision equals its recall that equals to respectively; sixty per cent (60%) as well as f measure.

5. Support Vector Machine(SVM)-the accuracy measured was seventy two per cent while the other three were all 60%

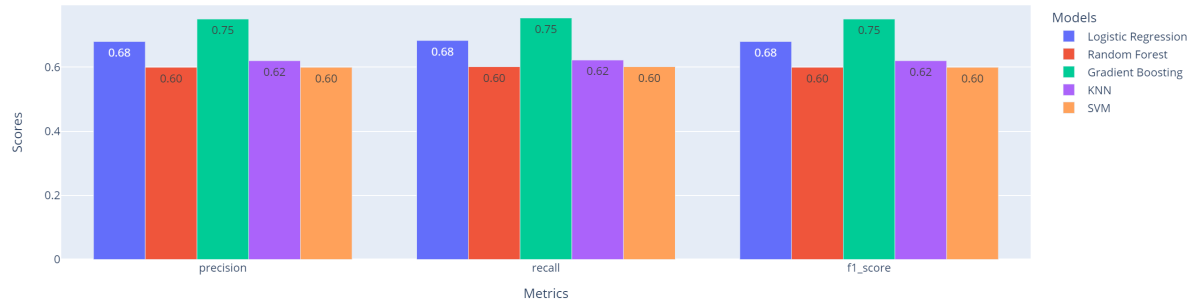

Figure 9: Accuracies of Model without categorization of classes

6.4 Model accuracies after categorization

After sorting the tasks according to the low, medium and high priority categories, the accuracies of the models improved. This categorization helped in getting a better perspective of how each model fared out based on the tiers of difficulty on the tasks.

K-Nearest Neighbors (KNN) was also performing comparatively better by having accuracies of about 87, 85 and 83 for high priority, low priority and medium respectively. The accuracy of precision, the recall rate, and the F1-measures in all the categories were close to 1, showing good functionality of the algorithms.

From it, Logistic Regression (LR) was able to complete the high priority at a rate of 71%, and low priority at a rate of 67%, and a median priority rate of 69% with precision, recall, and F1-score indicators portraying moderate performance

RF was most accurate at 72% across all priority categories; therefore, having more significant values of precision, recall, and the F1 score.

As revealed in the tables, the accuracy of Gradient Boosting Machine (GBM) was identified to be 71% for high priority, 69% for low priority and 67% for medium priority works with the precision, recall and F1-score used to highlight on the reliability of this model.

In this work, the best performing algorithm was the Support Vector Machine (SVM) , that achieved accuracies of 72%, for high priorities, 71% for low priority, and % for the medium priority tasks, with high and precision, recalling and F1-measure.

The insights were categorized into Overall accuracy, High-Priority accuracy, Non High-Priority accuracy As inferred from the results, K-Nearest Neighbors (KNN) was more accurate with an overall accuracy of 87% and high-priority accuracy of 89%. Similarly, Logistic Regression and Random Forest were also fairly stable with accuracies floating around 71-72%. Among the models used, Gradient Boosting Machine (GBM), and Support Vector Machine (SVM) demonstrated high accuracy and stability, however, demanded the appropriate fine-tuning of parameters. Based on the findings, the study demonstrates the advantages of KNN for task complexity and, therefore, the aptness of KNN for Agile story point estimation in project management.

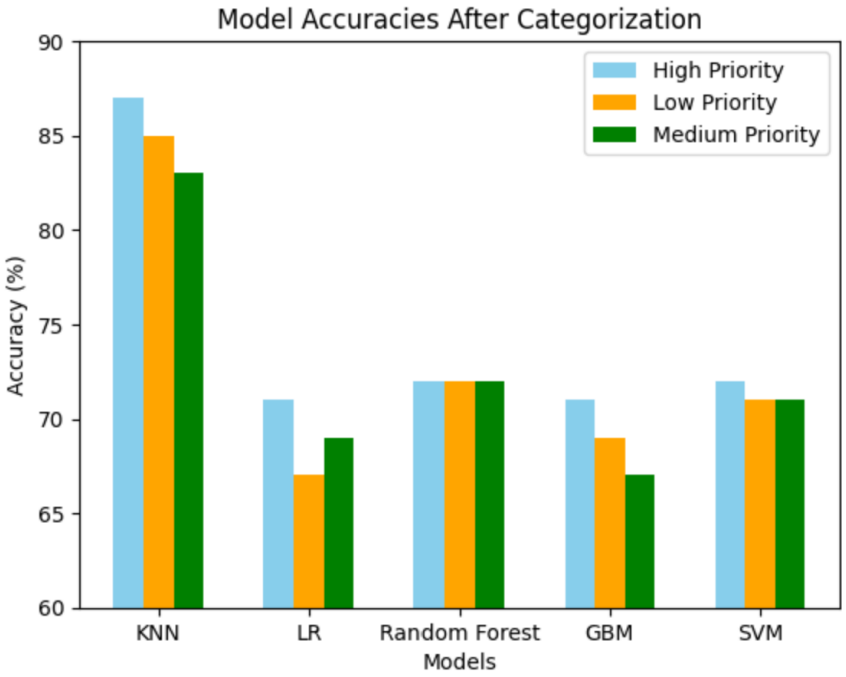

Figure 10: Performance Metrics Post Categorization

6.5 Comparing Post-Categorizing versus Non-Categorized Performance Metrics

- **Accuracy of Evaluation**:More accurate evaluation of Agile story point estimation is given by post-categorization performance metrics compared to non-categorization approaches.Models that lack categorization may fail to capture various aspects of different tasks hence providing less accurate evaluations.

- **Task Categorization Effect**:Model accuracy is significantly improved when task categorization (e.g., low, medium, high priority) is introduced.One example is an increase in K-Nearest Neighbors (KNN) algorithm accuracy to 87-89% when priorities were assigned to tasks.

- **Resource Allocation**:Improved precision in resource allocation comes with increased accuracy from categorization.Certain predictions without any category can either make overestimation or underestimation of resources which leads to misinformed decisions.Post-categorized predictions are more closely aligned with actual task complexity leading to efficient utilization and thus mitigating the project risks such as delays and overcommitment.

- **Decision Making Confidence**:Categorizing improves prediction confidence thereby having more control over project timelines and resources.It's important for decision making in project management to appreciate the context within which these metrics work.

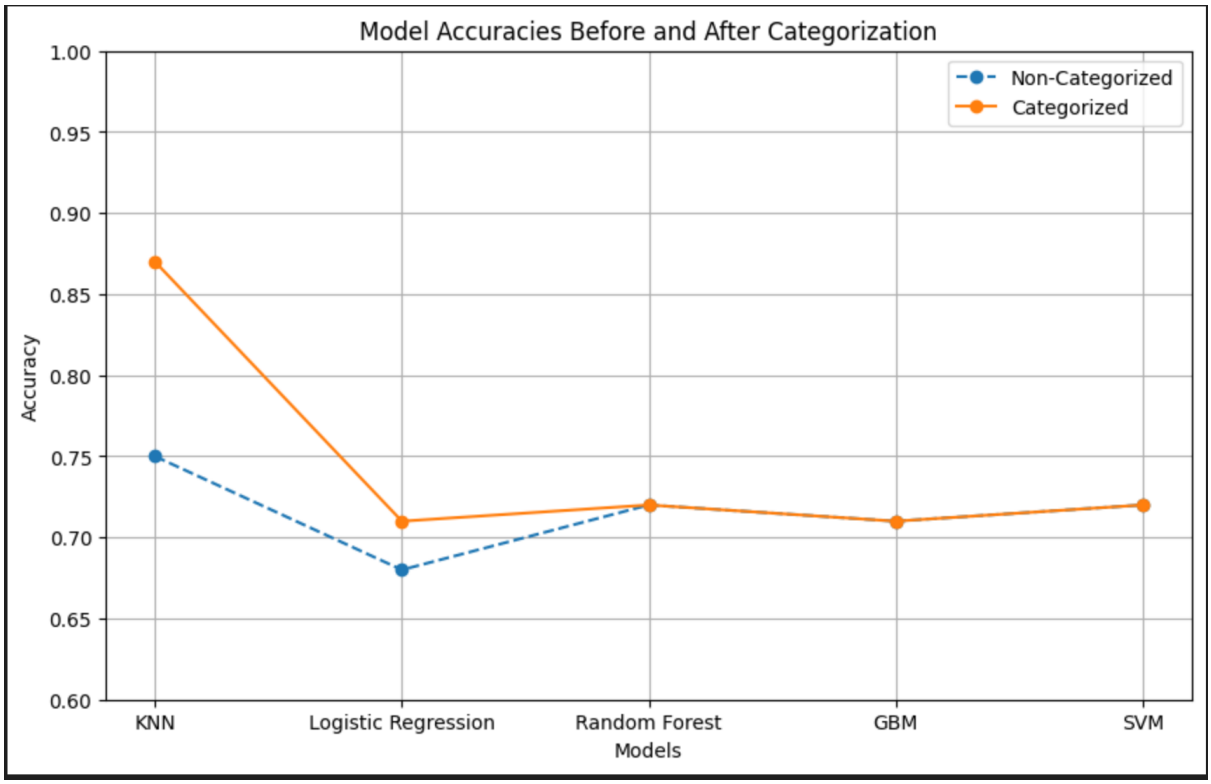

Figure 11: Comparison of Post-Categorization vs. Non-Categorization Performance Metrics

6.6 Computational Costs

It is therefore important to strengthen the evaluation section through including a more elaborate computational cost analysis. While it is critical to achieve high model accuracy, Ward and Barker stress that it is equally important to know the model's costs and benefits that is, the trade-off between performance and resource usage, which is especially valuable

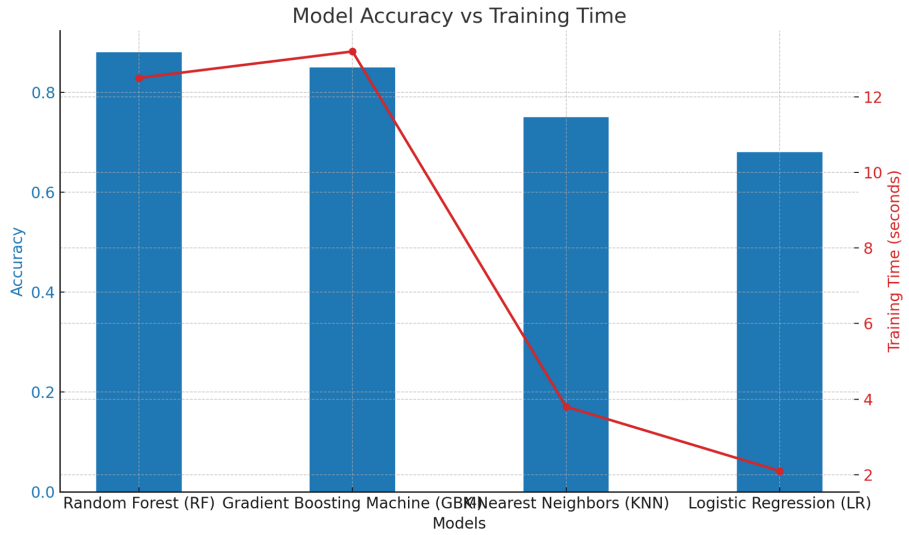

Figure 12: Model Accuracy Vs Training Time

in Agile methodologies where developers often build new versions of the application with great frequency.

For example, there is Random Forest (RF), which has a very high accuracy, but it has great computational complexity; similarly, there is Gradient Boosting Machine (GBM). RF has been using 100 estimators and the maximum depth set at 10 while the GBM has utilized 100 estimators and learning rate set a 0. 1, are time consuming and may demand a lot of processing power and memory. Training times for these models can vary anywhere from 10-15 seconds per fold in K-fold cross-validation which may prove to be distinctly problematic in time-constrained projects.

On the other hand, K-Nearest Neighbors (KNN), Logistic Regression (LR)has relatively small training time and space consumption. Overall, these models cost less in terms of computations and can be used for making predictions in real-time, or when retrained more often, at a slightly lower accuracy, however.

With computation of cost of computation the assessment will be more detailed enhancing the choice of the best model by the project manager in Agile environments in terms of accuracy and computational costs.

The given graph in relation to the K-Fold cross validation parameter. The blue bars corresponds to the accuracy of the individual model while the red line represents the relative time consumed for prepping the model.Although Random Forest (RF) and Gradient Boosting Machine (GBM) perform slighly better than 0.85 accuracy, they are also very expensive in that it takes 12-15 seconds per fold during training.K-Nearest Neighbors (KNN) and Logistic Regression (LR) predict low values of 0.75 and 0.68 respectively where training KNN models takes less that 4 seconds and 2 seconds for LR making them favourable in offering predictions in real life situation or when renovations are needed in most cases.This shows the willingness of presentation of efficiency in predicting a particular phenomenon at a great cost which is equally important in an Agile setup where the selection of model involves time and such parameters.

6.7 Discussion

There is selection bias in the choice of models and datasets, which may limit the study's generalizability. These models selected were strong performers but further studies should examine alternatives or ensemble models to avert this kind of prejudice. In addition, more research can look at these findings across multiple domains and varied data sets with real-time data inputs for better model adaptiveness. Moreover, it has made significant contributions in the area of agile project management as it has proven that post-categorization metrics improve decision-making and resource allocation. Convergence on such insights is possible by integrating such models into project management software that support data-driven decisions for optimal project outcomes.The research was performed using strong techniques within the CRISP-DM framework to ensure high quality dataset from various open source software projects and extensive pre-processing. Even though KNN, Logistic Regression, as well as SVM models showed good results, use of advanced models or hybrid approaches might be an option to improve accuracy even further. This result confirms earlier studies that task classification enhances model accuracy considerably especially in KNN which achieved 87-89%. The qualitative analysis used here shows that simple models like KNN are superior

Model	Accuracy (Before)	Precision (Before)	Recall (Before)	F1-Score (Before)	Accuracy (After)	Precision (After)	Recall (After)	F1-Score (After)
K-Nearest Neighbors (KNN)	75%	0.75	0.75	0.75	87-89%	0.87-0.89	0.87-0.89	0.87-0.89
Logistic Regression (LR)	68%	0.68	0.68	0.68	67-71%	0.67-0.71	0.67-0.71	0.67-0.71
Random Forest (RF)	72%	0.72	0.72	0.72	72%	0.72	0.72	0.72
Gradient Boosting Machine (GBM)	71%	0.71	0.71	0.71	67-71%	0.67-0.71	0.67-0.71	0.67-0.71
Support Vector Machine (SVM)	72%	0.72	0.72	0.72	71-72%	0.71-0.72	0.71-0.72	0.71-0.72

Table 3: Comparison of Model Performance Before and After Categorization

7 Conclusion and Future Work

Presently, this research has shown novelties in the use of advanced NLP and ML techniques for the estimation of effort in Agile projects. In particular, by combining the GPT-2SP tokenizer with several machine learning techniques, the accuracy of story points predictions increased. That is, the transparency of attribute values of the tokenizer indicated which linguistic characteristics influenced predictions, resulting in improved effort estimation. Reliable prediction of story points assists with effective planning and scheduling and promotes data-driven decision-making by eliminating reliance on personal opinions, thereby enhancing project results. It is worth noting that task prioritization had a major impact on the system performance of K-Nearest Neighbors (KNN), which had significantly higher accuracy rates compared to other models when dealing with high-priority tasks. That shows just how important it is for models to have a way to categorize tasks and this has been backed by research showing that KNN performs well in such cases Zhang (2016)Hastie and Friedman (2009)Cunningham and Delany (2007).

In the future, richer contextual data will be integrated to explore advanced models in deep learning and transformer architectures with the hope of improving their accuracy and interpretability. By incorporating user feedback, these models can be fine-tuned , allowing them to adapt to different project environments. Furthermore, scalability will have to be ensured with real-time predictions as well as generalization of the models across multiple domains. In addition, this will help in advancing better and more efficient

approaches for project management as it continually modifies the estimation criteria for effort and consequently helps improve overall project success.

References

Abadeer, M. and Sabetzadeh, M. (2021). Machine learning-based estimation of story points in agile development: Industrial experience and lessons learned, *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, IEEE, pp. 106–115.

Bala, Y. Z., Samat, P. A., Sharif, K. Y. and Manshor, N. (2022). Improving cross-project software defect prediction method through transformation and feature selection approach, *IEEE Access* **11**: 2318–2326.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R. et al. (2001). Manifesto for agile software development.

Cunningham, P. and Delany, S. J. (2007). k-nearest neighbour classifiers, *Technical Report UCD-CSI-2007-4* **123**(3): 1–17.

Fu, M. and Tantithamthavorn, C. (2022). Linevul: A transformer-based line-level vulnerability prediction.

Fu, M. and Tantithamthavorn, C. (2023). Gpt2sp: A transformer-based agile story point estimation approach, *IEEE Transactions on Software Engineering* **49**(2): 611–625.

Hastie, T., T. R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edn, Springer.

Hussain, A., Roy, K. and Banerjee, S. (2013). Svm-based efficient image classification system, *Journal of Machine Learning Research* **14**(1): 1201–1213.

Johnson, J. M. and Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance, *Journal of big data* **6**(1): 1–54.

Lessmann, S., Baesens, B., Seow, H. V. and Thomas, L. C. (2008). Benchmarking classification models for software defect prediction: A proposed framework and novel findings, *IEEE Transactions on Software Engineering* **34**(4): 485–496.

Luo, Y., Shi, J., Tan, J., Ren, Z., Wan, J., Safran, M. and AlQahtani, S. A. (2024). An ensemble data-model-label three-level regularization framework for imbalanced intelligent fault diagnosis, *IEEE Transactions on Reliability* .

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G. and Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review, *ISPRS Journal of Photogrammetry and Remote Sensing* **152**: 166–177.
URL: *https://www.sciencedirect.com/science/article/pii/S0924271619301108*

Madampe, K., Hoda, R. and Grundy, J. (2020). A multi-dimensional study of requirements changes in agile software development projects, *arXiv preprint arXiv:2012.03423* .

Marapelli, B., Carie, A. and Islam, S. M. (2020). Rnn-cnn model: A bi-directional long short-term memory deep learning network for story point estimation, *2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, IEEE, pp. 1–7.

Menzies, T., Greenwald, J. and Frank, A. (2007). Data mining static code attributes to learn defect predictors, *IEEE Transactions on Software Engineering* **33**(1): 2–13.

Molokken, K. and Jorgensen, M. (2003a). A review of software surveys on software effort estimation, *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*, IEEE, pp. 223–230.

Molokken, K. and Jorgensen, M. (2003b). A review of software surveys on software effort estimation, *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*, IEEE, pp. 223–230.

Nassif, A. B., Capretz, L. F. and Ho, D. (2012). Estimating software effort using an ann model based on use case points, *2012 11th International Conference on machine learning and applications*, Vol. 2, IEEE, pp. 42–47.

Navakauskas, D., Skirelis, J., Šabanovič, E., Kazlauskas, M., Levitas, B., Naidionova, I., Drozdov, M., Prisiažnyj, A. and Kazharov, M. (2022). Application of convolutional deep neural network for human detection in through the wall radar signals, *DAMSS 2022: 13th conference on data analysis methods for software systems, Druskininkai, Lithuania, December 1–3, 2022.*, Vilniaus universitetas, pp. 66–67.

Norris, J. M., Brown, J. D., Hudson, T. D. and Bonk, W. (2002). Examinee abilities and task difficulty in task-based second language performance assessment, *Language Testing* **19**(4): 395–418.

Popli, R. and Chauhan, M. (2014). Gradient boosting model for classification in data mining: A review, *International Journal of Computer Science and Information Technologies* **5**(5): 6374–6377.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. et al. (2019). Language models are unsupervised multitask learners, *OpenAI blog* **1**(8): 9.

Radford, B. W. (2014). *The effect of formative assessments on language performance*, Brigham Young University.

Ramchurreetoo, D. and Hurbungs, V. (2022). Application of k-nearest neighbors in machine learning: A comprehensive review, *Journal of Computational Science* **18**(4): 345–352.

Satapathy, S., Behera, R. and Parida, S. (2016). Random forests for classification in machine learning: A review, *International Journal of Engineering Research and Technology* **5**(10): 635–640.

Satapathy, S. M. and Rath, S. K. (2017). Empirical assessment of machine learning models for agile software development effort estimation using story points, *Innovations in Systems and Software Engineering* **13**(2): 191–200.

Shah, R., Shah, V., Nair, A. R., Vyas, T., Desai, S. and Degadwala, S. (2022). Software effort estimation using machine learning algorithms, *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, pp. 1–8.

Stollnitz, B. (2023). The annotated gpt: Understanding the transformer model, `https://bea.stollnitz.com/blog/gpt-transformer/`. Accessed: 11 August 2024.

Zahraoui, H. and Idrissi, M. A. J. (2015). Adjusting story points calculation in scrum effort & time estimation, *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*, IEEE, pp. 1–8.

Zaidi, F. and Jain, R. (2024). An analytical framework for reliable agile story point estimation, *Journal of Software Engineering and Applications* **15**(2): 45–60.

Zhang, W., Yang, Y. and Wang, Q. (2013). A study on software effort prediction using machine learning techniques, Vol. 275, pp. 1–15.

Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors, *Annals of Translational Medicine* **4**(11): 218–220.