Characterizing and Predicting Tasks at Risk in Team Task Management

by

Nouran Soliman

B.Sc, Arab Academy for Science, Technology and Maritime Transport

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2022

Author . . . .
Department of Electrical Engineering and Computer Science
January 21, 2022

Certified by . . . .
David R. Karger
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Characterizing and Predicting Tasks at Risk in Team Task Management

by

Nouran Soliman

Submitted to the Department of Electrical Engineering and Computer Science
on January 21, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

Collaborative project management involves interacting with various tasks in a shared planning space where members add, assign, complete, and edit project-related tasks to have a shared view of the project's status. This process directly impacts how individual team members select, prioritize, and organize tasks on which to focus on a daily basis. However, such coordination and task prioritization can become increasingly challenging for individuals working on multiple projects with big teams. Accordingly, tasks could become at risk and eventually not be completed on time, leading to personal or team losses in many situations. To support task-doers in completing their tasks, we conducted a mixed-methods study focusing on Microsoft Planner—a collaborative project management tool—to understand how users manage their tasks in a team setting, what challenges they encounter, and their preferred solutions. Based on the findings from a qualitative survey with 151 participants and our Planner log data analysis, we further developed a task at risk prediction model using various task characteristics and user actions. Our experimental results suggest that a task at risk can be classified with high effectiveness (accuracy of 89%). Our work provides novel insights on how users manage their tasks in team task management tools, what challenges they face, how they perceive a task at risk, and how tasks at risk can be modeled. Such an application can significantly improve the user experience in such tools by providing a personal assistant that helps users prioritize their tasks and pay attention to critical situations.

Thesis Supervisor: David R. Karger
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

This research was conducted at Microsoft. I gratefully thank Mark Encarnación, Michael Gamon, Nalin Singal for their guidance through out this research. A special thanks to Nadin Merali and Si Meng from Microsoft Planner Team who provided us with all the resources we needed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

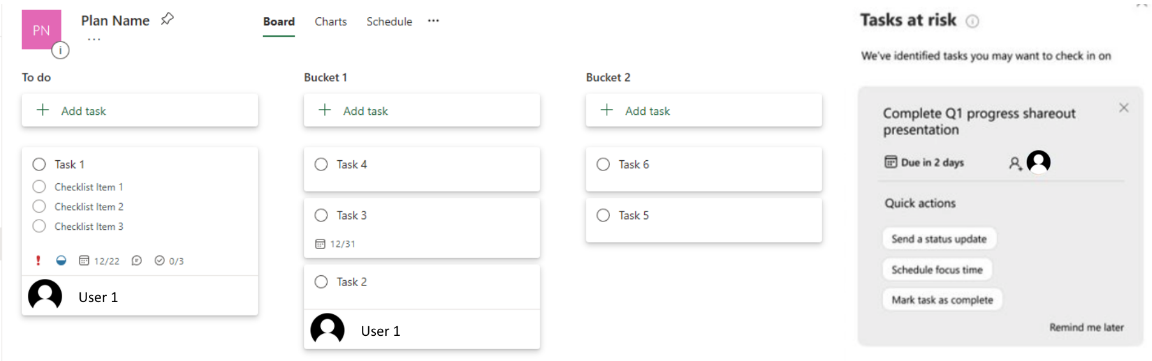


Figure 1-1: Microsoft Planner Snapshot. Planner has 4 main entities: plan, bucket, task and user. Our vision is to bring user's attention to tasks that are unlikely to be completed successfully.

Collaboration is an essential component in communities such as organizations. Previous research has extensively studied how people collaborate over long distances [48, 58], in virtual environments [5, 24], and in in-person settings [48, 58], in various fields [48, 35, 59]. In a team, collaboration is defined as working on a common goal and sharing responsibilities between team members with mutual influence via open communication, conflict resolution, and innovation support [15, 37, 2, 3].

1.0.1 Project Management

Project management aims to facilitate collaboration in teams [42], however, it still remains challenging due to communication problems, failure to meet project objectives, and limited resources, which are critical factors for successful project management [60]. Therefore, asynchronous computer-mediated collaboration tools are being developed by many companies and researchers. Such technologies have automated many work processes and aided in project execution in remote and in-person collaborations.

1.0.2 Challenges of Team Task Management Tools

Although project and task management tools have significantly improved team collaboration, there have been many challenges arising from using these tools, including integration with existing software tools and processes, lack of customization, and lack of motivation to use the tool. Previous research on task management has spanned multiple lines of research such as formulating a theoretical framework for task categorization [19, 36], intelligently classifying tasks such as micro-task detection [63], estimating task time [61], predicting task difficulty [40, 31, 38], task prioritization [65, 55], and much more. However, little is known about how task management tools are utilized in teams day-to-day, what challenges and risks users face when using such tools in a team project, and what can be done to facilitate task management in teams.

1.0.3 Microsoft Planner

This paper focuses on understanding how people manage their tasks day-to-day in teams, what challenges they face, how they envision an ideal task management environment, and how they mentally categorize and identify tasks that they are unlikely to get done successfully on time. We focus on Microsoft Planner team task management tool in our study as we had access to huge and diverse datasets of real teams and projects donated by users, which could support the generalizability of our results. Microsoft Planner has 4 main entities: plan, bucket, task, and user as shown in Fig.1-1. A plan is equivalent to a project or a team space where tasks are managed. Inside of a plan, users can categorize their tasks into buckets. Each task has various attributes to describe it such as priority, due date, description, checklist items, etc. Users can assign and get assigned zero or more tasks. Our goal is to be able to identify and notify users with tasks they need to pay attention to.

Our work is composed of three main stages: a qualitative study to understand users' perspectives, a quantitative dataset analysis to confirm and generalize qualitative insights and extract patterns around task management, and machine leaning (ML) modeling of tasks that are likely to fail. This thesis is organized as follows: Section 2 describes related work and our contributions, Section 3 describes our methods as well as qualitative results, Section 4 describes quantitative data analysis, Section 5 formulates our prediction problem and presents our prediction results. Discussion and Design Implications, and Conclusion are discussed in Sections 6 and 7, respectively.

Chapter 2

Related Work

Numerous tools, methods and techniques have been studied and developed to support collaborative task and project management. This section analyzes relevant previous work on task and project management, and discusses various assistive studies and applications in this domain.

2.0.1 Task & Project Management Methods & Behaviors

Various project management methods are valuable for all project types and fields [11]. As teams seek methods that are more easily adapted to their needs, agile approaches and best practises have been gaining a lot of popularity today. For example, the Essence specification [56] defines a framework that allows teams to describe practices and rules (e.g. Scrum), in terms of concepts, such as checklists, states, artefacts, competencies, templates, etc, to be be followed, and monitored by task management tools. Another project management method focuses on describing collaboration phases. In virtual teams, a 5-stage heuristic lifecycle model is described [23] with specific tasks and topics that should be addressed by team. The 5 stages of this model are: mission preparation, activities launch, performance management, team development, and achievements recognition. Another survey [60] shows that current project management methods include digital management tools, decision-making techniques, risk assessment tools, and computer models. Various methods overlap in scope and methods, which makes selecting a suitable project management challenging. Therefore, teams increasingly follow lightweight approaches, such as in task management tools. A more recent qualitative study explored the features and challenges teams face in collaboration while using a task management tool UpWave [11]. This study highlighted multiple challenges teams face including prioritizing and organizing projects and tasks, task delegation in terms of task owners and due dates, dividing projects into smaller and concrete tasks, and keeping track of time allotted to tasks. However, this study did not explore how users manage and select their tasks on a daily basis in teams.

2.0.2 Task & Project Management Tools

Many task management software tools support teams collaborations in various ways such as knowledge management, coordination, information exchange, communication, and collaborative learning [17]. In this section we examine some well-known software tools supporting team task management, team communication, and personal and team productivity management. Trello, Asana, Teamwork and Microsoft Planner are all examples of tools that support online project and task management. These tools share a common infrastructure which provides users with features such as tasks, projects, conversations, notifications, calendars, comments, file attachments, progress views, and dashboards. Users can create checklists, add labels and due dates, assign tasks to people, and connect with other applications. One example of tools that support team communication is Slack. Slack provides an open channel to organize team conversations for a project, a topic or a team by providing various features including direct messaging, file sharing, various privacy levels and integration with other software. There has been a number of tools focused on team productivity management such as Todoist and Microsoft Todo. These tools provide personal or collaborative spaces to manage to-do lists with many other features such as setting up tasks, collaborating on a shared task and different filtering and categorization methods. These tools are focused on creating to-do lists unlike Microsoft Planner which is centered around creating tasks in team plans.

2.0.3 Task Assistance Applications

Extensive research has been done on analyzing the task management space and building better and smarter tools to aid users in managing their tasks. We reflect on four main lines of work: time management and planning, time estimation, task classification, and productivity tracking.

Time management & Planning

Previous work on time management and planning has focused on finding best practices to help people manage their time more effectively personally [1], which yielded the development of tools to better support personal time management [6, 46] including helping users to organize, filter, prioritize, and execute tasks. A lot of these systems depend on intelligent scheduling systems which help individuals [25, 51] and teams [26, 13] find time for tasks and coordinate schedules of multiple individuals. Digital assistants such as Amazon Alexa, Google Assistant, and Microsoft Cortana also play an important role in helping people track short time durations during their day and create reminders to remember to perform future tasks [20] without the need of a precise task timeframe [52].

Time Estimation

Research has shown that people usually are subject to planning fallacies [7, 29] where they underestimate or overestimate the time taken to complete their tasks [28, 29, 32, 18] which is mainly due to wishful thinking [50] and overconfidence [45], or lack of experience. Therefore, extensive research has been done on developing new strategies to overcome these biases such as enumerating complex task steps [34] which could be facilitated by assistive technologies to help structure sub-tasks. Furthermore, event durations have been mined from search logs [21, 33] and news articles [49]. Another line of research focuses on using machine-learned models to estimate different properties of tasks, such as task completion status [62] and task duration [61]. Other research uses natural language to reason about temporal aspects of events such as duration by applying machine learning techniques [61, 47, 66, 67].

Task Classification

Prior work has established theoretical frameworks for representing tasks using a facet-oriented approach. For example a task could be described by its goals, complexity, assigner, assignee, interdependence, etc [36, 41]. Similar representations could inspire the training of digital assistants to categorize tasks and help users in task prioritization, an idea which will be touched on by this study. More recently, intelligent systems have been used to automatically classify or detect various kinds of tasks or task attributes. Some applications have focused on supporting micro-tasking [9, 63] with the goal of helping people utilize small amounts of time to work on quick tasks or to progress on larger tasks. Another line of work studies and predicts different characteristics of search tasks including complexity [8], difficulty [4], type [39] and time sensitivity [44].

Productivity Tracking

A number of studies have worked on applying micro-productivity strategies such as task decomposition to help people in completing personal tasks [57]. These strategies have proven to have a positive impact on work quality [12] due to helping people in making progress in short period of downtime. These strategies have been implemented to tools spanning various domains such as writing tasks [27, 30] and software development [64]. Another project incorporated reminders for micro-tasks into social media [22].

2.0.4 Contributions

Prior work focused on understanding how individuals manage, organize and prioritize their tasks individually. In this research, we focus on understanding task management behavior of individuals in a team setting as well as learning what kinds of tasks could fail. We then use our insights to predict tasks at risk. We make the following contributions with this research:

- Develop a generalizable deeper understanding of how people perform individual task management and prioritization in teams, how task management applies to Microsoft Planner, and how users envision an ideal task management environment through a qualitative study.

- Establish a formal Task at Risk definition based on task attributes and user activity through our qualitative study and Microsoft Planner actions log analysis. We find correlations between specific actions and task attributes which describe different task facets.

- Train machine-leaned models to accurately predict if a task is at risk from 4 levels of features: task-level, user-level, plan-level and bucket-level. We experiment with different model architectures (logistic regression, sequence model, ensemble model) and feature sets (task level, user-level, plan-level and bucket-level).

- Present implications of predicting tasks at risk. We also describe the future directions of this work.

Chapter 3

Qualitative Data Collection and Analysis

To address the problem, we took a mixed approach to understand the task management behavior of Planner users and, in the process, identify tasks that are at risk of being incomplete soon. In particular, we took a concurrent triangulation approach [14] in collecting and analyzing qualitative and quantitative data.

With the goal of gaining an understanding of how Planner users use task management tools to collaborate, manage and prioritize tasks in daily basis, we conducted an extensive survey.

3.1 Survey Participants

We recruited 151 participants from a large technology company for a detailed survey. Of the 151 survey participants, 36% were female, 58% were male, 2% were non-binary and 4% chose not to disclose. The median age group was 35-44 years. The majority of participants (85%) held a bachelor's or an advanced degree (e.g., Master's, PhD). Our pool of participants had a very diverse set of work roles including managers, principal researchers, recruiters, software architects, software engineers, designers, and interns with a median experience of 5-10 years. All participants were or had been Microsoft Planner users. We randomly selected one third of the survey participants to be compensated with a $20 Starbucks gift card each. For the purpose of the reporting the results, we labeled the participants with numbers (e.g., Participant 1 or P1).

3.2 Survey Preparation

Based on our literature review (described previously) and initial research on Planner management tools, we prepared the survey questions. The primary goal of the survey was to get a detailed overview of how Planner users manage their tasks on a day-to-day basis, whether and how they use the Planner application, and finally, how they would describe tasks that they are unlikely to get done successfully on time. The insights from the existing work and our exploration of task management behaviors which include studying application documentation, discussions with task management application developers, expert scholars from the task management, productivity, and artificial intelligence research fields, inspired four main lines of research questions that formed the main focus of the survey: (i) how individuals manage their tasks in general, (ii) how individuals use Microsoft Planner in task management in teams, (iii) what the ideal management environment is, (iv) and how they would describe tasks that they are unlikely to complete successfully on time (Tasks at Risk). The survey had 34 questions in total, with multiple choice and free-form questions. Our survey was approved by Microsoft's Internal Review Board.

3.3 Data Analysis

From the survey, we collected 151 unique, complete responses. Using a grounded theory approach [10], we took a multi-step approach to annotate and analyze the responses. In the first annotation round, two members from the research team independently annotated major and broader themes observed in the open-ended descriptive responses. At the end of the first round of annotation, two annotators compared all the themes that emerged in the data, discussed differences and overlapping categories, and finally agreed on a common annotation scheme. Following the same process, the annotators identified more detailed and specific themes under the larger thematic categories from the first stage in the second round. Finally, they used the final annotation schema or codebook to refine the annotations further. Following a grounded-theory approach, it was unnecessary to compute an inter-rater reliability score [43].

3.4 Preliminary Observations

In the following sections, we discuss the some of the major findings from the survey data analysis.

3.4.1 Understanding Individual Task Management in Teams

Our findings from the survey highlighted two types of task management practice in Planner: personal task management on a day-to-day basis and team task coordination on a weekly basis. Based on the list of tasks assigned to participants by the team, participants stated that they flesh out the details of tasks in a to-do list tool or on paper, or block time off on their calendar in order to manage this list of tasks. One participant mentioned that *I tend to manage personal tasks in Outlook by blocking time* [P25]. Three other participants mentioned *Often I rely on To-Do to surface which Planner tasks I need to work on* [P38], *I rely on Outlook, To Do, or OneNote to keep track of my day-to-day work* [P48], and *Honestly, I still keep an offline paper list next to me because planner is not effective as an individual tool* [P49]. These responses emphasized Microsoft Planner's suitability for tracking the big picture of team progress on projects.

3.4.2 Task management in Microsoft Planner

Participants were asked to reflect on an active team plan in Microsoft Planner that they contributed to frequently. Among the plans selected by participants, the mean number of plan members was *7* and the median task completion rate ranged from

25%-75%. The average task lifespan of most plans ranged between a week to a month with *40%* of responses being *2-3 weeks.* Participants highlighted two main purposes behind using Microsoft Planner in team task management: task prioritization for personal task management, and progress communication for team task coordination.

Task Prioritization

Even though participants stated that they use external tools to break down tasks on a day-to-day basis, they still mentioned that they use Microsoft Planner to help them prioritize and select a task list – *I use planner to remind me of what I need to prioritize, and then I usually manually create a list for the week or day based on that of the key items across multiple plans* [P27]. Participants also described various strategies, features and attributes that they use or look for in Planner to help them in task prioritization and selection. One of the most popular strategies was categorizing tasks into buckets where each bucket contains tasks with similar characteristics. Many participants indicated that most of the time they create buckets to describe different lines of work (e.g., modeling and evaluation), different task types (e.g., presentations and logistics) or, progress (e.g., to-start, in-progress, and under-review). Many participants mentioned that they rank and select tasks based on priority, due date and effort: *I triage tasks by priority, deadline, and amount of effort needed to complete–part of my day is structured around long-term or in-depth work, and part of it is for smaller tasks to keep everything moving* [P94]. Furthermore, many participants found some of the planner views such as the schedule view to be helpful in tracking timeline and progress personally and with their team: *I try to organise the tasks into buckets and set as much information as possible (comments, documents, checklists..etc). Charts will give me an idea on the progress that I'm doing on the plan that I'm using (alone or with the team)* [P42]; *I like to have a Plan for my role. Then within that Plan I create buckets to outline the parallel work streams that make up that job. Within those buckets, I create to-do items, set priority, and due dates. I have found the Schedule view to be helpful* [P77].

Progress Communication

Almost all participants indicated that they use Microsoft Planner for *'managing tasks among [their] team'*, and following and communicating the big picture of the project to the team. One participant mentioned that *I use planner to keep a high-level tracking of my tasks. Planner also helps team-members see what I've made progress on.* Various plan management scenarios were selected by participants to describe how they interact with their team. Most participants indicated that they add their own tasks to the plan, interact with their tasks by changing various attributes, and mark their tasks completed when done. In few plans, any team member including managers could also create, edit and assign tasks to any other member. Participants highlighted their interest in communicating their progress to the team immediately when tasks come up or get completed. This behavior contradicts findings from previous research around micro-task detection in to-do lists [63], where users often mark tasks complete in batches, regardless of completion time. This could be due to the absence of the motivation of communicating progress to other people in personal to-do lists, which is one of the main motives in team task management environments as found by our study. To communicate progress, participants interact with and edit various task attributes such as task due date, description, etc based on the current state of their task. In particular, most participants indicated that they had changed the due dates and descriptions of at least one task in this plan before as the task was in most cases *more complex or high-effort than what they expected*, or *they forgot about the task and needed to reschedule it.* This indicates that the action history on a task could provide proxy signals to task complexity, effort needed and much more. If we can learn a pattern from these signals, we will be able to identify these critical tasks earlier in the process and notify the user. This raises the question of what each action in Microsoft Planner indicates and how it is correlated to characteristics of tasks at risk, which will be explored in the following sections.

3.4.3 Ideal Task Management Environment

Difficulty in arranging task priorities is the top challenge participants face as they are engaging with multiple tasks from different plans, a situation that is perceived as overwhelming which causes tasks to slip off their mind and be forgotten, which aligns with some of the previous findings [11]. Participants were asked to indicate what kinds of challenges they face when managing their tasks in teams. In response, one participant mentioned that *I organize by due date and priority. The challenge comes in where there are too many tasks feeding into my day.* [P30]; Participants also reflected on the reasons behind adding tasks to the team plan and never getting back to them. This was mainly due to *being stuck and failing to complete the task*, *the task being complex or requires high-effort*, or *the task being no longer important.* Again, this shows that the activity of interacting with tasks could indicate progress being done on tasks. This could be used to find patterns indicating the state of being stuck on a complex task.

Participants described three main ways in which they envision an ideal task management environment in Microsoft Planner: (i) integrating Microsoft Planner with their preferred individual task management tools, (ii) having more advanced task searching and filtering methods with better visuals and views, (iii) and smart prioritization with reminders. As described earlier, participants reflected on two motives behind using Microsoft Planner: task prioritization and progress communication. Participants showed a huge interest in integrating these two aspects by connecting planner to calendar (*being able to link the task to calendar so that calendar updates (ie changes a category) for tasks assigned to [them] personally* [P25]), to-do list tools (*Associating [Planner] with Microsoft TO DO and other project plans* [P53]), or email (*a great feature would be to take an email from Outlook and turn it directly into a task* [P154]). Such integration will allow participants to have a fuller picture of the team plan status linked to their own individual prioritization space, which will aid in the individual prioritization process. Participants also reflected on the need for both the functionality and user interface of more advanced searching, filtering, labeling, categorization and linking of tasks. Some example quotes include *Ability to filter tasks. Also see the tasks in different ways visually so each person can view the same set of tasks as they wish (i.e. on a calendar, as a list, etc.)* [P23]; *More metrics, more labels, better visual organization* [P44]; *I wish I could connect a task to other tasks in some sort of timeline. A "tag" can connect tasks but I can't 1) sort them or 2) click on one and see the task "flow"* [P33]; Again, such features were mainly connected to providing individuals with clearer plan status that would help them in their own task prioritization. A significant number of participants mentioned that having smart prioritization was something that they need - *Task management should be driven by an "intelligent" prioritization agent that looks across the breadth of assigned tasks and "tee's up" a suggested daily work priority list based on due date, assigned sub-task(s) and progress, etc. to each team member/individual (who can opt in/out at any time)* [P130]; *I would love to see AI that can suggest things like, adding more/missing details. Reminders that the task is getting stale. Suggestions on which task could be completed for the day based on depth of detail score* [P30].

Thus our results show that, even though participants showed interest in three kinds of improvements mentioned above, all requested improvements were directly tied to helping them in individual task prioritization in a team setting. When particularly asked about being reminded of tasks at risk, 66% of participants highlighted their interest in this feature. The majority of the remaining 34% highlighted that even though they ***don't like the computer to think for them*, they would still be interested in this feature if they could understand how the computer calculated this risk.** This shows the potential of such a feature in aiding in task management in teams.

3.4.4 Defining Tasks at Risk

A high level definition of a task at risk was given to participants. A task at risk is a task that has a high chance of not being completed successfully on time. Participants were asked to think of task characteristics that shape a task at risk for them based on this definition. In our literature review, we found that risk could be task related or user related. For example, task complexity is a risk factor that describes a task characteristic whereas progress mostly depends on the task doers' motivations and schedule. Accordingly, we identified two main categories for identifying a task at risk: task characteristics and characteristics of the task doers. In each category, we extract two main themes: (i) intrinsic factors naturally describing the task or coming from the user, and (ii) extrinsic factors coming from the external environment.

Task Characteristics

Participants mentioned a wide range of task characteristics regarding risk definition. These characteristics can be grouped into two main classes: intrinsic and extrinsic.

Intrinsic Characteristics Intrinsic task characteristics are characteristics which naturally describe a task without interference of task doers and regardless of other tasks. These characteristics included task complexity, pre-defined length of the task, task type/nature, and urgency (for example, a task mark as urgent by the assigner). Participants stated that *complex tasks*, or *long-term* tasks are those that are most at risk of never being completed. Especially with exploratory tasks which *don't tie directly back to a business goal* [P11]. This point was further emphasized by explaining that *typically long term or complex tasks can be vague or unstructured in nature. This can be an issue of task quality, in that more details about the tasks need to be gathered before the task can be properly defined* [P128], and that for *long term tasks, when the due date for the task is out, it generally doesn't get prioritized for a while and then has to be done in a hurried manner at the end* [P133]. Participants also stated that tasks of a collaborative nature *may lead to confusion about who is doing what, and if things can't be done in parallel, and tasks are dependent on other tasks, progress may be slowed* [P129]. Some participants indicated that urgency of a task is also another important characteristic defining risk and that these tasks *should be brought to [their] attention as prompt action on [their] part is required* [P80]. Furthermore, participants tied task complexity and nature to their behavior on Planner. For example, many changes in the due date or description of a task could imply that the task turned out to be more complex than expected and took time to flesh out.

Extrinsic Characteristics Extrinsic task characteristics are characteristics that still describe a task but are set and changed by users assigned as the task-doer. These include due date, priority, dependencies and assignees. The majority of participants marked due date as the number one factor related to risk. Participants even highlighted that giving *immediate attention* to *tasks with the closest deadlines also reflects on work aptitude and proficiency* [P36]. When an *upcoming due date* is paired with *high priority* flag set by team or manager, participants thought that tasks become *particularly concerning.* One participant mentioned that *If something was assigned to me (by anyone) and marked as high priority, and had an upcoming due date I'd consider that at risk* [P9]. Participants also mentioned that *complicated dependencies* of a task put this task at the *highest risk*, specifically if paired with *tighter deadlines, multiple teams involved and with high priority* [P15]. If a task has no assignee and the due date is near, this also puts the task at a higher risk as this is the kind of tasks *the team is avoiding* [P40].

Task doers Characteristics

Characteristics of the task-doers or the user-related factors that triggered the user's sense of risk towards a specific task. User-related factors can also be grouped into intrinsic and extrinsic.

Intrinsic Drivers Intrinsic features are focused on user motivations that depend on the user's personal perception of the world without the impact of any external factors. Participants referred to four main user intrinsic motivators: progress, effort, cost of failure and impacting others. Participants believed that *tasks that are left idle are identified as at risk* [P88]. Participants also thought that if slow progress was accompanied by other extrinsic task characteristics such as an upcoming due date and complex dependencies, tasks become more at risk. *High effort* was also correlated to risk by participants. High effort mainly depends on the user and their expertise, therefore, the amount of effort on a task could vary from one user to the other. Participants also related *Impact* to a task at risk. Two types of impact were mainly mentioned: (i) consequences if a task were to fail which are mostly personal, and (ii) impact on other team members such as blocking them on other tasks. In both cases, participants indicated that *negative impact if work is not completed* indirectly implies *high risk* for them.

Extrinsic Drivers Extrinsic user-related features are user motivations that come from external triggers. Participants discussed one main extrinsic feature: important creator or assigner of a task. For example, one participant mentioned that *manager tasks always take priority as completion is a reflection on performance* [P36]. Since the cost of failure in tasks assigned by the managers are higher, any risk of the tasks is also higher.

Due date, progress, task complexity and task priority were the top 4 factors participants referred to or paired together to indicate risk. One participant quoted that 'Typically if the task is very complex, with a close due date and no progress on it. That means the task is at risk. Due to the complexity of the task and its close due date there is a high chance that it will not be completed or will require additional funds or time to complete. Additionally if it is complex that also means that it might have additional dependencies. Also because *I'm not the one working on it there is a reliance on other team members that I might not have visibility on its progress*' [P78]. This mixture of task characteristics and user related factors inspired us to perform quantitative analysis of user log data on four different levels: task level, user level, bucket level and plan level. Our qualitative results greatly shaped our exploration and risk signal search in the dataset in the next steps.

Chapter 4

Quantitative Data Analysis

Informed by the qualitative insights, we study the task at risk definition quantitatively based on user action logs. We analyzed a dataset of anonymized action logs of Microsoft Planner users around the world over a one month period from mid June, 2021 to mid July, 2021. The logs do not contain any textual content of tasks. Microsoft Planner logs span 18 different actions (each action belongs to interacting with a specific task attribute such as description action, priority action, and so on.) Every action has three main kinds of change type: create, modify and delete. Some actions do not have a delete change type such as task title related actions. The actions log contained actions performed on tasks with timestamps and other metadata including type of action, changes based on action, etc. However, the dataset did not contain a risk label. Based on our qualitative analysis, time is the top characteristic related to risk, therefore using lateness of a task as an indication of risk is a good proxy signal. In Microsoft Planner, a task is late if it passes its due date without being marked completed.

4.1 Dataset Filtering

The dataset was filtered to include only tasks with the following criteria: 1) the task had a set due date; 2) the task had creation action in the dataset; 3) the task was completed (had completion action in the dataset); 4) the task had 100 or fewer actions,

5) the task had a lifespan of 30 days or less; These criteria were developed to be able to calculate the ground truth risk label and focus on actionable tasks only that had concrete due dates. Points 4 and 5 were used to exclude data anomalies. Our dataset ended with 250K+ actions with 12K+ unique tasks, 6K+ unique buckets, and 3K+ unique plans. The average number of actions of unique tasks was 19 with a standard deviation of 6.3. The average task lifespan was 3.5 days with a standard deviation of 1.3. The dataset had 2 classes: At Risk = False (tasks that had been completed before due date), At Risk = True (tasks that had been completed after due date.) The dataset was almost balanced among the 2 classes with a class distribution of 45% is at risk is false and 55% is at risk is true.

4.2 Quantitative Exploration

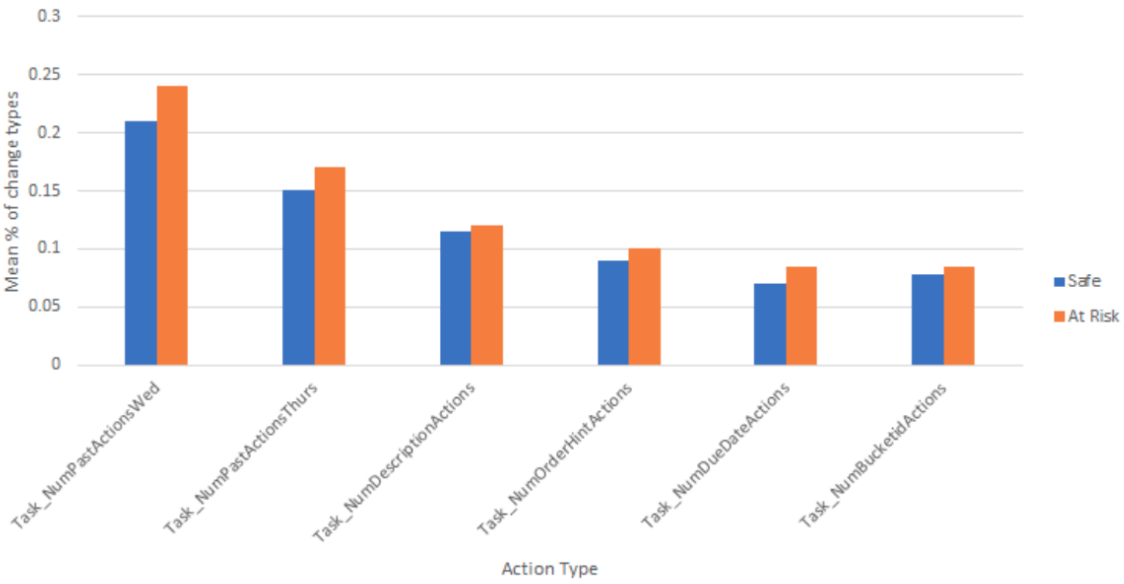


Figure 4-1: Top 6 actions types showing significant percentage differences among risk classes

Initially, we looked for action patterns that could be correlated to task characteristics indicating risk as described by qualitative results. We compared mean percentage of various actions of tasks in both classes (at risk, safe). We also explored this on a more granular level across time meaning we also compared the mean percentage of various actions of tasks per day (for all week days). The resulting plot (Fig. 4-1) highlights the top 6 features showing a significant difference in the percentage of actions between the 2 classes. Tasks at risk showed a higher percentage of description actions, due date actions, task ranking actions, and task bucketizing actions. In our qualitative study, these actions were specifically related to a task being at risk as they could indicate high task complexity (for example if description changes many times this means the task is more complex.) This quantitatively confirms our qualitative study results which shows the potential of using these features as signals for predicting tasks at risk. Furthermore, the percentage of actions performed on Wednesdays and Thursdays were significantly higher in tasks at risk. This could be due to rushing to finish stale or forgotten tasks before an end-of-week deadline or report on Friday. These kinds of tasks are also at higher risk of not being completed successfully.

4.3 Task State Feature Space

Inspired by these quantitative signals, our qualitative study results and previous literature on task classification problems [61, 63, 53], we composed a list of 85 features spanning 4 levels: task level (72), bucket level (4), plan level (4) and user level (5) shown in Table 4.1. Task level features mainly describe the task history until the current state by recording counts of various actions with high granularity on change type and time axes (includes various actions such as description changes. For each action, features include its change type, description deletion count, description creation, and description modification count, etc.) This feature set also includes number of actions per task and per day of week. User level features focus on capturing task management style (number of actions per session, number of different views accessed such as schedule view) and workload (number of completed actions, number of user sessions, etc) of users assigned to the task. Finally, the plan and bucket levels describe the activity and progress inside the plan and bucket respectively. For each action A_i (where A_i is the i^{th} action) on a task T in the dataset, we computed a corresponding task state S_i, describing task history from A_1 to A_i. This resulted in a dataset with the same number of data points (each data point with 85 features) but in a new feature space 'Task State feature space'. We call this dataset 'Main Dataset' for future reference in the paper. Finally, we computed the conditional probability for every feature given task class. We then used these probabilities to compute standard error confidence intervals for all probabilities via bootstrapping, and find that all differences between the conditional probabilities per class are statistically significant with $p < 0.05$. We use this analysis as a validation step that there is a relationship between the computed feature set and class labels and not for feature selection.

Table 4.1: **Extended Feature List composed of 4 levels: task-level (72), bucket-level (4), plan-level (4), and user-level (5)**

Level	Signal	Name	Description
Task-Level	General History	NumPastActions (total, Mon, Tues, Wed, Thurs, Fri, Sat, Sun)	Number of total actions done since task creation till this moment. Computed for each day as well (number of actions happened on a Monday, etc.)
		NumCategoriesActions (total, create, delete)	Number of actions of assigning labels to a task. Computed for total and for each type of action (creation of labels and deletion of labels).
		MeanCatCharLength	Mean of number of characters of applied labels.
		MedianCatCharLength	Median of number of characters of applied labels.
		StDevCatCharLength	Standard deviation of number of characters of applied labels.

Continued on next page

Table 4.1 – continued from previous page

Level	Signal	Name	Description
		NumBucketIdActions	Number of times a task's bucket has been changed.
		NumPreviewTypeActions	Number of times a task has been viewed.
		LastPreviewType	The type of last view of a task (schedule view, user view, plan view, etc.) Presented using one-hot-encoding.
		TopPreviewType	The type of highest view of a task. Presented using one-hot-encoding.
		NumUserSchedulePreview	How many times action doer has opened schedule view.
		NumPlanSchedulePreview	How many times action doer has opened plan view.
		AgeInDays	Number of days since creation of a task.
		NumOrderHintActions (total, create, modify)	Number of times the order of a task has been created or changed inside of its bucket.
		OrderHintValue	The current order of the task inside its bucket.
		NumStartDateActions (total, create, modify)	Number of times start date has been created or changed.
		StartDateValue	Number of days since start time of a task.

Continued on next page

Table 4.1 – continued from previous page

Level	Signal	Name	Description
		NumPriorityActions (total, create, modify)	Number of times priority of a task has been created or changed.
		PriorityValue	The current priority value of a task.
		NumTasksBySameCreator	The number of tasks created by the same task creator.
		AssignedToSelf?	Boolean indicating if the task creator was the same person as the task assignee.
	Complexity	NumUniqueInteractors	Number of unique users that have done any action related to this task.
		NumAssignmentActions (total, create, modify, delete)	Number of times an assignment has been created, modified or deleted.
		NumUniqueAssignees	Number of unique users assigned to a task.
		NumTitleActions (total, create, modify)	Number of times a task title is created or changed.
		TitleCharLength	The current number of characters of the task title.
		NumDescriptionActions (total, create, modify)	Number of times a task description has been created or changed.
		DescriptionCharLength	The current number of characters of description of the task.
		NumDueDateActions (total, create, modify)	The number of times the due date is created or changed.

Continued on next page

Table 4.1 – continued from previous page

Level	Signal	Name	Description
		DueDateValue	The current due date of the task as the number of days left.
		NumReferencesActions (total, create, modify, delete)	Number of times attachments have been added, modified or deleted in the task.
		MeanRefsCharLength	Mean of number of characters of attachments to the task.
		MedianRefsCharLength	Median of number of characters of attachments to the task.
		StDevRefsCharLength	Standard deviation of number of characters of attachments to the task.
		NumConversationActions (total, create, modify)	Number of comments created or changed on the task.
		NumUniqueUsersInConv	Number of unique users involved in the conversation on the task.
	Progress	NumChecklistActions (total, create, modify, delete)	Number of checklist items created, changed or deleted in the task.
		NumCompChecklistActions	Number of 100% completed checklist items in the task.
		NumPercentCompActions (total, create, modify)	Number of times progress on a task has been recorded or changed (0%, 50%, or 100%.)
		PercentCompleteValue	The current progress percentage on the task (0%, 50%, or 100%.)

Continued on next page

Table 4.1 – continued from previous page

Level	Signal	Name	Description
Bucket-Level	General History	NumTasks	Number of other tasks inside the same bucket of this task.
		NumUniqueInteractors	Number of unique users interacting with the bucket of this task.
		NumTitleActions	Number of times title of a bucket has been created or changed.
		NumOrderHintActions	Number of times the order of a bucket has been changed.
Plan-Level	General History	NumTasks	Number of other tasks inside the same plan of this task.
		NumCompleteTasks	Number of 100% completed tasks inside the same plan of this task.
		NumUniqueInteractors	Number of unique users interacting with the plan of this task.
		NumActions	Number of total actions on the plan of this task.
User-Level	Workload	NumCompTasksByUsrInPln	Number of other tasks in this plan 100% completed by assignee of this task.
		NumOfSessionsByUser	Number of times action doer logs in to the system and interacts with this task (a session is time-based and can include many actions).
	Management Style	NumActionsByUsrPerSess	Number of other actions done by the action doer in the session.

Continued on next page

Table 4.1 – continued from previous page

Level	Signal	Name	Description
		NumUsrSchedViewByUsr	Number of times action doer views their personal schedule.
		NumPlanSchedViewByUsr	Number of times action doer views the plan's schedule.

Chapter 5

Task at Risk Prediction

In this chapter, we discuss methods for training machine-learned models to accurately predict tasks at risk from various features guided by our qualitative and quantitative results. We begin by formulating our classification task and then we describe four main experiments (later task stages, sequence modeling, ensemble voter and extended feature set) and their results performed on different subsets of features and model architectures. We describe our experiments in the following subsections. A summary of all experiments and system architecture is shown in Fig. 5-1.

5.0.1 Problem Formulation

The goal of studying this problem is to be able to predict if a task is at risk or not and eventually help users in prioritizing their tasks in teams by bringing tasks at risk to their attention. We formulated this problem as a binary classification problem with an 'At risk' label based on late label.

5.0.2 Data

In our first experiment we used 'Main Dataset' (described in section 4) after excluding user-level, plan-level and bucket-level features. This yielded a dataset with 250K+ data points, each of dimension 72 corresponding to number of task-level features. We call this 'Dataset 1.'

Our second experiment consists of 2 sub-experiments. In the first sub-experiment, we use a modified version of 'Dataset 1' by describing each task as a sequence of states. We describe in detail how we performed this modification in section 5.0.4. We refer to this dataset as 'Dataset 2.1.' We then modify Dataset 2.1 by performing masking on some of the task states for each task (details described in section 5.0.4). We then use a random sample of this modified version as the dataset to perform the second sub-experiment. We refer to this dataset as 'Dataset 2.2' before random sampling for experiment 2.2.

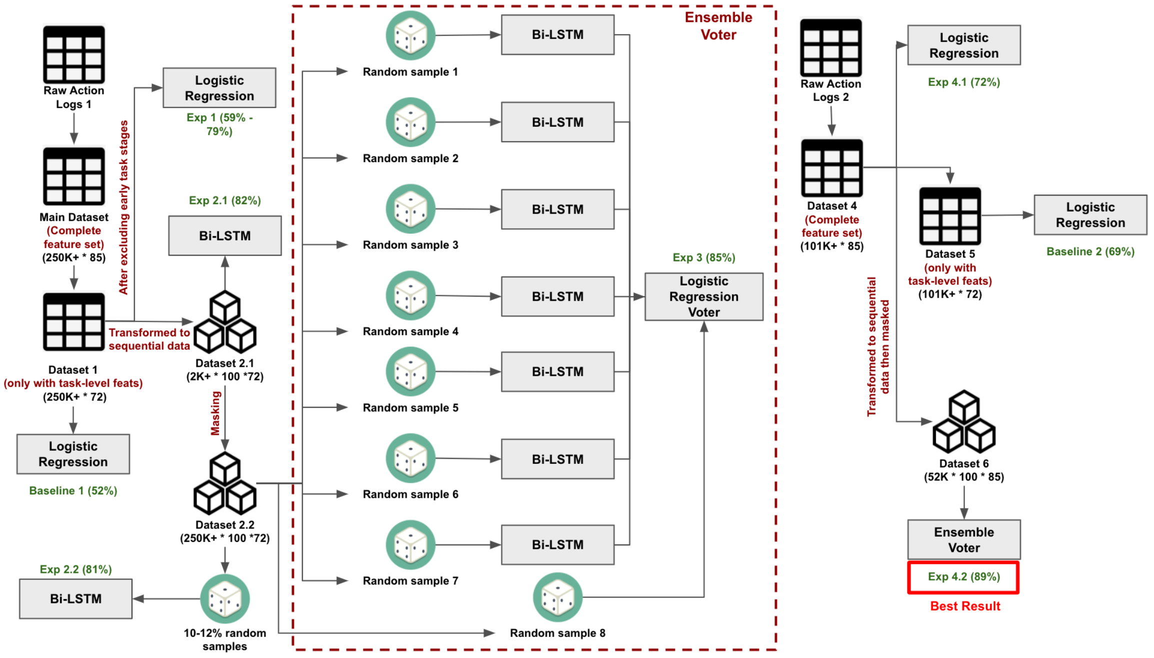
Our third experiment uses multiple random samples from 'Dataset 2.2.'

Due to data retention requirements, we lost access to the 'Main Dataset' which we wanted to use to compare impact of different levels of features on prediction in experiment 4. Therefore, we performed our fourth experiment on a different dataset of the same structure. We followed the same filtering criteria and feature transformation methods as performed on the 'Main Dataset.' We also repeated the same statistical hypothesis test yielding consistent statistically significant results with p ¡ 0.05. We call this dataset 'Dataset 4.' 'Dataset 4' had 101K+ actions with 2K+ unique tasks, 900+ unique buckets, and 400+ unique plans and almost the same number of actions, task lifespan and label class distributions as the 'Main Dataset.' We use 'Dataset 4' in our fourth experiment. The final size of 'Dataset 4' had 101K+ * 85.

5.0.3 Experiment 1: Later Task Stages

We used 'Dataset 1' as is in this experiment. The simplest baseline for this experiment (a simple model always predicting majority class At Risk = True) achieves 55%. We used a logistic regression classifier with a 80%-20% training-testing split of the dataset as our baseline model (We refer to this as 'Baseline 1' experiment.) Our baseline achieved an accuracy of 59%. After performing error analysis, we found that most of the failure cases corresponded to early states of tasks. We then repeated this experiment multiple times with exactly the same initial setup except that every time we exclude earlier task states incrementally. We do this by first excluding all task states that happened on day 0 in a task's life, then we exclude days 0 and 1, and so

Figure 5-1: System Architecture and Experiments Summary



on. All results are shown in Table 5.1. Our results show that as earlier task states are excluded the accuracy increases. This intuitively makes sense as later task states have a richer task history and thus are more descriptive of a task. This indicates that it might be more challenging to predict risk for small task with a short lifespan.

5.0.4 Experiment 2: Sequence Modeling

Each data point in the previously described 'Dataset 1' represents a state of some task at a point in time when some action happened. Alternatively, we could describe a task as a sequence of states with each state representing new actions happening. In this case, a task would be a sequence of all states corresponding to all actions that were performed on this task. Our hypothesis was that representing tasks as sequences of states would capture more information about the development and the future of a task. This would also allow us to leverage the powerful capabilities of deep sequence models. We first transformed 'Dataset 1' used in experiment 1 into a 3D array of dimensions (number of unique tasks * number of task actions * number of task-level features) 12K+ * 100 * 72. We chose 100 to be the maximum threshold of number of

Model Features	Precision		Recall		F-1		Accuracy
	At Risk = False	At Risk = True	At Risk = False	At Risk = True	At Risk = False	At Risk = True	
Task-Level features + LR	0.60	0.59	0.83	0.30	0.69	0.40	0.59
Task-Level features + LR (excluding states of age 0)	0.65	0.71	0.49	0.82	0.56	0.76	0.69
Task-Level features + LR (excluding states of age < 2)	0.61	0.73	0.37	0.88	0.46	0.79	0.70
Task-Level features + LR (excluding states of age < 5)	0.62	0.78	0.37	0.90	0.46	0.84	0.75
Task-Level features + LR (excluding states of age < 10)	0.69	0.81	0.42	0.93	0.52	0.87	0.79

Table 5.1: Results for experiment 1 (Later Task Stages)

actions of any task based on how we filtered the data initially. The majority of tasks had less than 100 actions, however, for tasks with more 100 actions, we used the first 100 actions. Before running our experiment, we masked out all states at or after task completion in a task sequence to prevent the model from cheating (if completion date is less than due date, then risk = false) and making decisions without learning any patterns from the data. We refer to this dataset as ‘Dataset 2.1.’ We then fed the ‘Dataset 2.1’ to a bidirectional LSTM model with 80%-20% as training-testing split ratio. This model achieved 82% accuracy (experiment 2.1.) Even though this model improved on the baseline, it is based on getting almost all sequence of states of a task. This is not practical if early intervention is aimed for, which is very crucial to a task being at risk.

To solve this, we modified ‘Dataset 2.1’ by masking task states in steps of 5 starting from the last task state moving towards the beginning. All masked task states of a task are added to the dataset as new distinct data points. A step of 5 was selected as the average number of actions per user session was 4.2 and at creation approximately 5 actions happen. This approach takes various task stages into account which is more realistic. This resulted in a larger dataset with shape 250K * 100 * 72, which refer to as 'Dataset 2.2'. Multiple random samples of 10%-12% size were drawn from 'Dataset 2.2' to perform the experiment many times. All drawn samples had an almost balanced class distribution of 55%+-2% (risk = false): 45%+-2% (risk = true). Each sample was then fed to the same model with the same experiment settings as the previous experiment. This experiment achieved an average of 81% accuracy (experiment 2.2.) This result indicates that the model was able to learn a pattern from the sequential data by learning the changes that happen to a task across its stages and the relationship of these changes to risk.

5.0.5 Experiment 3: Ensemble Voter

To improve on the results of the previous experiment, we decided to build an ensemble voter by training the same model type with random sub-samples of the data to leverage the multiple drawn random samples in voting rather than averaging the final accuracy, which is proven to yield better results [16]. We drew 8 random samples from 'Dataset 2.2' following the same approach as experiment 2.2 while also making sure all samples are mutually exclusive. We used the first 7 samples to independently train 7 bidirectional LSTM models with the same architecture as previous experiments. We then used the 8^{th} dataset to train a Logistic Regression voter. All training followed an 80%-20% training-testing split ratio. The results of this experiment are shown in Table 5.2. This model achieved the highest accuracy of 85% with high precision, recall and F-1 scores among our experiments so far which reflects the strength of ensemble models in achieving better results.

5.0.6 Experiment 4: Extended Feature Set

In this experiment, we were interested in comparing the extended feature set (including task, user, bucket and plan level features) with the task-level feature set used in all previous experiments. For a valid comparison, we retrained a logistic regression

Experiment	Precision		Recall		F-1		Accuracy
	At Risk = False	At Risk = True	At Risk = False	At Risk = True	At Risk = False	At Risk = True	
Baseline 1	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0.59
Experiment 2.1	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0.82
Experiment 2.2	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0.81
Experiment 3	0.86	0.85	0.79	0.90	0.82	0.87	0.85
Baseline 2	0.68	0.69	0.33	0.91	0.44	0.78	0.69
Experiment 4.1	0.73	0.71	0.39	0.91	0.51	0.80	0.72
Experiment 4.2	**0.90**	**0.89**	**0.91**	**0.88**	**0.90**	**0.88**	**0.89**

Table 5.2: Results for all experiments

baseline with ‘Dataset 5’ which is ‘Dataset 4’ after excluding user-level, plan-level and bucket-level features (size of 101K+ * 72.) We call this ‘Baseline 2’ experiment. We then re-ran the baseline model on all features in ‘Dataset 4’ giving a dataset of size 101K+ * 85 (experiment 4.1). ‘Dataset 4’ was then transformed to the 3D shape and masked using the previous methods in experiment 3 for re-running the ensemble voter model on the extended feature set. The final dataset size after masking was 52K * 100 * 85 (‘Dataset 6.’) Finally, we repeated the same ensemble voter experiment again with ‘Dataset 6.’ We call this ‘experiment 4.2.’ All sub-experiments were done exactly as the ones described in experiments 1, 2 and 3. All results are shown in Table 5.2. This experiment shows that adding more signals to the feature set to indicate the user’s workload and management style as well as ongoing activity in the same plan and bucket is very effective as it reflects a better idea of other external factors indicating whether a task is at risk.

Our results demonstrate that tasks at risk can be predicted with high accuracy. Our best model uses all 4 levels of features: task, user, plan and bucket with ensemble voting of bidirectional LSTM models. Our best model outperforms existing solutions in literature focused on similarly formulated classification problems with accuracy of 89% [61, 63, 53, 54].

Chapter 6

Discussion, Limitations and Implications

Our results show that task management in teams is divided into 2 main steps: individual task prioritization and breakdown, and team coordination. Most of the time there is a disconnect between these 2 aspects which leads to more challenging task prioritization and tracking. This overwhelms users and usually leads to some tasks slipping off their mind. Our study shows that there is a considerable need for predicting tasks at risk that are prone to failure due to planning fallacies, task complexity, huge workloads, lack of attention, or much more. We also developed a task at risk definition and representation based on our qualitative study which was confirmed by analyzing a large-scale action logs dataset of Microsoft Planner users. This definition is mainly based on various task attributes and facets, and user activity. We try various experimental setups to predict whether a task is at risk and we outperform our baseline with 89% accuracy. We explored the impact of excluding early task stages, sequence modeling of tasks, ensemble voting of multiple randomly sampled subsets of data, and various subsets of features on the prediction task. Our best model uses ensemble voting of multiple sequence models with the a full feature set containing task level, user-level, plan-level and bucket-level features. We believe that applications to predicting tasks at risk in digital assistance and task management tools would bridge the gap between managing personal task list and team task planning and coordination.

6.1 Generalizability of Methods

In our work, we focus on Microsoft Planner as it is simple and intuitive and we had access to a large action logs dataset of real task management projects. We argue that our hypotheses about task management and task at risk definition are generalizable as we explored the problem qualitatively with a reasonable amount of participants (151) and quantitatively by analyzing a large and diverse dataset of actual projects. Furthermore, our dataset analysis confirmed our qualitative results and aligned with some prior work. Our feature sets and prediction models are also interpretable and can be applied to any task management tool with the most basic components: a task and a user.

6.2 Entity Dependency

Our experiments focus on historical features of 4 main entities: tasks, users, plans and buckets. However, dependencies and relations between these different entities were not leveraged to capture a more accurate situation. We believe that capturing these relations via heterogeneous graph neural networks could further improve the prediction problem. For example, if a task is blocked or is a blocker to other tasks or team mates, then this puts the task at higher risk as described by many participants in our qualitative study 'If I'm not making progress on a task it may be an indication there's an external blocker to making progress which makes it likely to be at risk. Collaborative tasks may lead to confusion about who is doing what, and if tasks are dependent on other tasks, progress may be slowed.' [P29]

6.3 User Workload and Management Style

Even though we used good proxies for approximating a user's workload and task management style, we believe that more accurate representations could be obtained from the user's calendar and email data. Calendar and email data are one of the best pointers to a user's workload [53] as workflows heavily depend on these tools. Due to the lack of access to such data, we were not able to experiment with more accurate proxies.

6.4 Risk Labeling

Based on our qualitative study, time (represented with due dates) is the top feature participants attributed to risk. Therefore, we used 'Lateness' as a proxy for automatically labeling tasks at risk. Solutions such as crowd-sourced data labeling were considered before choosing the labeling method, however, we believe that having someone label a task that they never worked on or were involved with would yield very inaccurate labels. Risk of a task depends on many dynamic attributes part of which depend totally on the assignee. Therefore, we suggest that having the original task doers label the tasks together with presenting their knowledge and expertise is the most accurate solution for a model to learn. This also works around the problem of task management style [63], when some users mark tasks complete in batches after the due date. In our work, we assumed that users mark tasks complete as soon as they complete them based on our qualitative study results. This contradicts previous findings on individual task management style [63] but reflects that individuals behave differently in teams, where they want to immediately reflect progress and performance.

6.5 Textual Analysis

Our previously described experiments were performed using datasets of anonymized timestamped action logs of Microsoft Planner with no textual data. Believing that text incorporates rich information that could improve the performance of our prediction model [61], we collected the textual data (donated by Microsoft Employees) of a small subset of our main actions log dataset. The new joint dataset contained 21k data points of both textual and action logs data of tasks. We first performed data processing to textual data (fixing contractions, removing stop words, lemmatization, lowercasing all text, etc.) Then, we computed and analyzed TF-IDF weighted token n-grams (1-3). Our analysis showed some difference in word occurrence between late and non-late tasks. For example, the occurrence of words that indicate vagueness of the nature of the task such as ‘review’ or hyperlinks is higher in Late tasks, which aligns with previous qualitative findings. On the other hand, tasks that involve ‘customers’ tend to be non-late. This could be due to the skewed nature of the donated dataset as it was collected from a small number of teams inside of Microsoft that deal with and prioritize tasks involving customers. We believe that having a fuller set of textual data of tasks paired with action logs could yield very interesting insights about the nature of tasks and more accurate predictions of risk.

Chapter 7

Conclusion

Task prioritization is an important and perhaps the most challenging aspect of collaborative task management. Ineffective prioritization puts tasks at a risk of failure and could eventually lead to personal or team losses if work is not done. We perform a 3-stage study (qualitative, quantitative and ML modeling) to identify and predict tasks at risk. Our best model suggests that tasks at risk can be identified with high accuracy (89%). This has several implications on improving task prioritization in digital assistants and task management tools. Further work could be done to improve our study by using textual content of tasks along with action logs. Task interdependence could also be explored and modeled using more complex models such as Graph Neural Networks.

Bibliography

[1] David Allen. *Getting things done: The art of stress-free productivity*. Penguin, 2015.

[2] Teresa M Amabile, Chelley Patterson, Jennifer Mueller, Tom Wojcik, Paul W Odomirok, Mel Marsh, and Steven J Kramer. Academic-practitioner collaboration in management research: A case of cross-profession collaboration. *Academy of Management Journal*, 44(2):418–431, 2001.

[3] John D Aram and Cyril P Morgan. The role of project team collaboration in r&d performance. *Management Science*, 22(10):1127–1137, 1976.

[4] Anne Aula, Rehan M Khan, and Zhiwei Guan. How does search behavior change as search becomes more difficult? In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 35–44, 2010.

[5] Victoria Bellotti and Sara Bly. Walking away from the desktop computer: distributed collaboration and mobility in a product design team. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 209–218, 1996.

[6] Ann E Blandford and Thomas RG Green. Group and individual time management tools: what you get is not what you need. *Personal and Ubiquitous Computing*, 5(4):213–230, 2001.

[7] Roger Buehler, Dale Griffin, and Michael Ross. It's about time: Optimistic predictions in work and love. *European review of social psychology*, 6(1):1–32, 1995.

[8] Katriina Byström and Kalervo Järvelin. Task complexity affects information seeking and use. *Information processing & management*, 31(2):191–213, 1995.

[9] Carrie J Cai, Philip J Guo, James R Glass, and Robert C Miller. Wait-learning: Leveraging wait time for second language education. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3701–3710, 2015.

[10] Kathy Charmaz, LL Belgrave, and G Ritzer. The blackwell encyclopedia of sociology, 2007.

[11] Dimitra Chasanidou, Brian Elvesæter, and Arne-Jørgen Berre. Enabling team collaboration with task management tools. In *Proceedings of the 12th International Symposium on Open Collaboration*, pages 1–9, 2016.

[12] Justin Cheng, Jaime Teevan, Shamsi T Iqbal, and Michael S Bernstein. Break it down: A comparison of macro-and microtasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 4061–4064, 2015.

[13] Justin Cranshaw, Emad Elwany, Todd Newman, Rafal Kocielnik, Bowen Yu, Sandeep Soni, Jaime Teevan, and Andrés Monroy-Hernández. Calendar. help: Designing a workflow-based scheduling agent with humans in the loop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2382–2393, 2017.

[14] John W Creswell, VL Plano Clark, Michelle L Gutmann, and William E Hanson. An expanded typology for classifying mixed methods research into designs. *A. Tashakkori y C. Teddlie, Handbook of mixed methods in social and behavioral research*, pages 209–240, 2003.

[15] Robert C Daley. The role of team and task characteristics in r&d team collaborative problem solving and productivity. *Management Science*, 24(15):1579–1588, 1978.

[16] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[17] Martin J Eppler and Oliver Sukowski. Managing team knowledge: core processes, tools and enabling factors. *European Management Journal*, 18(3):334–341, 2000.

[18] Darryl K Forsyth and Christopher DB Burt. Allocating time to future tasks: The effect of task segmentation on planning fallacy bias. *Memory & cognition*, 36(4):791–798, 2008.

[19] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. A taxonomy of microtasks on the web. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 218–223, 2014.

[20] David Graus, Paul N Bennett, Ryen W White, and Eric Horvitz. Analyzing and predicting task reminders. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 7–15, 2016.

[21] Andrey Gusev, Nathanael Chambers, Divye Raj Khilnani, Pranav Khaitan, Steven Bethard, and Dan Jurafsky. Using query patterns to learn the duration of events. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, 2011.

[22] Nathan Hahn, Shamsi T Iqbal, and Jaime Teevan. Casual microtasking: Embedding microtasks in facebook. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–9, 2019.

[23] Guido Hertel, Susanne Geister, and Udo Konradt. Managing virtual teams: A review of current empirical research. *Human resource management review*, 15(1):69–95, 2005.

[24] Judith A Holton. Building trust and collaboration in a virtual team. *Team performance management: an international journal*, 2001.

[25] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166, 1999.

[26] Jiang Hu and Mike Brzozowski. Preference-based group scheduling. In *IFIP Conference on Human-Computer Interaction*, pages 990–993. Springer, 2005.

[27] Shamsi T Iqbal, Jaime Teevan, Dan Liebling, and Anne Loomis Thompson. Multitasking with play write, a mobile microproductivity writing tool. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 411–422, 2018.

[28] Robert A Josephs and Eugene D Hahn. Bias and accuracy in estimates of task duration. *Organizational Behavior and Human Decision Processes*, 61(2):202–213, 1995.

[29] Daniel Kahneman and Amos Tversky. Intuitive prediction: Biases and corrective procedures. Technical report, Decisions and Designs Inc Mclean Va, 1977.

[30] Bumsoo Kang, Chulhong Min, Wonjung Kim, Inseok Hwang, Chunjong Park, Seungchul Lee, Sung-Ju Lee, and Junehwa Song. Zaturi: We put together the 25th hour for you. create a book for your baby. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 1850–1863, 2017.

[31] Jeonghyun Kim. Task difficulty as a predictor and indicator of web searching interaction. In *CHI'06 extended abstracts on human factors in computing systems*, pages 959–964, 2006.

[32] Cornelius J König. Anchors distort estimates of expected duration. *Psychological Reports*, 96(2):253–256, 2005.

[33] Zornitsa Kozareva and Eduard Hovy. Learning temporal information for states and events. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 424–429. IEEE, 2011.

[34] Justin Kruger and Matt Evans. If you don't want to be late, enumerate: Unpacking reduces the planning fallacy. *Journal of Experimental Social Psychology*, 40(5):586–598, 2004.

[35] Filippo Lanubile, Christof Ebert, Rafael Prikladnicki, and Aurora Vizcaíno. Collaboration tools for global software engineering. *IEEE software*, 27(2):52–55, 2010.

[36] Yuelin Li and Nicholas J Belkin. A faceted approach to conceptualizing tasks in information seeking. *Information processing & management*, 44(6):1822–1837, 2008.

[37] Jeanne M Liedtka. Collaborating across lines of business for competitive advantage. *Academy of management perspectives*, 10(2):20–34, 1996.

[38] Chang Liu, Jingjing Liu, and Nicholas J Belkin. Predicting search task difficulty at different search stages. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 569–578, 2014.

[39] Jingjing Liu and Nicholas J Belkin. Personalizing information retrieval for multisession tasks: The roles of task stage and task type. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 26–33, 2010.

[40] Jingjing Liu, Chang Liu, Michael Cole, Nicholas J Belkin, and Xiangmin Zhang. Exploring and predicting search task difficulty. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1313–1322, 2012.

[41] Peng Liu and Zhizhong Li. Task complexity: A review and conceptualization framework. *International Journal of Industrial Ergonomics*, 42(6):553–568, 2012.

[42] SJ Mantel, JR Meredith, SM Shafer, and MM Sutton. United States: John Wiley & Sons. Inc, 2011.

[43] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. Reliability and interrater reliability in qualitative research: Norms and guidelines for cscw and hci practice. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–23, 2019.

[44] Nina Mishra, Ryen W White, Samuel Ieong, and Eric Horvitz. Time-critical search. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 747–756, 2014.

[45] Don A Moore and Paul J Healy. The trouble with overconfidence. *Psychological review*, 115(2):502, 2008.

[46] Karen Myers, Pauline Berry, Jim Blythe, Ken Conley, Melinda Gervasio, Deborah L McGuinness, David Morley, Avi Pfeffer, Martha Pollack, and Milind Tambe. An intelligent personal assistant for task and time management. *AI Magazine*, 28(2):47–47, 2007.

[47] Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. Torque: A reading comprehension dataset of temporal ordering questions. *arXiv preprint arXiv:2005.00242*, 2020.

[48] Rosalie J Ocker and Gayle J Yaverbaum. Asynchronous computer-mediated communication versus face-to-face collaboration: Results on student learning, quality and satisfaction. *Group Decision and Negotiation*, 8(5):427–440, 1999.

[49] Feng Pan, Rutu Mulkar-Mehta, and Jerry R Hobbs. Annotating and learning event durations in text. *Computational Linguistics*, 37(4):727–752, 2011.

[50] Mark V Pezzo, Jordan A Litman, and Stephanie P Pezzo. On the distinction between yuppies and hippies: Individual differences in prediction biases for planning future tasks. *Personality and individual differences*, 41(7):1359–1371, 2006.

[51] Ioannis Refanidis and Neil Yorke-Smith. A constraint-based approach to scheduling an individual's activities. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 1(2):1–32, 2010.

[52] Xin Rong, Adam Fourney, Robin N Brewer, Meredith Ringel Morris, and Paul N Bennett. Managing uncertainty in time expressions for virtual assistants. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 568–579, 2017.

[53] Bahareh Sarrafzadeh, Ahmed Hassan Awadallah, Christopher H Lin, Chia-Jung Lee, Milad Shokouhi, and Susan T Dumais. Characterizing and predicting email deferral behavior. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 627–635, 2019.

[54] Bahareh Sarrafzadeh, Sujay Kumar Jauhar, Michael Gamon, Edward Lank, and Ryen W White. Characterizing stage-aware writing assistance for collaborative document authoring. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–29, 2021.

[55] Meera Sharma, Punam Bedi, KK Chaturvedi, and VB Singh. Predicting the priority of a reported bug using machine learning techniques and cross project validation. In *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 539–545. IEEE, 2012.

[56] O Submitters. Essence–kernel and language for software engineering methods. 2012.

[57] Jaime Teevan, Daniel J Liebling, and Walter S Lasecki. Selfsourcing personal tasks. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 2527–2532. 2014.

[58] Jeremy I Tutty and James D Klein. Computer-mediated instruction: A comparison of online and face-to-face collaboration. *Educational technology research and development*, 56(2):101–124, 2008.

[59] Judith Davidson Wasser and Liora Bresler. Working in the interpretive zone: Conceptualizing collaboration in qualitative research teams. *Educational researcher*, 25(5):5–15, 1996.

[60] Diana White and Joyce Fortune. Current practice in project management—an empirical study. *International journal of project management*, 20(1):1–11, 2002.

[61] Ryen W White and Ahmed Hassan Awadallah. Task duration estimation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 636–644, 2019.

[62] Ryen W White, Ahmed Hassan Awadallah, and Robert Sim. Task completion detection: A study in the context of intelligent systems. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 405–414, 2019.

[63] Ryen W White, Elnaz Nouri, James Woffinden-Luey, Mark Encarnacion, and Sujay Kumar Jauhar. Microtask detection. *ACM Transactions on Information Systems (TOIS)*, 39(2):1–29, 2021.

[64] Alex C Williams, Harmanpreet Kaur, Shamsi Iqbal, Ryen W White, Jaime Teevan, and Adam Fourney. Mercury: Empowering programmers' mobile work practices with microproductivity. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 81–94, 2019.

[65] Chuxu Zhang, Julia Kiseleva, Sujay Kumar Jauhar, and Ryen W White. Grounded task prioritization with context-aware sequential ranking. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–28, 2021.

[66] Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. " going on a vacation" takes longer than" going for a walk": A study of temporal commonsense understanding. *arXiv preprint arXiv:1909.03065*, 2019.

[67] Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. Temporal common sense acquisition with minimal supervision. *arXiv preprint arXiv:2005.04304*, 2020.