Building The Ultimate Teams With Psychology and AI

Creating a SVM machine learning model that makes team recommendation based on employees' dynamic traits using data from Slack and Task Manager API

BenchSci x TKS

Creating a roster of individuals based on dynamic traits to make the most efficient teams

Executive Summary

Problem

Benchsci follows the **conventional method** for team selection (i.e. having dedicated TAs and and managers that select team members who are the best fit based on the requirements of the project). It is a serious gap when it comes to project management in MultiDisciplinary Teams (MDTs), Cross-Functional Teams (CFTs), and even Functional Teams (FTs) because t**eam compatibility based on dynamic traits is not the key factor for team selection**. Moreover, BenchSci has grown from 200 to 400+ employees who are in different geographical locations and time zones. The rapid pace of scaling up makes it difficult for management to be aware of the traits of their fellow workers and assign employees to a team to improve effectiveness manually.

Solution

Benchbuddy (our solution) fetches data from slack API and is able to filter teammates relevant to project description and recommend best teams using calculated trait scores via ML techniques**.** The input data is taken from a shared Slack workspace by creating a SlackBot token that has certain permissions to view and read channels. "**Find Possible Teammates**" featured is performed by **BERT model** to compute similarities between a given project description and users' introduction messages . By using **DistilBERT mode**l for sentiment analysis , **user traits such as participation, work done, compatibility, and adaptability are calculated**. Lastly, **K-Means clustering** to **group users based on their trait scores** and **recommends the best teammates**.

Impact

By implementing BenchBuddy, Benchsci would have **more efficient teams** which will **accelerate product development** and **boost revenue**, thus giving it an **edge over its competitors**.

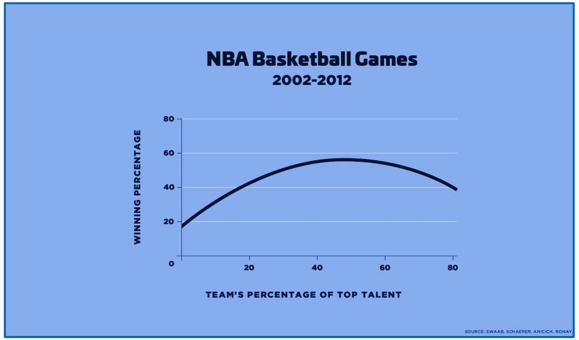
An automated system can provide a **bias-less evaluation of the skills and experience of potential team members**. It may **increase transparency** and help managers make more informed decisions in regards to team composition.

Employees will enjoy working with team members that compliment each other resulting in **increased employee satisfaction levels**.

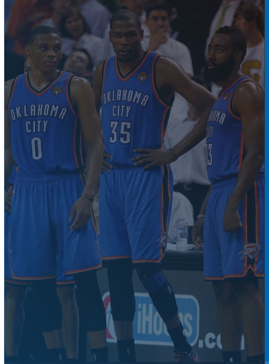
Super-Teams Underperform

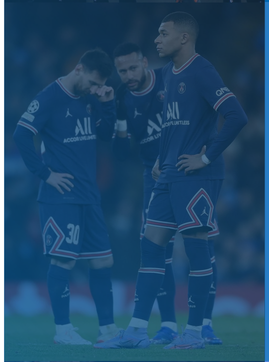
Many organisations employ highly intelligent, qualified people on the assumption that they will automatically combine their collective brainpower to produce magical results. Yet such groups often fail to cash in on their talents with poor creativity, lost efficiency and sometimes overly risky decision making. On Wall Street, teams with mostly top analysts make **worse financial recommendations** than teams that have a mix of stars and average performers. The same is true in soccer: national teams with too many top players are less likely to win World Cup qualifying matches.

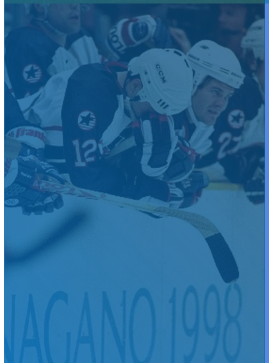
In a study of NBA basketball over a decade, teams with only three star players **won more games** than teams with four or five. The star-studded teams had fewer assists, missed more of their shots and grabbed fewer rebounds. The players struggled to coordinate. The Miami Heat faced the same problem when they introduce 2 new stars players, LeBron James and Chris Bosh, into their team which already had Dwyane Wade as its star. Contrary to their expectations of winning championships for years, they struggled to even win close games in the regular seasons, now having three players who were all used to taking the game-winning shot.



According to the study of NBA basketball over a decade, winning percentage of the team exceeds with increase in talent at a decreasing rate. If the percentage of teams top talent **exceeds 50%**, the chances of winning decrease. In other terms, teams with only three star players won more games than teams with four or five.







The Psychology Behind Why Super-teams Underperform

A study by the University of California, Berkeley looked at data from over 1,000 athletes who competed in the NCAA Division I Men's Basketball Tournament from 1985 to 2012 and found that athletes who scored high on the California Psychological Inventory (CPI) test on the scales of Dominance, Capacity for Status, Socialization, and Responsibility were more likely to succeed at the highest levels of competition. While super-teams may have the best individual players in a given sport, they may not have the right mix of personality traits to succeed.

1 LACK OF TRUST

All-star teams are made up of stellar performers with enormous egos, which can translate into a lack of trust. Superstars like the limelight; they may not easily trust others or be trusted themselves. Without trust and respect, a team lacks a solid foundation.

2 ABSENCE OF TEAM CHEMISTRY

A successful team needs to have the right players in the right positions. Star talents often struggle to take on roles that can help the team most, preferring instead to go for the spotlight. Team members need to get to know each other professionally and personally in order to form trust and comradeship that allows for better delegation of tasks and getting tasks done efficiently.

3 POOR DECISION MAKING

Oftentimes groups fail to agree on decisions, reaching an impasse, or overly complicating a problem by incorporating all viewpoints – all of this can be extremely damaging for a team's productivity. On the flip side, desire to reach a decision in a short time frame can cloud judgement and lead to faulty decisions.

Compatibility: The Key To A Team's Success

What a good team looks like

A good team doesn't necessarily comprise of individuals who are the most talented or have same fixed traits, but rather a combination of people with dynamic personalities that align and who are compatible as a group. This allows each member to participate equally, instead of having one or two people dominate. Irrespective of their compromised technical depth, their chemistry and alignment makes them greater than the sum of their parts.

How Google & Apple optimize for team formation

Companies like Apple, Netflix, Google, and Dell are **40% more productive** than the average company, according to research from the leadership consulting firm Bain & Company. However, it's not because these companies attract top-tier employees – high performers who are naturally gifted at productivity, but rather how they delegate these star talents across teams evenly to achieve considerable success.

"Our research found that these companies (Google and Apple) have 16% star players, while other companies have 15%. They start with about the same mix of star players, but they are able to produce dramatically more output. They select a handful of roles that are business critical, affecting the success of the company's strategy and execution, and they fill 95% of these roles with A-level quality. The rest of the roles have fewer star players."

– Michael Mankins*, partner and leader at strategy practices at Bain & Company*

Example: iOS 10 by Apple

An example of how this plays out is Apple and Microsoft in the early 2000s. It took 600 Apple engineers less than two years to develop, debug, and deploy iOS 10. Contrast that with 10,000 engineers at Microsoft that took more than five years to develop, debut, and ultimately retract Vista. The difference is in the way these companies chose to construct their teams.

Since iOS 10 was a mission critical initiative, Apple selected a team of highly exceptional performers and they were able to make it work because their values and traits aligned.

"If given a choice between choosing the serious-minded Team A or the free-flowing Team B, you should probably opt for Team B. Team A may be filled with smart people, all optimized for peak individual efficiency. But the group's norms discourage equal speaking; there are few exchanges of the kind of personal information that lets teammates pick up on what people are feeling or leaving unsaid. There's a good chance the members of Team A will continue to act like individuals once they come together, and there's little to suggest that, as a group, they will become more collectively intelligent. In contrast, on Team B, people may speak over one another, go on tangents and socialize instead of remaining focused on the agenda. The team may seem inefficient to a casual observer. But all the team members speak as much as they need to. They are sensitive to one another's moods and share personal stories and emotions. While Team B might not contain as many individual stars, the sum will be greater than its parts.

"We had lots of data but there was nothing showing that a mix of specific personality types or skills or backgrounds made any difference. The " who " part of the equation did not seem to matter."

—Anita Woolley, part of Google's Aristotle project"

Psychological safety as a driving factor for team success

GOOGLE'S ARISTOTLE STUDY

Google's Aristotle study created a five-hour battery of tasks that together tested four different kinds of thinking: generating new ideas, choosing a solution based on sound judgement, negotiating to reach compromise, and finally, general ability at task execution (such as coordinating movements and activities).

According to their finding, the good teams shared two behaviours. First, all members spoke in roughly the same proportion, a phenomenon the researchers referred to as ''equality in distribution of conversational turn-taking.'' On some teams, everyone spoke during each task; on others, leadership shifted among teammates from assignment to assignment. But in each case, by the end of the day, everyone had spoken roughly the same amount. Second, the good teams all had high 'average social sensitivity' - they were skilled at intuiting how others felt based on their tone of voice, their expressions and other nonverbal cues. People on the more successful teams in Woolley's experiment scored above average on the Reading the Mind in the Eyes test, which is designed to gauge social sensitivity by viewing people's eyes.

The Five Keys to Effective Team Dynamics

How these practices lead to success

Executives from large companies across 12 industry sectors worldwide said three components of human capital impact productivity more than anything else: time, talent, and energy. And the top quartile have efficient teams that do their business in a way that they’re **40% more productive** than the rest and consequently have **profit margins that are 30%-50% higher than industry averages**.

They’re able to do more work everyday, and this difference compounds yearly; over a decade, they can **produce 30 times more than the rest, with the same number of employees**.

The members of these teams have chemistry, feel motivated, and are genuinely interested in doing their tasks. Dell Technologies recognized the productivity difference between inspired and average teams and found that **sales teams led by an inspiring leader are 6% more productive** than those that have an average leader. If you extrapolate from that 6%, it **accounts for an extra $1 billion in annual revenue**.

Implementing Team Compatibility In Smaller Companies

Why this hasn't trickled down to smaller companies

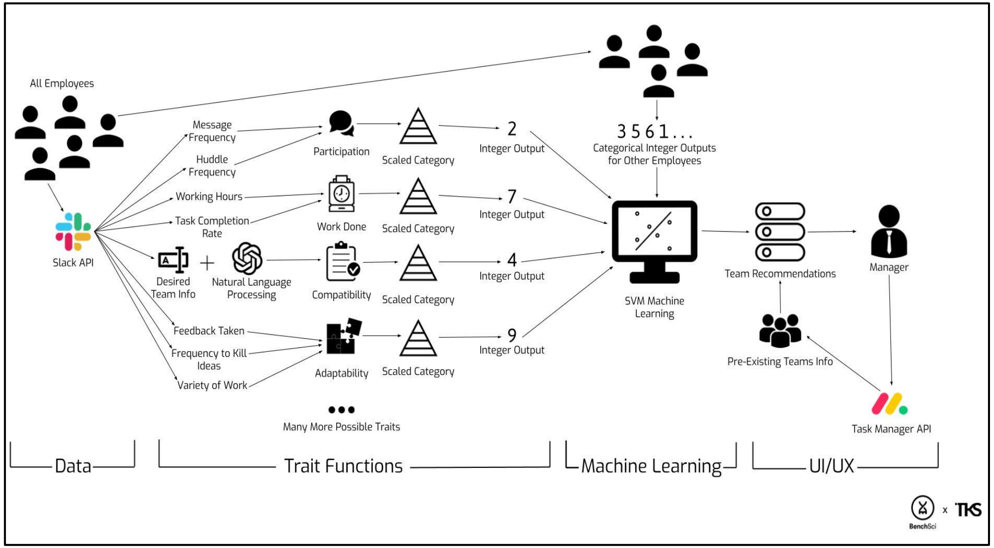
Smaller companies aren't implementing this system of creating teams based on dynamic trait compatibility. Often due to a rapid scale up, they don't have enough time to build and optimize such systems, and don't consider it worth investing with their limited resources. Moreover, small businesses in specific industries or niches may require employees with very specific skills and expertise. As a result, they may prioritize hiring for specialized roles over creating teams with diverse traits. Their operations are often smaller and simpler than giants like Google, and they may prefer a structured approach to team composition.

How to bridge the gap

In order to be more successful, small companies need to make team compatibility more of a priority. They can invest in team compatibility assessment tools which are able to assess team members' skills, personalities, and values, and recommend the best teams for the company's project in a short span of time. It's also important to give team members the opportunity to get to know each other in-person by organising group hangouts and 1:1s on regular basis and then analyse these interactions for team formation.

Solution Overview: BenchBuddy

Using machine learning models to make team recommendations based on employees' dynamic traits using data from Slack and Task Manager API



1 **Get user data from Slack API:** The input data for BenchBuddy is fetched from Slack API using a token which allows it to access and read every channel. A new channel called #introductions will be created by Benchsci where employees will introduce themselves, describe their skills, experiences, and areas of interest. The data from this channel will be used for filtering teammates and data from other channels that represent the general chat history of users will be used for trait core calculation.

2 **Filtering Teammates:** The user will first enter the brief description of the project in the "Project Description" text area **and then click on** "Find Possible Teammates" button. On the backend, **BERT model** compares project description with the introduction messages of all Slack users. Users whose introductions align closely with the project are shortlisted based on a similarity threshold. The filtered teammates are saved in a CSV name 'possible_teammates.csv'

3 **Calculating Trait Score:** The user will now click the "Calculate Trait Scores" button. The **DistilBERT model** analyzes the general chat messages of the shortlisted users. Based on the sentiments of these messages, trait scores (**participation**, **work done**, **compatibility**, **adaptability**) are calculated for each user. These trait scores are saved in 'output_trait_scores.csv'.

4 **Recommending Best Teams:** Lastly the user will click the "Select Best Team" button. The application will use **K-Means clustering** on the trait scores to group similar teammates. It will recommend a team based on the proximity of cluster centroids to the global average of traits, ensuring a balanced team.

Ensuring Data Privacy

How is data gathered? The input data for BenchBuddy is taken from a shared Slack workspace. This is implemented by creating a SlackBot token that has certain permissions to view and read channels. The data taken from the workspace would be in two forms: **Background** and **General message history**.

Background: As a team selector for new projects, BenchBuddy needs data on an employee's background - their "fixed traits", such as what they specialize in, what their degree is in, the projects they have worked on in the past or are often found working on, to ensure that not only are all the teammates generally compatible - but that they fit the project's requirements. For this, we suggest the creation of a new Slack channel in the BenchSci workspace, called #introductions. In this channel, employees would introduce themselves, describe their skills, experiences, and areas of interest. These introductions give a snapshot of what each member brings to the table without the need for a more tedious data collection method. A future feature could be the supplementation of other manually inputted information, such as their previous projects at BenchSci, to ensure an up-to-date and wholistic purview of a person's interests and specialities

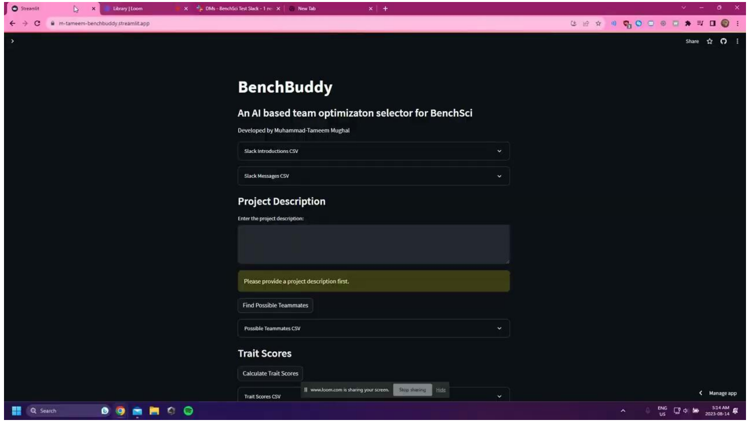
General message history: However, for the core part of BenchBuddy, which involves analyzing user's personality traits, the Slack API would be called to return the publicly viewable Slack Messages on a user by user basis, creating a dataset of a BenchSci employee and their message history. This message history would provide insights into a user's personality traits - such as participation levels, work contributions, and adaptability, which can be analyzed and used to determine a best fit for a team.

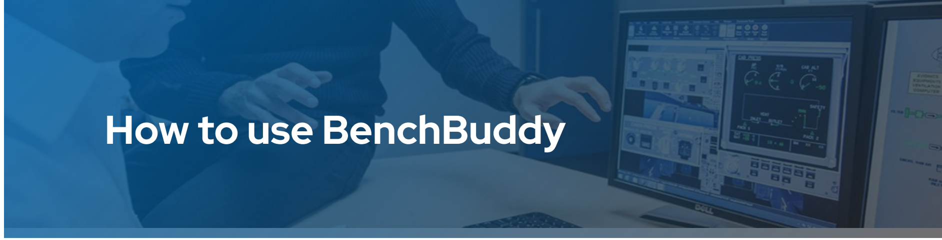
Ethics: We understand that integrity, inclusion, privacy, and respect are all core beliefs at BenchSci, and that BenchSci heavily emphasizes on treating their employees with respect. Because of that, though already taken from "public" channels or private channels related to projects only, BenchBuddy's Slack API permission can be altered within Slack to solely be able to view and return messages from work-related channels, and not channels used for mere conversation, leisure, or otherwise. Channels for minorities, such as pride and other ethnic channels could be completely disregarded, to ensure that no bias inherent to the LLM is amplified, and to ensure that all data collected is purely necessary information regarding work and not related to the employee's personal lifestyle.

BenchBuddy: The Prototype

The Prototype: We've created a functional prototype of BenchBuddy that demonstrates the implementation of the Slack API calls, the filtering of teammates relevant to the new project's description, how the trait scores are calculated, and finally, selecting the best team.

Resources for interacting with the prototype:
Slack token:
xoxb-5722573884021-5711411142343-dlplimHuAlna3mVqQHrtP0gS
Slack invite:
https://join.slack.com/t/testfakebsci/shared_invite/zt-21827a8ep-2uWE9pZNIEHL8dvbucCSPA





How to use BenchBuddy

Launching the App:
When you start the application, you're greeted with a title "Slack Team Selector" and a sidebar that lets you choose between various functionalities: "Main App", "View CSV Data", and "Refresh Data".

Main App:
Viewing Stored Data:
Slack Introductions CSV: Click on the expander to view the introductions of all Slack users, which have been fetched and stored.
Slack Messages CSV: Click on the expander to view general messages of all Slack users.

Entering Project Description: Begin by providing a brief description of your project in the "Project Description" text area.

Finding Possible Teammates: Once you've entered the project description, click the "Find Possible Teammates" button. Here's what happens behind the scenes:
The application uses the **BERT model** to compare your project description with the introduction messages of all Slack users.Users whose introductions align closely with the project are shortlisted based on a similarity threshold. The filtered teammates are saved in a CSV name 'possible_teammates.csv'.

How to Use BenchBuddy: Cont'd

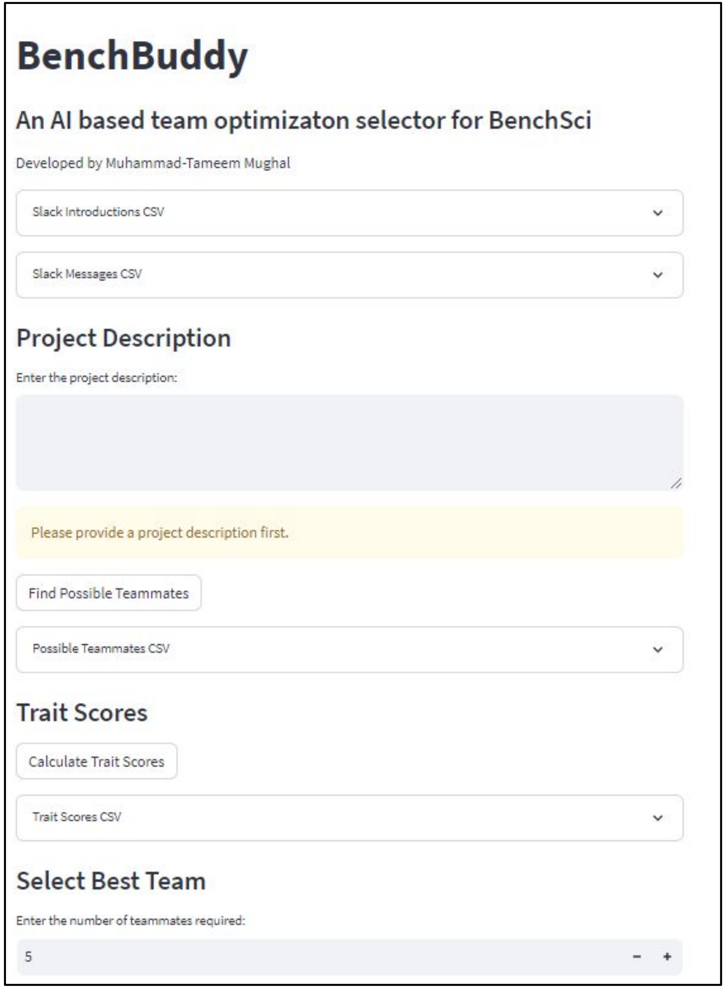
Calculating Trait Scores:
After shortlisting possible teammates, it's time to dive deeper into their traits. Click the "Calculate Trait Scores" button. The **DistilBERT model** analyzes the general chat messages of the shortlisted users. Based on the sentiments of these messages, trait scores (**participation**, **work done**, **compatibility**, **adaptability**) are calculated for each user. These trait scores are saved in 'output_trait_scores.csv'.

Selecting the Best Team:
Define the number of teammates you need in the given input box. Click the "Select Best Team" button. The application then uses **K-Means clustering** on the trait scores to group similar teammates. It recommends a team based on the proximity of cluster centroids to the global average of traits, ensuring a balanced team.

Other Tabs
If you're interested in the raw data, the **View CSV Data** tab allows you to view the different CSV files the program uses: Slack messages, introductions, possible teammates, and trait scores.

The **Refresh Data** tab allows you to update the data provided to the app for the demo.

- Enter your Slack token in the provided input box. (Must have permissions and be a bot-level token.)
- Click "Refresh Slack Messages" to update general messages.
- Click "Refresh Introduction Messages" to update introduction messages.
- If testing or demonstrating, you can also opt to "Load Fake Data" which populates your dataset with dummy data.



BenchBuddy’s Prototype Features

Loading Fake Data: BenchBuddy has an integrated function that can load fake data which may be useful for testing or demonstrations.

Fetching Slack Data: Directly communicates with the Slack API to retrieve all user messages and introduction messages.

Cleaning and Storing Messages: Conversations and introduction messages fetched from Slack are stored in CSV files.Messages are cleaned by removing user mentions and URLs.

BERT-based Similarity Calculation: Uses the well-known BERT model to compute cosine similarities between a given project description and users' introduction messages - the similarity threshold can be tweaked.

DistilBERT-based Trait Analysis: Uses a specific version of BERT (the DistilBERT model) for sentiment analysis to derive user traits from chat messages. Current traits include participation, work done, compatibility, and adaptability.

Clustering for Team Recommendations: Implements K-Means clustering to group users based on their trait scores. Recommends teammates based on cluster centroids' proximity to the global average of traits.

Areas For Improvement

There's a couple things that can be improved in this prototype. For one, the Slack API call could be converted to a batch call that only takes in messages from a certain period of time to avoid rate limiting or overworking the internal servers and computer power.

Additionally, the models - though a working proof of concept, could all use much better fine-tuning, provided a small team were assigned to them dedicated to fine-tuning and creating labelled datasets to improve the models further. An idea to be explored could be using OpenAI's preexisting GPT/LLMs and calling their API to analyze traits - though they would have to be fine tuned very heavily to ensure the proper output.

Moreover, a feature that was developed but not able to be fully implemented would be an automated creation of a slack channel, google doc, and notion page for all the teammates that were selected by the model.

BenchSci: An ideal candidate for this solution

Benchsci is the leading provider of AI-powered tools for life sciences research with over **800% revenue growth form 2018-2021.**

~400 employees means that BenchSci has multiple experts in the same field to choose from based on dynamic traits.

BenchSci is **remote-first** which allows it to attract and **retain top talent from all over the world**.

With offices in Vancouver, Ottawa, Montreal, Toronto, Cambridge, UK and employees working remotely from different locations worldwide, BenchSci has **employees with a variety backgrounds, skills and traits**.

Invests in DEI with platforms like BenchShe+ and named one of the 2023 Best Workplaces for Women by Great Place to Work® Canada

BenchSci is a global leader in machine learning applications for novel medicine development. It is working on innovative and complex solutions for accelerating life saving research which involves multidisciplinary research and expertise. It needs teams that are compatible, highly efficient and productive as this will allow better alignment with their core FASTT values: Speed.

With a rapid scale up, BenchSci may not have the necessary time and resources for manually creating a roster of best fit employees and assigning them to teams. The Automation of this process may be a feasible solution for creating teams that are compatible based on dynamic traits.

How This Will Benefit BenchSci

Automated ML based system generating compatible team recommendations in minutes in a fraction of cost will offer several benefits:

1. **A competitive edge:** Having more efficient teams will accelerate product development and boost revenue, thus giving BenchSci an edge over its competitors.
2. **Intra-company health:** Employees will feel motivated and enjoy working with team members that compliment each other which will allow retention of employees for longer duration and increased employee satisfaction levels.
3. **Diversity and inclusion:** Human bias can sometimes affect the way that managers assemble teams. An automated system can provide a bias-less evaluation of the skills and experience of potential team members.
4. **Transparency:** Using an automated system may increase transparency and help managers make more informed decisions in regards to team composition.

Microsoft Project 1980s

BenchBuddy finds itself in a similar situation to that of the launch of Microsoft Project. Initially used as an internal tool within Microsoft to streamline and track ongoing tasks, it was later made publicly available in 1984. Prior to Project, workers would use analog pencil and paper to track and manage resources, budgets, timing, and other tasks. Project proved to be a game-changer by providing specific tools for project managers and inspired a new generation of project management solutions. Project had a huge boost from Microsoft's branding, especially with the explosive success of Windows 3.0 and Windows 95 and beyond.

Similarly, Benchsci building in-house software will massively set it apart from its competitors who also lack such system with heightened team efficiency. The produced ML system could also later be considered for commercial sale, and may open new avenues for revenue generation .

Addressing The Potential Problems & Solutions

Possible Scaling Issues

API Rate Limits: Fetching messages for each user from every channel can hit Slack's rate limits. The program already implements a sleep timer to avoid this, but as the number of users or channels increases, this may need to be optimized.
Data Size: As the volume of messages grows, so will the computational requirement, especially for models like BERT and DistilBERT.
Memory Consumption: Clustering on a large dataset can be very memory-intensive.

Solutions for mitigating it

Batch API Calls: Instead of fetching messages for each user one by one, utilizing batch API requests from slack may prove better.
Optimizing Data Handling: Only fetching and processing the necessary data, perhaps by introducing a time window (e.g., last 6 months).
External Databases: Storing data in external databases and fetching only when necessary.
Distributed Processing: For computation-intensive tasks, consider distributed processing or cloud-based solutions.

Model Limitations

Generalization: Models like BERT and DistilBERT, although powerful, are still generic. Their performance can be improved by fine-tuning on domain-specific data.

Suggestions

Fine-tuning: Fine-tune models on labelled Slack message data to improve accuracy.
Integrate API calls: Instead of using large models locally, API calls to services like OpenAI's GPT-based models can be considered for certain tasks.

On A **Personal** Note

I am super grateful to have worked along Benchsci. It was a great learning experience for me and I have gained insights into how the company operates.

jacobappleton2@gmail.com

Thank you so much for the opportunity to learn about a new market! I had an amazing time developing for this challenge.

mtameem.mughal@gmail.com

We really appreciate the opportunity to work alongside you guys for this challenge, and we can't wait to see Benchsci scale up further.

easha.noor.7@gmail.com

Thank you for this opportunity! I enjoyed learning about BenchSci and having the chance to explore new areas..

colyichen82@gmail.com

APPENDIX

More Information Available Here: **Memo**