THE IMPACT OF TECHNOLOGICAL INNOVATION ON THE FINANCE INDUSTRY: A COMPREHENSIVE OVERVIEW OF HISTORICAL MILESTONES, CURRENT TOOLS, AND FUTURE TRENDS

COMSBC3997 Projects in Computer Science

MAY 1, 2023

BARNARD COLLEGE
Lily Cai, cc4672

Acknowledgment

I would like to take this opportunity to express my sincere gratitude to Professor Brian Plancher for providing the opportunity to investigate a topic of interest for this course. The course has been invaluable in expanding my knowledge and understanding of the impact of technology on the finance industry. I would also like to extend my thanks to all the guest lecturers who generously shared their insights and experiences, as well as the individuals who were interviewed and provided their valuable input for my research. Additionally, I am grateful for the resources provided during the course, including the YouTube video and tutorial sites, which were immensely helpful in exploring the technical aspects of my research. Overall, I am grateful for this enriching experience, and I believe it has helped me develop a deeper understanding of the complex relationship between technology and finance.

Abstract

This academic paper investigates the impact of technology on the finance industry, including its potential to displace human workers in areas such as customer service and trading. The paper begins by examining the history of technological adoption in finance, highlighting the major milestones and breakthroughs that have transformed the industry over time. The paper then transitions to focus on current tools being utilized in finance and explores the positive impacts and potential drawbacks of increasing technologization in finance. Additionally, the paper examines the differences between traditional financial tools and newer technologies, assesses how current college curriculums may be falling short in preparing students for the workforce, and explores how companies leverage technology to streamline financial processes and services. Finally, the paper proposes potential future trends and technological developments that will likely shape the finance industry in the coming years based on research and experimentation. By exploring these various aspects, this paper aims to provide a comprehensive overview of how technology has been utilized in finance and its potential to transform the industry.

Introduction

The rapid pace of technological innovation in finance has accelerated technology adoption in recent years. The adoption has sparked intense debates regarding its impact on traditional practices and employment. One of the primary concerns is the potential displacement of human workers as machines and algorithms replace their roles in areas such as customer service and trading. While some argue that the increasing use of technology in finance will inevitably lead to job losses, others contend that it will create new opportunities and require a different skill set. This paper seeks to investigate the impact of technology on the finance industry by delving into the industry's history of technological adoption and examining the major milestones and breakthroughs that have transformed the sector over time. Then transitions to focus on the current tools being utilized in finance and explores how technology can make tasks more automated while also considering both the positive impacts and potential drawbacks of increasing technologization in finance. In addition, the aim is to explore the differences between traditional financial tools and newer technologies, gain insight into how current college curriculums may be falling short in preparing students for the workforce, and examine how companies leverage technology to streamline financial processes and services. Lastly, the paper will propose potential future trends and technological developments that will likely shape the finance industry in the coming years based on research and experiments. By exploring these various aspects, the paper aims to provide a comprehensive overview of how technology has been utilized in finance and its potential to transform the industry.

Methodology

There are two approaches employed in this research methodology. First, the research involves a comprehensive analysis of past cases in the field of computer science to determine the current state of the art in relevant areas. Additionally, the project experimented with existing tools to test their effectiveness and determine their limitations. These two approaches combined allow for more meaningful conclusions and develop novel solutions to address gaps in current knowledge and technology. By combining historical analysis and practical experimentation, the paper aims to create a nuanced understanding of the subject matter and contribute to the body of knowledge in computer science. This research aims to produce insights and solutions that can benefit practitioners and researchers in the field, both novice and experienced industry practitioners. Meanwhile also advances personal understanding of the complex challenges and opportunities in this ever-evolving field.

Background

With technological advancements such as ChatGPT, concerns about the future job market have become increasingly dire. A report by Osborne and Frey from the University of Oxford estimates that up to 47% of jobs are at risk of being displaced by automation over the next two decades. Industries such as finance are particularly susceptible, with Bloomberg Businessweek projecting that the impact of AI and automation on the financial sector may displace up to 230,000 jobs within the next decade. Other industries, such as manufacturing and sales, are also expected to face varying degrees of impact and transition due to ongoing technologization.

In the face of ongoing debates and concerns about the potential impact of technology on the finance industry, students need to prepare themselves for the evolving job market. This raises important questions about the future for the next generation of students. To answer these questions, it is necessary to examine the history of technology in the finance industry, including the major milestones and transitions that have taken place. Students can better position themselves for success in a rapidly changing job market by understanding how technology has evolved within the finance industry.

History of Technology in the Finance World

In the 1960s, the launch of electronic trading systems and mainframe computers marked the beginning of the automation of financial markets. The New York Stock Exchange introduced the Automated Trading System (ATS) in 1969, the first-ever electronic trading system.

Over the two decades from the 1970s to the 1990s, the development of technology in the finance industry has been marked by both physical and digital innovations. While the introduction of the first automated teller machine (ATM) in the 1970s revolutionized customer access to bank accounts, the emergence of personal computers and the internet in the 1980s

transformed the delivery and consumption of financial services. In addition, the late 1980s and early 1990s saw the launch of online trading platforms such as E*TRADE and TD Ameritrade, which allowed retail investors to trade stocks and other securities with greater ease and accessibility. These innovations improved the customer experience and opened up new opportunities and challenges for the finance industry.

The 2000s marked the rise of mobile devices and the widespread availability of internet connectivity, leading to the development of mobile banking apps and payment systems like Venmo and PayPal. Additionally, the launch of Bitcoin, the first cryptocurrency in 2009, introduced blockchain technology, which could disrupt traditional financial systems and revolutionize financial transactions. In the 2010s, using artificial intelligence, machine learning, and big data analytics enabled financial institutions to analyze vast amounts of financial data, leading to more informed investment decisions, risk management, and customer preferences. Automated investment platforms such as robo-advisors became more popular, making investing more affordable and accessible for retail investors.

On the one hand, these changes in technology over the past several decades have transformed the finance industry, making it more accessible, efficient, and global. Investment transactions are now conducted electronically; customers can access financial services from anywhere with an internet connection. However, on the other hand, new risks and challenges arise from cyber-attacks, integrity and security breaches that stem from more accessible and personalized technologies. In addition, the growing technology in finance has significant implications for college and younger students, particularly those considering a career in the industry.

Here is a summary of the trends and patterns I concluded based on my research. The trends include how technology has influenced the financial world and future trends:

- The emergence of technology has had a profound impact on the finance industry, leading to significant changes in how financial services are delivered and consumed.
    - Increased accessibility of the web and digitalization have allowed customers to access financial services virtually anywhere at any time.
    - Transitioned to more personalized and customized services as financial institutions leveraged customer data to provide tailored products and services.
    - Expanded automation in the finance industry technologies allowed for faster and more efficient transactions.
    - Furthered the emphasis on cybersecurity and legal compliance has become increasingly important as financial institutions become more reliant on technology.
- As technology continues to reshape the finance industry, students should be prepared to adapt and acquire new skills that will be in demand in the future job market. This includes a greater emphasis on data analysis, machine learning, and programming skills as these technologies become more integral to the industry.
- Additionally, the rise of fintech startups and other disruptive technologies means that traditional business models are being challenged, and students will need to be comfortable with rapid change and innovation. On the other hand, technology is also making financial services more accessible and affordable, which is particularly beneficial for younger people who may have limited access to traditional financial institutions.

Overall, college and younger students should remain proactive in their learning and skill development to stay ahead of the curve in a rapidly changing job market. They can also benefit from increased access and affordability of financial services.

Methodology

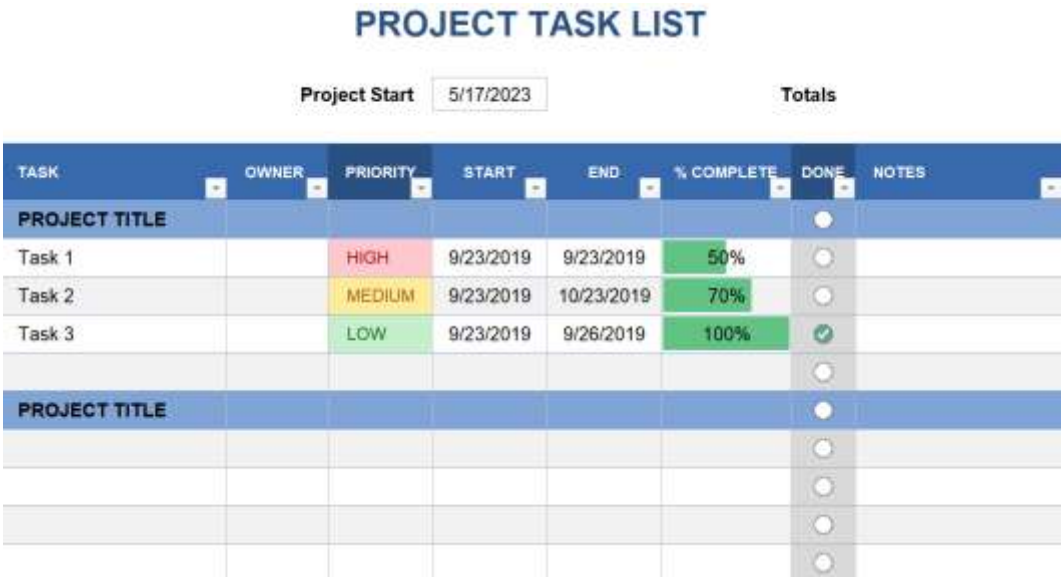
Exploring some current financial practices and possibilities of using technology to solve the same issues:

Organizing Tasks (Programming Language: Java)

While exploring technology's capabilities, I became curious about its potential for organizing and managing tasks. My curiosity started from what I observed during my internship. I was interested to see how my manager coordinated tasks and wondered if technology could simplify team coordination and task tracking. To model a real-world scheduler, I created a task manager using Excel. I included the task name, who owns the task, priority, start and end date, percentage completion, and notes for this sample task manager. For the most part, the tracker was relatively easy to create. Some more advanced features, such as documenting how much of the work is completed with a colored bar required some research. However, many tutorials on YouTube were less than twenty minutes, so by watching a few of them could gain a relatively solid understanding. It was also easy to work with Excel and didn't require much prior knowledge or background.

Despite the ease, I have also discovered that Excel might not necessarily be the most convenient way to organize tasks. First, Excel's usefulness is limited to managing simple lists of data. As the complexity of a project grows, it becomes more challenging to manage and track progress using Excel alone. Additionally, Excel lacks the functionality to collaborate effectively, particularly when team members work remotely. Once Excel is shared with more people, and as people edit simultaneously can lead to collides and errors. This can result in delays, miscommunication, and mistakes, making keeping everyone on the same page challenging. Furthermore, Excel doesn't provide real-time insights into the progress of a project, making it difficult to adjust and adapt to changing circumstances.

Overall, the positive part was how easy and how many resources there were available for building a basic project tracker that currently exists. However, the drawback was that while this model was effective for fewer tasks and team members, the system became slow and error-prone as the number of tasks and team members increased.



PROJECT TASK LIST

Project Start 5/17/2023 Totals

TASK	OWNER	PRIORITY	START	END	% COMPLETE	DONE	NOTES
PROJECT TITLE							
Task 1		HIGH	9/23/2019	9/23/2019	50%		
Task 2		MEDIUM	9/23/2019	10/23/2019	70%		
Task 3		LOW	9/23/2019	9/26/2019	100%		
PROJECT TITLE							

Figure 1: An image of the project task list using Excel.

To experiment with how programming can help resolve some issues, I developed a task manager using Java, which could handle many tasks and team members without slowing down or crashing. Yet, I soon realized that my vision for a complex and sophisticated task manager might not be practical. Firstly, technical issues arose due to my lack of recent experience in Java, resulting in debugging challenges and difficulty remembering how to code. While I have created a basic version capable of adding and deleting items using an Array List, I still need to find a way to store and coordinate tasks between team members. That would require more engineering and think through the specific data structure I want to use to keep the information. Secondly, not all team members have the same programming proficiency, which could hinder a technology-based scheduler's effectiveness.

Figure 2: Using an online compiler to run the Java code.

Thus, the current scheduler may not be the best option for team coordination. It would require everyone on the team to have at least a basic level of proficiency with programming and understanding code. Also, the version I created in Java is a bare minimum prototype, lacking a well-thought-out data structure and visually appealing interfaces. This would make developing an architectural framework and implementing visually appealing interfaces a challenging endeavor. Therefore, for the next steps, I would want to spend more time investigating connecting to the database management system and re-designing the interface so that the scheduler could be shared with others, store the information, and have a better visual display. Additionally, there is a risk that if the code breaks, if the other members don't know to code, it may be challenging to fix and continue using the model if there are any errors.

Conducting Financial Modeling (Programming Language: Python)

Financial practitioners currently conduct financial modeling using various tools and techniques, with Microsoft Excel being the most popular one. Excel's ease of use, wide availability, and versatility make it a popular choice for financial modeling. However, using Excel for financial modeling has its benefits and drawbacks. On the one hand, it allows

practitioners to create complex models with little effort, conduct sensitivity analysis, and produce charts and graphs that aid decision-making. However, on the other hand, Excel has several limitations that can create problems, such as a lack of transparency, difficulty in tracking changes, and susceptibility to errors.

For the project, I tried using Excel to model and perform calculations based on 5-year historical data to simulate how financial practitioners might utilize Excel to analyze trends and make informed decisions. One of the primary advantages of using Excel is the ease of use and organization into cells that creates a clean visualization. Excel also has a wide range of built-in functions and formulas, which can save time and improve accuracy when performing complex calculations. In addition, some shortcuts made it easy to apply across the rows and columns when Excel can detect a pattern (for instance, auto-fill through CTRL+R). However, using Excel has several drawbacks, mainly when working with large or complex datasets. For example, as the number of rows increased, my Excel would freeze or crash more often. As a result, I had difficulty redoing part of the work due to Excel crashing and not saving the work beforehand.

171	D. Returns						
172	Net Income (Net Earnings)	2914	2914	2914			
173	Shareholder's Equity	10,953	11,709	11,297			
174	*Return on Equity Ratio*	0.27	0.25	0.26			
175							
176	Total Assets	40785	#REF!	41290			
177	*Return on Assets Ratio*	0.07	#REF!	0.07			
178							
179	E. Liquidity						
180	Items:						
181	Current Assets	11990	11990	11990	11990	11990	11990
182	Current Liabilities	12707	13201	15014	14487	20125	21747

Figure 3: Frozen Excel screen.

One issue I faced was the potential for errors or inaccuracies arising from data entry mistakes, formula errors, or other issues. For example, it was easy to make careless mistakes

such as referencing the wrong cell. In those cases, the cell might lead to "#REF" (reference) issues, which can take time to figure out the problem. The larger the number of cells made it more difficult to scroll through the model and identify the error or issue.

282	Asset						
283	Cash and Cash Equivalents	2643	#REF!	#REF!	#REF!	#REF!	#REF!
284	Inventory	8,597	#REF!	#REF!	10961.59	12,417	13,620
285	Other Current Assets	1,300	#REF!	#REF!	#REF!	#REF!	#REF!
286	**Total Current Assets**	12540	#REF!	#REF!	#REF!	#REF!	#REF!

Figure 4: Image of a potential error that happens while using Excel.

Dealing with multiple years of reports in Excel was also an extremely frustrating experience. Extracting relevant information becomes arduous due to different tab names in each report, and changes made to these reports over time. Unfortunately, not even an AI tool such as ChatGPT can give me past figures when I prompt it. Therefore, doing financial modeling with Excel might lead to opening several tabs at once, ensuring none of the data are wrong, and cleaning up the data set. All of these make it less efficient and effective.

give me target's account receivable from 2017-2022

I'm sorry, but I'm an AI language model and I don't have access to specific company's financial records or information. In order to obtain a company's accounts receivable for a specific period, you would need to consult their financial statements or contact the company directly.

Figure 5: Asking ChatGPT to give the Target's Account Receivable.

In addition, the unorganized naming convention makes it challenging to identify and organize data effectively, wasting valuable time. The lack of consistency in the formatting and

structure of these reports can also contribute to errors and inconsistencies, further exacerbating the problem. Finally, there may be limitations to Excel's capabilities, mainly when dealing with more advanced statistical analyses or machine learning models, which may require more specialized software or programming skills. Despite these challenges, Excel can be a valuable tool for analyzing and modeling historical data, mainly when used with other devices and techniques.

From there, I tried utilizing Python to conduct the exact financial modeling with past historical data. Unfortunately, the process was more challenging and frustrating than expected. First, the importing and cleaning of data took incredible effort. Companies are different from one another, and even within the company, the organization may be other. Hence, it requires looking into Excel and ensuring a solid understanding of the data structure within each file. Further, organizing the data has been difficult. Although Pandas and Numpy (Python packages often used for data analysis) offer it in a table manner, Excel can do the same work with fewer lines of code. Learning Excel quickly rather than all the Python syntax and principles is also easier. The debugging process was frustrating and can only make it more difficult for someone in the financial world if they have yet to learn to code.

Therefore, for this semester, I only got as far as writing several bugs before realizing my inefficient knowledge of the subject. Hence, I am currently enrolled in classes, hoping to gain more about the topic. I found that the courses, videos, and tutorials offered through various educational platforms can be frustrating for several reasons. Firstly, the courses' earlier and more "beginner" levels can be overly simplistic and time-consuming. While I understand the importance of laying a solid foundation, I often feel bored and frustrated by the lack of challenge in these initial stages. However, skipping over these earlier parts can sometimes lead to

difficulties with later, more complex chapters. Secondly, the available practice examples are not always practical enough to simulate real-world scenarios. Many examples are heavily staged and provide clear hints about what steps to take, which doesn't reflect the challenges faced in actual work settings.

Moreover, each concept only has a few practice problems, making it difficult to internalize and apply the material effectively. Finally, the time constraints of balancing other coursework or professional responsibilities can make it challenging to complete these courses efficiently. It's like a student juggling multiple classes while having to self-learn or an entry-level employee with daily job responsibilities but needs to learn something. Finding the energy and motivation to engage with the material can be challenging. These factors can make learning effectively and efficiently through online courses, videos, and tutorials difficult.

Although this portion of the research project did not end as smoothly as I expected, I did start to learn how to use Python to calculate simple financial equations such as compound interest. The benefit of Excel is that storing the values into variables can help reduce human error so long the number and equation are correct.

Figure 6: Snapshot of using Python for financial calculations.

Therefore, when it comes to financial modeling, there are various factors to consider when selecting the appropriate tool for the job. I would lean towards Excel even though Python offers more flexibility and scalability. One advantage of Excel is that it is a widely-used tool and is relatively easy to learn, making it more accessible to a wide range of users. Additionally, Excel is well-suited for smaller and less complex data sets and for creating charts and graphs. Furthermore, Excel's existing interface is easy to understand and manipulate. However, I would consider switching to using Python for financial models instead of Excel if data sets become more complex and tasks become more sophisticated. Moreover, if the data sets require repeated modeling and manipulation, then Python offers greater transparency and reproducibility of the code, which can be crucial in larger teams. Ultimately, the decision between Excel and Python depends on the specific needs and goals and the complexity of the task at hand. While Excel may be sufficient for specific tasks, those requiring greater flexibility and scalability may benefit from switching to Python modeling.

Doing Basic Data Analysis for Data Stored in Database (Programming Language: SQL)

Rather than solely focusing on Excel and Python, I wanted to try other software that might be useful in the finance industry. During my interview, a senior professional who has worked at Goldman Sachs, LibreMax, and HPS provided an overview of some of the tools he has used throughout his career. He mentioned that his go-to would be Excel, especially if it's more one-time computations. Then, he will use Python for repeated modeling and analyses. Lastly, SQL if there are databases so he can better grasp the relationships between data. After talking to him, I chose SQL to experiment with. Using SQL to model and analyze financial data can offer several benefits, particularly in the financial industry context. SQL is a powerful tool for querying and manipulating data stored in relational databases commonly used in financial systems. By using SQL, practitioners can easily extract, filter, and join data from multiple tables, enabling them to perform complex analyses and generate insights. SQL is also a widely recognized skill in the financial industry, and proficiency in SQL can enhance one's career prospects in finance. However, working with SQL can also present challenges, particularly for those new to the language or with limited database experience. Obtaining access to relevant databases and setting up connections can be time-consuming and require specialized knowledge. Navigating SQL syntax and designing efficient queries can also be difficult without proper guidance.

Therefore, instead of completely modeling how to finance practitioners would use SQL during their work, I used a "guided" project to get the task. Although the experience was intriguing, it felt staged and spelled out to the point that it didn't feel real. Many hints and step-by-step instructions weren't always available in the real world. There were also options to check answers, which in the daily work of an SQL analyst in finance, they might not get the same level

of guidance. Hence, not as satisfying and creates a potential divide between courses, real-world applications, and job settings.

Automate Task By Merging PDFs (Programming Language: Python)

The last area I tried was to automate tasks with programming to save time. In this case, I noticed how I was often frustrated at the existing merging of PDF platforms and services. There are service restrictions with the free version, while the premium version costs too much. Therefore, I tried to use Python to automate and merge PDFs. In my research, I used Python to create a program that automates merging multiple PDFs into a single document. This program was built using a variety of Python libraries, including PyPDF2 and PyPDF4, which provide tools for reading, writing, and manipulating PDF files. One of the main advantages of using Python for this task is the ability to automate the process, which can save a significant amount of time and effort compared to manual merging. Python's flexibility and customization options also allowed me to tailor the program to my specific needs and preferences, making it a valuable tool for my research.

However, using Python for PDF merging also has some potential drawbacks. One of the main challenges is the learning curve for beginners, as Python can be daunting for those unfamiliar with coding. Additionally, Python's flexibility can be a double-edged sword, as it can be time-consuming to fine-tune the program to achieve the desired results. For instance, PyPDF2 was outdated and needed to switch to PyPDF4, which had different package names or syntax. Hence, users would need to keep up to date with the revisions to the code and adjust accordingly. Furthermore, it could not be easy for an average learner to debug. However, once learned, it can save time and energy as the user will no longer need to rely on free online services with a limited number of daily merges. From my perspective, I would recommend this to average learner even

with little or no coding experience, because this is easy to pick up quickly compared to some other tasks. Also, even if someone doesn't know, the concepts and ideas can be explained, and the code won't require much manipulation. Therefore, someone more experienced could write the code, provide tutorials, and others could use it. Overall, the benefits of using Python for PDF merging outweigh the potential challenges, as it provides a powerful toolset for automating tasks and streamlining the research process. If I have more time, I would also be interested in investing in other ways to automate tasks with technology to see whether the efficiency and benefits are worth the learning curve and difficulty of starting.

```
import PyPDF2
from pptx import Presentation
from pptx.util import Inches
```

```
pdf_file = open('/content/chapter2 (1).pdf', 'rb')
pdf_reader = PyPDF2.PdfReader(pdf_file)
```

```
from pptx import Presentation
from pptx.util import Inches
import PyPDF2

# Open the PDF file for reading in binary mode
pdf_file = open('/content/chapter2 (1).pdf', 'rb')

# Create a PDF reader object
pdf_reader = PyPDF2.PdfReader(pdf_file)

# Create a PowerPoint presentation object
ppt = Presentation()

# Iterate over the pages of the PDF file, add each page to the PowerPoint presentation as an image
for page_num in range(pdf_reader.numPages):
    # Get the page from the PDF reader
    pdf_page = pdf_reader.getPage(page_num)
    # Convert the PDF page to a PNG image using the pdf2image library
    png_image = pdf2image.convert_from_bytes(pdf_page)
    # Add the PNG image to a new slide in the PowerPoint presentation
    slide = ppt.slides.add_slide(ppt.slide_layouts[1])
    pic = slide.shapes.add_picture(BytesIO(png_image), 0, 0, height=ppt.slide_height, width=ppt.slide_width)

# Save the PowerPoint presentation
ppt.save('SQL_filter.pptx')
```

```
---------------------------------------------------------------------------
DeprecationError                          Traceback (most recent call last)
<ipython-input-32-e975d461b98c> in <cell line: 15>()
     13
     14 # Iterate over the pages of the PDF file, add each page to the PowerPoint presentation as an image
---> 15 for page_num in range(pdf_reader.numPages):
     16     # Get the page from the PDF reader
     17     pdf_page = pdf_reader.getPage(page_num)

                                  2 frames
/usr/local/lib/python3.10/dist-packages/PyPDF2/_utils.py in deprecation(msg)
    349
    350 def deprecation(msg: str) -> None:
--> 351     raise DeprecationError(msg)
    352
    353

DeprecationError: reader.numPages is deprecated and was removed in PyPDF2 3.0.0. Use len(reader.pages) instead.

SEARCH STACK OVERFLOW
```

```
---------------------------------------------------------------------------
DeprecationError                          Traceback (most recent call last)
<ipython-input-14-68d3cb369cfd> in <cell line: 1>()
----> 1 page1 = pdf_reader.getPage(0)

                                  2 frames
/usr/local/lib/python3.10/dist-packages/PyPDF2/_utils.py in deprecation(msg)
    349
    350 def deprecation(msg: str) -> None:
--> 351     raise DeprecationError(msg)
    352
    353

DeprecationError: reader.getPage(pageNumber) is deprecated and was removed in PyPDF2 3.0.0. Use reader.pages[page_number] instead.
```


```
---------------------------------------------------------------------------
DeprecationError                          Traceback (most recent call last)
<ipython-input-14-68d3cb369cfd> in <cell line: 1>()
----> 1 page1 = pdf_reader.getPage(0)

                                  2 frames
/usr/local/lib/python3.10/dist-packages/PyPDF2/_utils.py in deprecation(msg)
    349
    350 def deprecation(msg: str) -> None:
--> 351     raise DeprecationError(msg)
    352
    353

DeprecationError: reader.getPage(pageNumber) is deprecated and was removed in PyPDF2 3.0.0. Use reader.pages[page_number] instead.
```


Figure 7: Some errors faced while using PyPDF2.

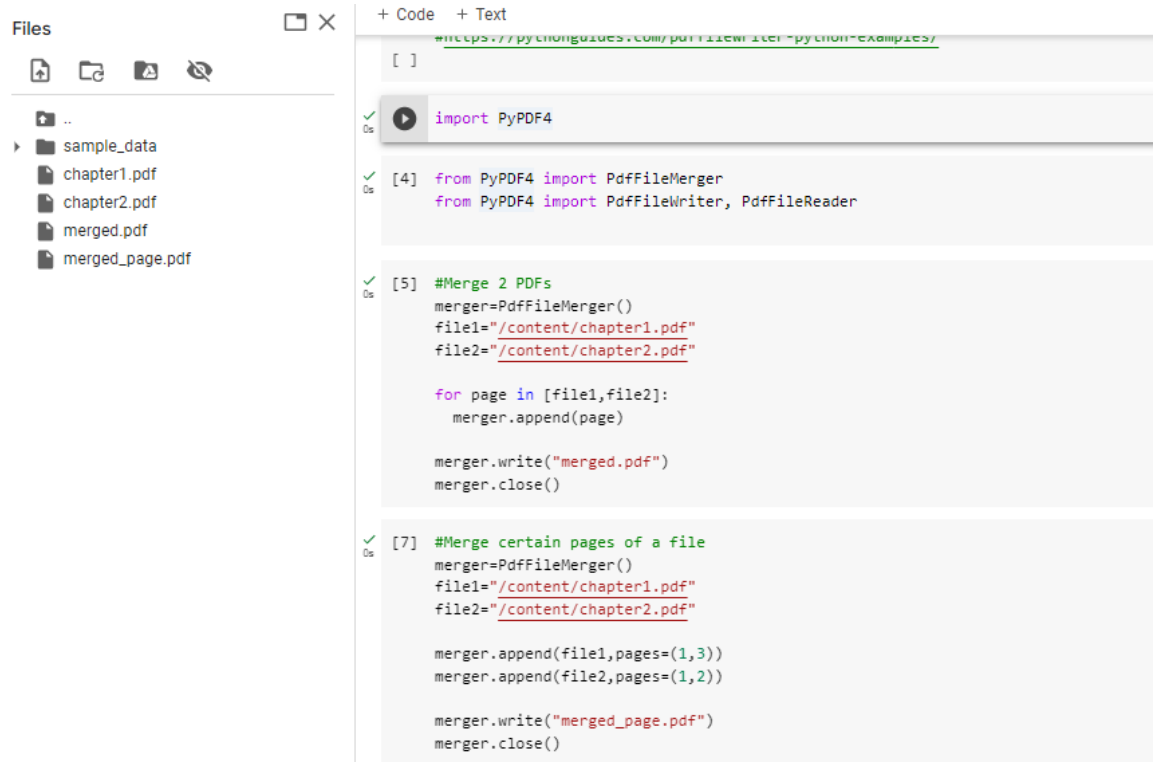


Figure 8: Using PyPDF4 and successfully merging documents.

Web Scraping Financial News and Information (Programming Language: Python)

In the finance industry, keeping up to date with current news and events is crucial for making informed decisions and managing risks. Financial professionals often rely on various sources to stay informed, including financial news websites, industry publications, and analyst reports. Many companies also employ dedicated teams to monitor news and market trends and use specialized software to analyze data and identify relevant news articles. Therefore, I tried to see whether I could use web scrapping to extract the critical information and news, then organize it into summaries to make it more efficient for people who require accessing this news daily.

As someone with prior knowledge in HTML and CSS due to UI design, I decided to delve into web scraping with Python. While automating data extraction from websites was intriguing, I quickly realized that the process was far from straightforward. Despite my familiarity with HTML and CSS, I encountered numerous obstacles in extracting data from websites. Some

websites had security measures that made it impossible to scrape data, while others required complex regex expressions to extract the desired data. Additionally, organizing the extracted data into meaningful lists and structures proved challenging. Even when my code appeared correct, the output often failed to match my expectations. As a result, I struggled through a steep learning curve, constantly searching for answers and troubleshooting my code. While tutorials and online resources were helpful, they often glossed over crucial details or made assumptions about the user's knowledge level.

Despite the challenges associated with web scraping, I believe that it has the potential to be a valuable tool for data analysis and research. With the vast amounts of data available on the internet, web scraping offers a way to collect and organize data that would be difficult or impossible to collect manually. Furthermore, web scraping with Python can be highly customizable, allowing users to tailor their scraping techniques to their research or analysis needs. However, as previously mentioned, the learning curve for web scraping can be steep, requiring time and effort to become proficient. This can be a challenge for finance professionals, who often have many other tasks and responsibilities competing for their time and attention. In addition, there may be concerns about the legality and ethical implications of web scraping, particularly when scraping personal or sensitive data. These factors may make some finance professionals hesitant to adopt web scraping and other emerging technologies, preferring to rely on more traditional data collection and analysis methods. Despite these challenges, emerging technologies and tools can offer significant advantages in terms of efficiency, accuracy, and speed of data analysis. Ultimately, the decision to adopt new technologies like web scraping will depend on a range of factors, including the specific needs and resources of the individual or organization and any regulatory or ethical considerations.

Implications

While the finance industry has seen significant advancements in technology and automation, there are still challenges to implementing large-scale transitions to complete automation. One of the main challenges is the steep learning curve associated with many of these technologies, which may require significant investment in training and development to become proficient. In some cases, the time and resources needed to implement these technologies may not be worth the benefits, particularly for smaller firms or those with limited budgets. Additionally, there is a growing recognition of the importance of human expertise and judgment in finance, particularly in risk management and compliance areas, which may be difficult to automate. Despite these challenges, the job market and statistics suggest that there are still ample opportunities and growing demand for skilled professionals in the finance industry. Therefore, a balance between technology and human expertise is likely the optimal approach, with technology as a tool to enhance rather than replace human capabilities.

As the million-dollar question of what will happen next, predicting the finance industry's future with technology use is difficult. Still, several trends will likely continue shaping the industry over the next 20 years. One trend is the increasing use of artificial intelligence and machine learning, which are already used for fraud detection, risk management, and customer service tasks. This trend is likely to continue, with AI and machine learning becoming more sophisticated and integrated into more areas of finance.

Another trend is the increasing importance of cybersecurity and data privacy as the finance industry becomes more reliant on technology and more vulnerable to cyber-attacks. This will increase the emphasis on developing and implementing robust cybersecurity protocols and training programs.

There is also likely to be continued growth in fintech, blockchain, and cryptocurrency, as new technologies and platforms emerge and become more widely adopted. The new disruptive changes will create new opportunities for innovation and disruption and new challenges for regulation and oversight.

While technology is likely to continue changing the finance industry in significant ways, it is unlikely to take away jobs completely. Instead, technology is more likely to augment and enhance human capabilities, allowing for more efficient and effective decision-making and task management. As such, professionals in the finance industry will need to continue adapting and developing new skills to stay competitive in a rapidly changing landscape.

Work Cited

"Artificial Intelligence & Robots: Are They Going to Take Our Jobs?" PeopleFacts, 13 March 2019, https://peoplefacts.com/artificial-intelligence-robots-going-take-jobs/#:~:text=The%20LA%20Times%20reports%20that,software%20now%20to%20compose%20articles.

"E-Commerce Automation: Robots Create More Jobs Than They Take" by The Associated Press. Inc.com, 17 September 2019, https://www.inc.com/associated-press/e-commerce-automation-robots-create-more-jobs-amazon-effect.html.

Frey, Carl Benedikt, and Michael A. Osborne. "The Future of Employment: How Susceptible Are Jobs to Computerisation?" Oxford Martin School, 17 September 2013, https://www.oxfordmartin.ox.ac.uk/publications/the-future-of-employment/.

"Report: Robots Will Replace 20 Million Manufacturing Jobs by 2030" by The Associated Press. U.S. News & World Report, 26 June 2019, https://www.usnews.com/news/economy/articles/2019-06-26/report-robots-will-replace-20-million-manufacturing-jobs-by-2030.