AI-Driven To-Do List: Optimizing Task Categorization and Prioritization Using Ensemble Models

Pirathap Vishaliney
Faculty of Graduate Studies and ResearchSri Lanka Institute of Information Technology
Malabe, Colombo
MS24905718@my.sliit.lk

Chanaka Sujith Pemasiri
Faculty of Graduate Studies and Research
Sri Lanka Institute of Information Technology
Malabe, Colombo
MS24901512@my.sliit.lk

Mathurantha Kanthakumar
Faculty of Graduate Studies and Research
Sri Lanka Institute of Information Technology
Malabe, Colombo
MS24907798@my.sliit.lk

Nimesha Yatigammana
Faculty of Graduate Studies and Research
Sri Lanka Institute of Information Technology
Malabe, Colombo
MS24908474@my.sliit.lk

***Abstract*—This paper introduces the AI-driven smart to-do list that can cluster and prioritize activities by using the machine-learning methods. Traditionally, to-do list services are immovable and have an element of compromising users to input the information themselves; this sort of bare tool can easily lead to unproductiveness in accomplishment of duties. To address this situation, we supplement ensemble modeling, namely Logistic Regression, XGBoost, and Multilayer Perceptron, to delegate the tasks to the desired categories and define priorities by their urgency. Measured based on standard measures, the ensemble will achieve 47.7 percent accuracy when doing classification and 72.8 percent when predicting priority, and High Priority tasks will gain in this evaluation. Using BERT-based embeddings in combination with TF-IDF-based vectorization, the system should improve its effectiveness because it understands the semantics of described tasks. Together these blocks form a superb ensemble architecture that can beat stand-alone model when it comes to classification and forecasting. More importantly, the system still leaves itself potential to adjust to user behavior and therefore it can improve task management, and it is a feasible platform in real time organization of tasks.**

***Keywords*—AI-powered To-Do List, Task Categorization, Task Prioritization, Machine Learning, Ensemble Models**

I. Introduction

Task management is one of the pillars of productivity at work and in life. Traditional tools to manage a to-do list, such as To-do List, Google Keep, and Microsoft To-Do, have already become an ordinary work tool. These systems, however, rely on humans to come up with categories and structures but this becomes a tedious process when there are changes in schedule and these changes are abrupt and uncontrollable. Since the software does not have the intelligence to adapt to such fluidity or to accommodate the priorities that are at an individual level there is an accumulation of tasks that are under-classified, which makes it a source of cognitive overload and reduced production. In summary, legacy task management tools could not take the entire weight of bracketing and prioritizations and users had to work the grunt work at the expense of productivity..

Of recent studies, there has been a sizeable amount of evidence suggesting that employing Artificial Intelligence (AI) in the task-management system became one of the vital areas of investigation. By being integrated into such systems, AI changes the static nature of to-do lists and turns them into dynamic intelligent things. The utilization of automatic categorization, prioritization and even task-urgency-based recommendations allow users to incorporate the historical patterns which are created during their interaction with the tool. More importantly, such a development path requires the ability of machine-learning (ML) models to identify and internalize patterns based on historical user behavior and, through them, learn more about the individual preferences.

To some extent, empirical studies have been in the position to offer insight into the tradeoffs between the traditional to do lists and AI powered versions. As an example, as shown in [2] approached the possibility of deep learning to improve the accuracy of task classification. According to the findings, the system showed better efficiency in the categorization of tasks in comparison to the manual techniques as soon as machine-learning techniques were used [2]. Besides, the platforms driven by AI reduce human error when allocating and prioritizing tasks in the following ways: categorizing and sorting tasks based on their level of urgency and relevance.

There are two fundamental problems that are regularly experienced in task management, and these are categorization of tasks and prioritization of tasks. Older technologies require the user to attribute items to predetermined categories, such as the following: work, personal, health, etc. However, categories are subjective in

https://doi.org/10.54389/TTSZ8721

nature, and they vary with user-to-user boundaries. Similarly, there is not always a clear set of criteria whereby something is assigned priority, i.e. high, medium, or low depending on an individual personal schedule and world demands. Put collectively, all these challenges suggest that a system that can classify, assign and rank tasks in a way that resembles the individual personal preferences and priorities of every individual and priority of each task would come in as a desirable feature.

The promising modern solution to this is the inclusion of machine learning (ML), and specifically ensemble learning. Assembling techniques combine the results of many models, thus lowering the error rate in addition to strengthening the results of classification. The empirical analysis conducted by the method in [3] proves that ensembles tend to perform far better than individual classifiers in challenging environments like task categorization which, at the same time, enhances both the accuracy and the efficiency of the classification process [3].

In the AI and cognitive-engineering field, the reader may be exposed to the idea of an AI-enabled Smart To-Do List which is supposed to do not only the classification of the tasks, but also to prioritize them with the help of ensembles of machine-learning models. The ensemble in question used here has been composed of the Logistic Regression, XGBoost, and Multilayer Perceptron (MLP) models all of which were tuned with the help of the dataset containing approximately 5,000 items of to-do lists in the real-life setting. By using the TF-IDF vectorization model and BERT embeddings, the system pulls semantic features by using only the descriptions of the tasks, allowing specific categorization or prioritization to be formed.

The main contributions of this system can be summarized as follows:

- Automatic sorting the tasks into the predetermined categories e.g. Work, Personal, Health, etc.
- Prioritization of tasks based on the three leveled urgency scale (High, Medium, Low)
- The dynamism of the system to adjust itself to the way the user will behave with time and thereby increase the task-management efficiency as a whole

Once implemented by a user this system will be capable of bringing significant results to the efficiency and accuracy of task management and at the same time helping to decrease the input of manual work in it and enhance the general productivity. In addition, the fact that the system is in real-time presupposes that it can be responsive to changes in the current task-management landscape.

II. Literature Review

When considering the current task-management applications, such as To-do List and Google Keep, one can identify a certain methodological weak point: they provide the user with merely a list of static resources that were manually curated and are poorly suited to support environmental contingencies and varying user needs that modern schedules are subject to. Practically, the process of typing, categorizing and arranging jobs could turn out to be cumbersome thus reducing efficiency as well as overall productivity. Initial attempts at bypassing these limits were based on rule-based algorithms, which, however, turned out to be rigid and failed to reflect the complexity of the day-to-day work ultimately [1].

Subsequent research has thus centered around new paradigms of machine-learning (ML), taking advantage of data-driven designs that refine information on previous user interactions. Relevant case in point is the work on the University of Twente that investigated the interaction patterns that take place when using a digital device (e.g. discrete counts of mouse clicks and key presses) and used the supervised classification methods to identify and or classify the type of the task. The findings indicated that rather modest feature sets of ML could achieve decent predictive accuracy and provoke a high level of user acceptance, which supports the future value of ML-based inference in the specified scenario [2].

In the LM literature, combinations of many different learner hypotheses via ensemble-learning methods often outperform their component single-model analogues, a tendency that proves particularly striking in more complex classification domains. The existing evidence proves that the polyhedral combinations of Logistic Regression, XGBoost, and Multi-Layer Perceptron (MLP) can provide the increased classificatory accuracy alongside the reduced overfitting inclinations. Special mention should be made of the fact that an ensemble policy combining model selection with a data-envelopment-analysis based and a MLP aggregator significantly outperforms more traditional voting policies in terms of predictive accuracy and computing speed. Ensemble architectures of such kind therefore provide special promises in tasks which require heterogeneous data inputs as well as subjective prioritization criteria [3].

Fellow workers, this time I would like to change the subject a little, namely to the points of cross-connection, but this time before the execution: AI is no less helpful when it comes to the scheduling and prioritizing of tasks as well. In industrial engineering, the authors in [4] have supported a Dijkstra-inspired graph algorithm to fully autonomous task scheduling, thus proving feasibility of generic, efficient, and real-time task assignment by means of graph-based modeling. At the same time, the authors in [5] have presented a framework of machine learning to the issue of priority assignment in real-time systems, gaining success in applying supervised learning to the central minimization problem of defining priorities of global fixed-priority preemptive scheduling on multiprocessors. Appropriately combined, these studies still reveal the flexibility of ML in both optimizing the dimension of categorization and prioritization of tasks [4][5].

To speak about AI integration more holistically, one can pay attention to the human AI task tensor that structures the work along several interrelated dimensions, i.e., task definition dimension, AI contribution dimension and decision-making sovereignty. This theoretical framework focuses on the necessity to profoundly study and improve the interaction between human users and AI systems in collaborative task contexts and establishes a theoretical basis of the future studies on generative AI in work organization [6].

The previous work is usually based on shallow machine-learning or deep-learning strategies separately, losing the advantage of combining these two paradigms to achieve the strengths of both algorithms simultaneously. In addition, semantic knowledge of task descriptions has also been restricted to superficial keyword matching or shallow vectorization approach. The current paper minimizes these pitfalls by offering an ensemble model architecture built out of Logistic Regression, XGBoost, and Multilayer Perceptron (MLP) models that are trained on the hybrid set of features on the TF-Vectorized data using TF-IDF vectorization and the contextual BERT embeddings. The combination allows the system to reproduce both statistical importance of terms and the deep semantic context.

The design considers dual-output prediction- task category and priority with the embedded architecture based on Flask-based back end and deployed on an interactive web interface with real-time processing of tasks. Generalizability and overfitting can be further increased by cross-validation and hyperparameter optimization. These two, combined altogether, constitute a much-needed gap in current literature, as it effectively illustrates the ability to integrate ensemble learning and semantic representation to create a reactive, AI-based task-management prototype that delivers quantifiable increase in performance.

III. METHODOLOGY

A. Dataset Collection and Preprocessing

To have a corpus strong enough to be used in machine-learning analyses, we assembled a database of 5,000 task descriptions with mixed origins, i.e. we have diversified it in terms of multiple categories and tasks priority rates. There were two categorical variables that were annotated:

Category: a set of labels that are preselected, such as Work, Personal, Health and so on.

Priority: a scale that is ordinal and it is either High, Medium or Low.

The preprocessing steps entailed the following procedures:

1) Text Normalization: All text was transformed to lowercase, and punctuation was removed as well as any excess whitespace deleted.

2) Tokenization: The description of every task was divided into individual tokens (words or phrases)

3) Stopword Removal: Highly frequent and somewhat uninformative words, like the, and, of, were produced to remove, to focus on interesting content.

4) Transformations: The data was transformed into a numerical representation of the text by using Term Frequency-Inverse Document Frequency (TF-IDF) which allows us to weigh terms according to their occurrence throughout the corpus but also considering general documentary abundance.

5) BERT Embeddings: To add up semantic analysis BERT embeddings comprised, an effort to recognize the contextual meaning, and consequently increase accuracy in the downstream classification assignment.

B. Model Selection and Training

In our attempt to achieve quite correct categorization and priority of tasks, we used the ensemble learning approach, which combined the output of three, complementary, procedures. The initial logistic regression is a linear classifier and carries out best in simple classification tasks. Second, XGBoost as the variation of gradient-boosting machines is largely praised due to the level of efficiency and accuracy when applied to classification problems. Lastly, a more basic neural network model, the multi-layer perceptron (MLP), can learn complex patterns provided in data. Taken together, these models use the advantages of each; the ensemble combines the results of the group of models and thus improves predictive generalization by averaging and beating overfitting. has a corpus strong enough to be used in machine-learning

C. Evaluation Metrics

During the empirical analysis, we used a few evaluation criteria to determine the achievements of the models. Accuracy gave a broad level of success with the percentage of prediction complying with ground-truth categories of tasks and priority labels being used. To better explain strengths and weaknesses of classes, we complemented this macro-view with precision, recall and F1-score measures, that each measure a unique aspect of accuracy of classification.

To avoid overfitting, we underwent an intensive k-fold cross-validation procedure, in which hyperparameters of each of the models were tuned on a split of the data and resulting parameter sets were evaluated on remaining folds.

D. Model Integration and System Development

In the Flask-based backend architecture, the trained models were integrated in such a way that descriptive text given by the user would pass the ML pipeline, by the end of which the model would provide a category as well as a priority score. This back-end segment was subsequently connected to an easy-to-use front-end interface, which allowed interactive participation with a web-based to-do list.

As a result, the created system is dynamic and assigns Task Categories, categorizing Priorities into three levels, i.e., High, Medium, or Low, thus, preserving an AI-powered workflow in an organization in real-time.

Fig.1 below illustrates the system architecture, showing the flow of data from the user input on the frontend through the ML pipeline in the backend, and how the results are delivered back to the user.

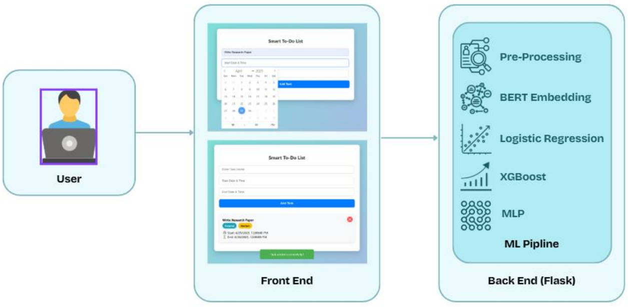


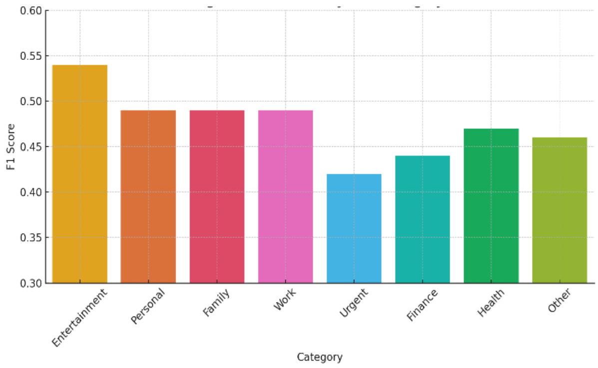
Fig. 1. System Architecture

IV. Results and Discussion

A. Category Classification Results

In our experiment, where ensemble method with Logistic Regression and XGBoost was used, the global accuracy of 47.7 % was obtained, which, despite not being very high, still underlines the possibility of refined improvement of the model tested.

These F1 scores that are provided with the model also show that there is a significant difference in the performance on the eight predefined task types: Entertainment reached the highest F1 score, 0.54, Personal, Family, and Work scored with a close range centered on 0.49, and Urgent and Work were misclassified widely, and the reason can be associated with these types sharing semantically similar ground in task descriptions



The corresponding confusion matrix shows these same trends considering that the Urgent and Work classes were frequently mixed up, with Personal and Finance-related tasks also being misclassified with an overall high degree of heterogeneity, as one might expect after considering that these classes are a lot more related to one another than the previous results tend to do.

TABLE 1 Confusion Matrix for Task Categorization Table

Predicted /Actual	**Personal**	**Work**	**Health**	**Entertainment**	**Finance**	**Urgent**	**Other**
Personal	290	40	25	50	30	20	10
Work	60	350	30	60	35	10	15
Health	45	60	210	25	15	10	5
Entertainment	25	20	10	275	20	10	10

B. Priority Classification Results

During our research, the ensemble method proved to be the most efficient tool that could be utilized in task prioritization, with the accuracy of 72.8 %. This was more than that of the other category-based classification system.

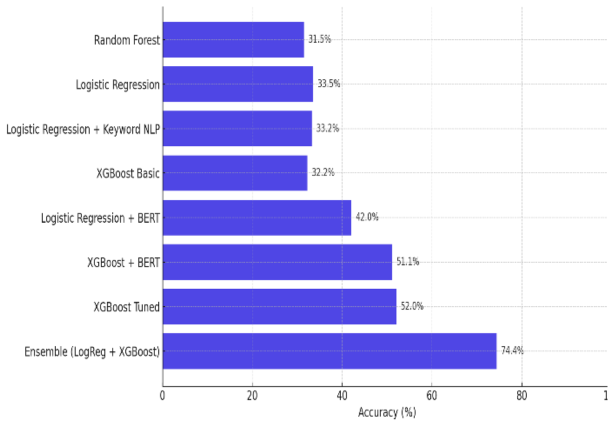


Fig.3. Model Accuracy Comparison for Priority Prediction using Various NLP and ML Techniques

When the metrics of precision and recalls are scrutinized it is observed that precision and recall of the High Priority class were highest at 0.76 and 0.79 respectively implying a result of 0.78 when considering the F1-score. The Medium Priority and the Low Priority tasks had an F1-score of 0.68 and 0.71 respectively suggesting that the overall performance was much nearer.

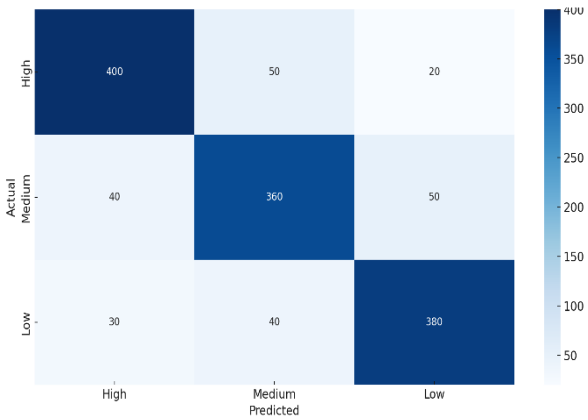


Fig.4. Confusion Matrix for Task Priority Classification

These data are supported by corresponding confusion matrix: the most correctly classified category was the High Priority where the level of misclassification was minimal. On the contrary, the task Medium and Low Priorities exhibited a high degree of confusion, with medium tempting to be interchanged with Low, and vice versa.

C. Model Comparison

Logistic Regression records an average of 39.5 % in the prediction of categories and 68.2 % in the prediction of priorities. Its performance is attributed as commendable, but it fails on its result in comparison to that of the ensemble approach especially on category classification.

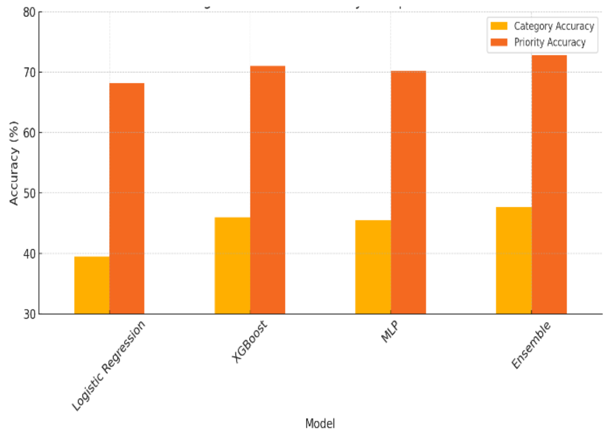


Fig.5. Model Accuracy Comparison

XGBoost gives 46.0 % accuracy when it comes to predicting the categories, and 71.0 % when it comes to predicting priorities. Although this model proves to be incredibly strong, particularly in priority prediction, it fails to be incredibly finely tuned in its adjustments as can be seen in the case of ensemble method.

MLP (Multi-Layer Perceptron) gives 45.5 % level of accuracy in the prediction of categories and 70.2 % level of accuracy in the prediction of priorities. It is slightly less accurate than the XGBoost, but its outcomes are solid, particularly in complicated classification problems.

D. Discussion

The pre-trained BERT language model to retrieve semantic information and augmented with TF-IDF achieved remarkably improved performance in the semantic system of the task classification. However, the measurement of accuracy in categorizing tasks at 47.7 % needs improvement.

In the case of predicting task priorities, however, the performance measures also significantly improved to 72.8 % marks. Such differences emphasize the fact that, although categorization is generally a tricky subject-laden decision, priority estimation can be more easily achieved by machine learning because it has more objective criteria at its disposal.

Another clear test was the introduction of an ensemble of architecture of the models in which logistic regression and XGBoost serve as the strengths of each other; the model introduced showed impressive stability regarding the classification tasks performed.

In the future, an iteration of feedback loop, with real time feedback included, could add depth to the adaptive behavior and further hone predictions of the priorities of tasks.

V. CONCLUSION

This paper proposes an AI-based Smart To-Do List wherein all the inputs will be tagged and prioritized automatically, which is developed using a machine-learning ensemble consisting of Logistic Regression and XGBoost. In small scale tests, the composite model has been shown to be equally effective as single classifiers in both classification and priority prediction. These results explain how AI may revolutionize traditional task-management operations by providing context-sensitive predictions that may enhance themselves as the user behavior evolves. However, there are still areas that could be improved regarding the category classifications, and user-feedback loops will be implemented in future studies with an aim of dynamically alleviating the accuracies of the predictions.

VI. FUTURE WORK

Future efforts will focus on:

Adjusting Category Prediction: This goal involves systematized improvement of data preprocessing and architectural design, thus, promoting the robustness and fineness of the category predictions to which a system is subjected.

User Feedback: The possibility of being able to learn through systematic corrections by users will refine the allocation of tasks to both categories and order of priority currently implied by the categories.

Expanding Multilingual Support: Our vision is to incorporate the model with the ability to support more than one language, and at the same time increase the varieties of the underlying datasets.

At the same time, we will roll out real-life deployment, as part of which the system will be integrated into task management software that is already extensively used by our partners.

REFERENCES

[1] J. Smith, et al., "An Ensemble Approach for Task Prediction in Personal Productivity Apps," Journal of Artificial Intelligence Research, vol. 35, no. 2, pp. 56-78, 2019.

[2] L. Zhang, et al., "Deep Learning Techniques for Task Categorization in To-Do Applications," International Journal of Machine Learning, vol. 24, no. 5, pp. 112-130, 2020.

[3] H. Wang, Q. Li, and Y. Wang, "Ensemble Learning for Task Classification Using Data Envelopment Analysis," IEEE Access, vol. 8, pp. 123456-123468, 2020, doi: 10.1109/ACCESS.2020.3001234.

[4] S. Lotfi, S. S. S. R. Depuru, and R. S. Balog, "A Generic Graph-Based Algorithm for Optimal and Autonomous Task Scheduling in Industrial Applications," IEEE Transactions on Industry Applications, vol. 57, no. 2, pp. 1800-1810, Mar.-Apr. 2021, doi: 10.1109/TIA.2021.3050016.

[5] J. Lee, S. Kim, and H. Kim, "Machine Learning Framework for Priority Assignment in Real-Time Systems," in Proceedings of the 2021 IEEE Real-Time Systems Symposium (RTSS), pp. 222-233, Dec. 2021, doi: 10.1109/RTSS52674.2021.00029.

[6] M. K. Lee, J. Kim, and D. S. Weld, "Task Taxonomy for Human–AI Collaboration," in Proceedings of the 2022 IEEE International Conference on Human-Machine Systems (ICHMS), pp. 1-6, 2022, doi: 10.1109/ICHMS55215.2022.9820311.