

1 המהפכה השקטה - מבוא למודלי שפה גדולים בעולם העסקי

תקציר

בעידן שבו טכנולוגיה משנה את עולם העסקים בקצב מסחרר, כמה מהפכה חדשה - שקטה אך עמוקה. מודלי שפה גדולים (Large Language Models, LLMs) משנים את האופן שבו ארגונים מתקשרים, מנתחים מידע ומקבלים החלטות. פרק זה מציג את היסודות המנהליים להבנת הטכנולוגיה, פוטנציאלה ומגבלותיה, ומספק כלים מעשיים להערכת התועלת העסקית שלה.

מטרות הלמידה

בסיום פרק זה, הקורא יוכל:

- להבין את המהות והפוטנציאל של מודלי שפה גדולים עבור ארגונים
- לזהות את שני התפקידים המרכזיים: אינטראקציה בשפה טבעית ועיבוד לוגיקה מורכבת
- להכיר את נקודות החוזק והחולשה של LLMs לצורך קבלת החלטות מושכלות
- לחשב ולהעריך את התשואה על ההשקעה (ROI) ביישום כלי AI
- לזהות תרחישי שימוש מתאימים ובלתי מתאימים למודלים אלו

2 פרולוג: בוקר רגיל בעולם חדש

שבע בבוקר. שרה, מנהלת שיווק בחברת SaaS בינונית, מתיישבת מול המחשב עם כוס קפה. לפנייה משימה מוכרת: כתיבת חמישה פוסטים לרשתות חברתיות לקמפיין החדש. בעבר, זה היה לוקח לה שעותיים לפחות. היום, היא פותחת את ChatGPT, מקלידה הנחיה קצרה עם ההקשר והסגנון המבוקש, וכעבור דקה וחצי - חמשת הפוסטים מוכנים. היא משקיעה עוד עשר דקות בעריכה ועיצוב, ועוברת למשימה הבאה.

באותו זמן, דן, סמנכ"ל הכספים, מעלה לממשק Claude דוח כספי בן 05 עמודים ומבקש סיכום של המגמות העיקריות ונקודות החריגה. כעבור שתי דקות, הוא מקבל ניתוח מובנה שבעבר היה דורש מבקר פיננסי שעה שלמה. הוא לא מסתפק בכך - הוא ממשיך לשאול שאלות המשך, ו-Claude עונה בהקשר מלא, כאילו הוא עמית שקרא את הדוח בעצמו.

בקומה השלישית, רונית ממשאבי אנוש מעבירה ראיון ראשוני עם מועמד. היא לא לבד - לידה פועל סוכן AI שמקליט, מתמלל ומנתח בזמן אמת את תשובות המועמד מול פרופיל התפקיד. כשהשיחה מסתיימת, רונית כבר רואה דוח מסכם עם המלצה ראשונית.

זהו לא מדע בדיוני. זה לא עתיד רחוק. זה היום, עכשיו, בעשרות אלפי ארגונים ברחבי העולם.

אבל מה באמת קורה כאן? מה הופך את הטכנולוגיה הזו לשונה מכל אוטומציה שראינו עד כה? ואיך מנהלים אמורים להבין, להעריך ולהטמיע אותה בארגון שלהם?

3 מהם מודלי שפה גדולים? הסבר אינטואיטיבי

1.3 המטאפורה: מכונת השלמת דפוסים

דמיינו לרגע ילד שגדל בסביבה שבה הוא שומע מיליוני שיחות, קורא מיליארדי משפטים, וחשוף לכמעט כל נושא אנושי אפשרי - היסטוריה, מדע, ספרות, עסקים, פילוסופיה. הילד הזה לא מבין בהכרח את העולם כמו שאנחנו מבינים אותו, אבל הוא מפתח יכולת מדהימה לזהות דפוסים: איך משפטים בנויים, איך רעיונות מתקשרים, איך בעיות נפתרות, איך אנשים מתקשרים בהקשרים שונים.

כשאתם שואלים את הילד הזה שאלה, הוא לא מחפש תשובה במאגר מידע. במקום זאת, הוא משתמש בכל הדפוסים שהוא למד כדי להשלים את המשפט הכי הגיוני, הכי סביר, הכי מתאים להקשר. אם שאלתם על אסטרטגיית שיווק, הוא יזכור מיליוני שיחות על שיווק שהוא "שמע", וישלים את התשובה בצורה שמשקפת את הדפוסים האלה.

זו, בקצרה, המהות של Large Language Model (LLM).

LLM הוא מודל מתמטי ענק שאומן על כמויות אדירות של טקסט - ספרים, מאמרים, אתרי אינטרנט, קוד תוכנה, ועוד. בתהליך האימון, המודל למד דפוסים סטטיסטיים מורכבים:

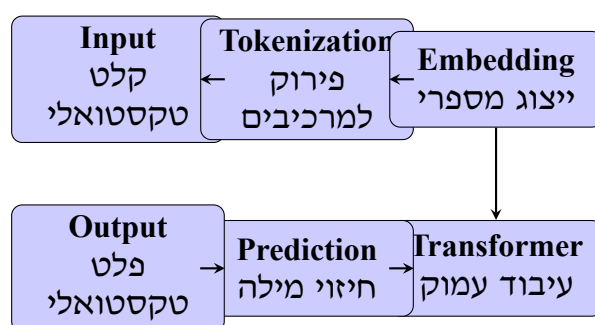
- איזה מילים מופיעות לצד אילו מילים
- איך משפטים בנויים בהקשרים שונים
- איך רעיונות מתקשרים זה לזה
- איך בעיות נפתרות בתחומים שונים

כשאתם כותבים prompt (הנחיה) ל-LLM, המודל "רואה" את הטקסט שלכם ומשתמש בכל הדפוסים שהוא למד כדי לחזות את ההמשך הכי סביר. הוא עושה זאת מילה אחר מילה, תוך התחשבות בכל ההקשר שלפניו.

נקודה קריטית למנהלים: LLM לא "יודע" דברים במובן האנושי. הוא לא מחפש מידע במסד נתונים. הוא יוצר טקסט חדש על בסיס דפוסים סטטיסטיים. זו גם החוזקה (יצירתיות, גמישות) וגם החולשה (אפשרות להזיות) שלו.

2.3 מתחת למכסה המנוע: ארכיטקטורה בסיסית

בלי להיכנס לפרטים טכניים עמוקים מדי, חשוב להבין את המבנה הבסיסי של LLM:



איור 1: ארכיטקטורה בסיסית של LLM - מקלט לפלט

1. **Input (קלט):** המשתמש מזין טקסט - שאלה, בקשה, או הקשר.
2. **Tokenization (טוקניזציה):** הטקסט מפורק ל"טוקנים" - יחידות בסיסיות שהמודל מבין. טוקן יכול להיות מילה, חלק ממילה, או סימן פיסוק. למשל, המשפט "שלום עולם" עשוי להיות 2-3 טוקנים.
3. **Embedding (הטמעה):** כל טוקן הופך לייצוג מתמטי - וקטור של מספרים שמייצג את משמעותו ביחס לטוקנים אחרים.
4. **Transformer (טרנספורמר):** זוהי הלב של המודל [1]. רשת נוירונים עמוקה שמנתחת את היחסים בין הטוקנים, מבינה הקשר, ומחלצת משמעות. כאן קורה "הקסם" - המודל "מבין" מה נשאל וכיצד לענות.
5. **Prediction (חיזוי):** המודל מחזה את הטוקן הבא הכי סביר, מוסיף אותו לרצף, וחוזר על התהליך עד שהתשובה שלמה.
6. **Output (פלט):** הטוקנים הופכים בחזרה לטקסט קריא שהמשתמש רואה.

למה זה חשוב למנהלים?

- **טוקנים = עלות:** רוב שירותי ה-API גובים תשלום לפי מספר הטוקנים שנשלחים (קלט) ומתקבלים (פלט). הבנת טוקנים חיונית לתכנון תקציב.
- **הקשר מוגבל:** לכל מודל יש "חלון הקשר" (context window) - מספר מקסימלי של טוקנים שהוא יכול לעבד בבת אחת. למשל, GPT-4 Turbo תומך ב-128,000 טוקנים (96,000 מילים), בעוד GPT-3.5 תומך רק ב-16,000.
- **מהירות תלויה בגודל:** מודלים גדולים יותר (יותר פרמטרים) בדרך כלל מדויקים יותר, אך איטיים ויקרים יותר.

4 שני הכוחות העל של LLM

1.4 כוח ראשון: תקשורת טבעית עם מכונה

במשך עשרות שנים, האינטראקציה שלנו עם מחשבים הייתה מוגבלת. רצינו שהמחשב יעשה משהו? היינו צריכים ללמוד את שפתו: לחצנים, תפריטים, שורות פקודה, שפות תכנות. המחשב לא הבין אותנו - אנחנו היינו צריכים להתאים את עצמנו אליו. LLMs הופכים את המשוואה. לראשונה בתולדות המחשוב, אנחנו יכולים לתקשר עם מכונה **בשפה שלנו**, בדיוק כפי שהיינו מדברים עם עמית.

דוגמה מעולם העסקים:

בעבר (אינטראקציה מסורתית עם תוכנה):

1. פתח תוכנת Excel
2. בחר טווח תאים
3. לחץ על "נתונים" → "סינון" → "סינון מתקדם"
4. הגדר קריטריונים מורכבים
5. בחר "עותק למיקום אחר"
6. בחר תא יעד

7. לחץ "אישור"

היום (אינטראקציה עם LLM):

"תסנן את הטבלה הזו ותראה לי רק לקוחות מאזור המרכז שרכשו מעל 10,000 ש"ח ברבעון האחרון"

המודל מבין את הכוונה, מזהה את הנתונים הרלוונטיים, ומבצע את הפעולה - או אפילו כותב לכם את הנוסחה המתאימה.

ההשלכות העסקיות:

- **הפחתת מחסום הכניסה:** עובדים לא צריכים להיות מומחי תוכנה כדי לבצע משימות מורכבות.

- **מהירות:** מה שלוקח 10 דקות בממשק מסורתי, לוקח 10 שניות בשפה טבעית.

- **גמישות:** אפשר לשאול שאלות המשך, לשנות דרישות, לחקור כיוונים שונים - בדיוק כמו בשיחה אנושית.

2.4 כוח שני: עיבוד לוגיקה מורכבת

אבל LLMs הם הרבה יותר מסתם ממשק נוח. הם מסוגלים לבצע **חשיבה מורכבת** על נתונים ורעיונות.

בואו נבחן כמה יכולות מרכזיות:

1.2.4 סיכום והפקת תובנות

ניתן להזין ל-LLM דוח של 100 עמודים ולבקש:

- "סכם את הנקודות העיקריות ב-5 משפטים"

- "מה הטרנדים המרכזיים שמופיעים פה?"

- "איזה נושאים חוזרים על עצמם?"

המודל קורא, מזהה דפוסים, ומפיק תובנות - עבודה שעד לפני כמה שנים דרשה אנליסט אנושי.

2.2.4 השוואה וניתוח

"השווה בין שלוש הצעות המחיר האלה מבחינת עלות, זמן אספקה ושירות, והמלץ על הספק המתאים ביותר לארגון שלנו."

המודל לא רק משווה - הוא **מנמק** את ההמלצה שלו על בסיס הקריטריונים שהגדרתם.

3.2.4 פתרון בעיות רב-שלבי

LLMs מודרניים יכולים לפתור בעיות שדורשות מספר שלבי חשיבה:

1. הבנת הבעיה

2. פירוק לתת-בעיות

3. פתרון כל תת-בעיה

4. שילוב התוצאות לפתרון כולל

"יש לנו 5 נציגי מכירות, 120 לידים חדשים החודש, וכל נציג יכול לטפל בממוצע ב-25 לידים בחודש. איך כדאי לחלק את הלידים בהתחשב בכך שנציג א' מתמחה בלקוחות ארגוניים, ב' ו-ג' בעסקים קטנים, ד' בסטארטאפים, וה' חדש ועדיין מתאמן?"

המודל יבנה תוכנית חלוקה מפורטת, יסביר את ההיגיון מאחוריה, ואפילו יתריע על בעיות פוטנציאליות.

4.2.4 תכנות וכתובת קוד

LLMs כמו GPT-4 [2] ו-Claude [3] מסוגלים לכתוב קוד תוכנה באיכות גבוהה. מנהל בלי רקע תכנותי יכול לבקש:

"תכתוב סקריפט Python שקורא קובץ Excel, מחשב את סכום המכירות לכל מוצר, ויוצר גרף עמודות"

והמודל יכתוב קוד מלא, מתועד, ומוכן להרצה.

השלכה עסקית: זה מוריד את המחסום לאוטומציה. משימות שבעבר דרשו מתכנת, היום יכולות להתבצע על ידי כל מנהל עם רעיון ברור.

3.4 המשמעות המשולבת: עובד דיגיטלי

כשמשלבים את שני הכוחות האלה - תקשורת טבעית ועיבוד לוגיקה מורכבת - מקבלים משהו חדש לחלוטין בעולם העסקים: **עובד דיגיטלי** שאפשר להדריך, לשאול שאלות, לתקן, ולשפר - בדיוק כמו עובד אנושי.

זה לא עוד כלי שמבצע משימה אחת קבועה. זה ישות דיגיטלית שיכולה:

- להבין הוראות מורכבות
- להתאים את עצמה למצבים שונים
- לשאול שאלות הבהרה
- ללמוד מדוגמאות שאתם נותנים
- להציע שיפורים

זו המהפכה האמיתית של LLMs.

5 נקודות החוזק: מה LLMs עושים טוב במיוחד

1.5 יצירתיות והפקת רעיונות

LLMs מצטיינים ביצירת תוכן חדש ומגוון:

- **תוכן שיווקי:** מודעות, פוסטים, דפי נחיתה, מיילים
- **תוכן טכני:** מסמכי דרישות, הצעות מחיר, תיעוד
- **רעיונות:** סיעור מוחות אוטומטי, זוויות חדשות לבעיות קיימות

דוגמה מעשית:

מנהלת שיווק בחברת B2B SaaS צריכה 10 כותרות שונות לקמפיין LinkedIn Ads שמקדם כלי ניהול פרויקטים לצוותי פיתוח. היא כותבת ל-ChatGPT:

"צור 10 כותרות לקמפיין לינקדאין למוצר ניהול פרויקטים לצוותי פיתוח תוכנה. המוצר חוסך 20% מזמן ההנהלה ומשפר שיתוף פעולה. קהל יעד: VP R&D וראשי צוותים. טון: מקצועי אך לא יבש, דגש על תוצאות עסקיות."

תוך שניות, היא מקבלת 10 אפשרויות, כל אחת בזווית שונה - ROI, פריון, איכות חיים, תחרותיות. היא יכולה לבחור, לשלב, או לבקש עוד וריאציות. **זמן שנחסך:** שעה של סיעור מוחות ומחשבה → 5 דקות.

2.5 הבנת הקשר ונואנסים

בניגוד לכלי אוטומציה מסורתיים שפועלים לפי כללים קשיחים, LLMs מבינים הקשר:

- הם "זוכרים" את כל השיחה ומתייחסים אליה
- הם מבינים כוונות, לא רק מילים
- הם מתאימים את התשובה לטון ולסגנון המבוקש

דוגמה מתמיכת לקוחות:

לקוח כותב: "המערכת שוב קרסה לי באמצע פעולה חשובה"
Chatbot מסורתי (מבוסס כללים) יזהה את המילה "קרסה" ויחזיר:

"אנא נסה לאתחל את המערכת ולנקות את הזיכרון הזמני."

סוכן LLM מבין את ההקשר הרגשי והעסקי:

"אני מבין את התסכול - זה בוודאי מעצבן כשזה קורה בעיצומה של עבודה חשובה. בואו נפתור את זה מהר. האם הבעיה קרתה בפעולה ספציפית? זה יעזור לי לאבחן את הגורם ולוודא שזה לא יקרה שוב."

ההבדל? אמפתיה, הקשר עסקי, וכוונה לפתרון ארוך טווח - לא רק טיפול תסמיני.

3.5 גמישות והתאמה

LLMs לא דורשים תכנות מראש לכל תרחיש. אפשר "ללמד" אותם "בדרך":

- **Few-Shot Learning** [4]: תן כמה דוגמאות, והמודל יבין את הדפוס
- **התאמה לסגנון:** "כתוב בסגנון פורמלי/חברי/טכני"
- **שינוי כיוון באמצע:** "לא, תשנה את הגישה ל..."

דוגמה - קטגוריזציה של פניות:

חברה מקבלת מאות פניות ביום לתמיכת לקוחות. היא רוצה לקטגר אותן אוטומטית ל-5 קטגוריות.

גישה מסורתית: שכירת מתכנת שיבנה מודל ML מותאם אישית, יאסוף נתוני אימון, ויקח שבועיים.

גישה LLM:

"קטגר את הפניות הבאות לאחת מחמש הקטגוריות: טכני, חיוב, שאלת מכירה, תלונה, בקשת פיצ'ר. הנה שלוש דוגמאות:
[דוגמה 1...] [דוגמה 2...] [דוגמה 3...]
עכשיו, קטגר את הפניות האלה: [רשימת פניות...]"

המודל יבין את הדפוס מהדוגמאות ויקטגר נכון 90-59% מהפניות.

זמן יישום: שבועיים → 30 דקות.

4.5 עבודה עם שפות מרובות

LLMs מודרניים כמו GPT-4, Claude, ו-Gemini מדברים עשרות שפות בצורה שוטפת. זה פותח אפשרויות:

- תרגום אוטומטי איכותי (מעבר לתרגום מילה במילה - הבנת הקשר תרבותי)
- תמיכת לקוחות רב-לשונית בלי צוות עצום
- יצירת תוכן בשפות מרובות באופן מיידי

דוגמה: חברה ישראלית שמוכרת לאירופה צריכה לתרגם מסמך טכני מעברית לגרמנית, צרפתית, וספרדית. במקום שלושה מתרגמנים מקצועיים ומספר ימים, Claude מתרגם את שלושת הגרסאות תוך דקות, עם שמירה על טרמינולוגיה טכנית עקבית.

6 נקודות החולשה: מה LLMs לא עושים טוב

1.6 הזיות (Hallucinations)

זוהי אולי החולשה הקריטית ביותר של LLMs: הנטייה "להמציא" מידע [5]. זכרו - LLM הוא מכונת השלמת דפוסים. הוא לא מחפש מידע במסד נתונים; הוא מחזה את המשך הסביר ביותר. לפעמים, אם המידע הנכון לא קיים בזיכרון הסטטיסטי שלו, המודל ייצר עובדות שנשמעות מהימנות - אבל שגויות לחלוטין.

דוגמה מסוכנת:

עורך דין ביקש מ-ChatGPT לספק תקדימים משפטיים לתמיכה בתביעה. המודל מסר רשימה של שישה תקדימים, כולל שמות תיקים, מספרי תיק, ותאריכים. הכל נראה לגיטימי. הבעיה? **אף אחד מהתקדימים לא היה אמיתי.** ChatGPT המציא אותם, כי הם נשמעו הגיוניים בהקשר.

המשפט הסתיים בסנקציות חמורות על עורך הדין.

למה זה קורה?

LLM נועד "להישמע" מהימן ורהוט. אין לו מנגנון פנימי שאומר "אני לא יודע". במקום זאת, הוא ממשיך לייצר את המשך הכי סביר, גם אם זה לא מבוסס עובדות.

השלכות עסקיות:

- אין לסמוך על LLM לעובדות ללא אימות. תמיד יש לבדוק מידע קריטי.
- מתאים ליצירת רעיונות וטיוטות ראשוניות, פחות מתאים לדוחות עובדתיים סופיים.
- חובה לשלב בקרה אנושית במערכות קריטיות.

2.6 חוסר עדכניות (Knowledge Cutoff)

כל LLM נאמן עד תאריך מסוים - ה"knowledge cutoff" שלו. למשל:

- GPT-4 (גרסה מ-4202): נתונים עד אפריל 2023

- Claude 3.5 Sonnet: נתונים עד אפריל 2024

המשמעות: המודל לא יודע כלום על אירועים, מוצרים, טכנולוגיות, או שינויים שקרו אחרי התאריך הזה.

דוגמה:

OFC שואל את GPT-4:

"מה שער הדולר מול השקל היום?"

התשובה תהיה מבוססת על נתונים ישנים, או לחלופין - הזיה.

פתרונות:

- שימוש במודלים עם גישה לאינטרנט (כמו ChatGPT Plus עם browsing mode)
 - שילוב RAG (Retrieval-Augmented Generation) [6] - הזרקת מידע עדכני למודל
 - שימוש ב-APIs חיצוניים שהסוכן יכול לקרוא להם
- הנפקות עסקיות:** בתחומים דינמיים (פיננסים, חדשות, נתונים תפעוליים), אין להסתמך על LLM לבדו. יש לספק לו מידע עדכני או לשלב אותו עם מקורות מידע חיים.

3.6 עלויות - לא זניח

שימוש ב-LLMs דרך API עולה כסף, והעלות יכולה להפתיע ארגונים שלא תכננו נכון. מודלים מתומחרים לפי **טוקנים**:

Model	Input (\$/1M tokens)	Output (\$/1M tokens)
GPT-4 Turbo	10.00	30.00
GPT-3.5 Turbo	0.50	1.50
Claude 3.5 Sonnet	3.00	15.00
Claude 3 Haiku	0.25	1.25
Gemini 1.5 Pro	1.25	5.00

טבלה 1: מחירי מודלים נפוצים (נכון למרץ 2024)

דוגמת חישוב:

נניח שחברה משתמשת ב-GPT-4 Turbo לתמיכת לקוחות. כל שיחה:

- קלט ממוצע: 2,000 טוקנים (הקשר + שאלת הלקוח)

- פלט ממוצע: 800 טוקנים (תשובה)

עלות לשיחה:

$$\begin{aligned}\text{Cost} &= \left(\frac{2,000}{1,000,000} \times 10 \right) + \left(\frac{800}{1,000,000} \times 30 \right) \\ &= 0.02 + 0.024 \\ &= \$0.044\end{aligned}\tag{1}$$

זה נשמע זניח, נכון? אבל אם יש 10,000 שיחות בחודש:

$$\text{Monthly Cost} = 10,000 \times 0.044 = \$440\tag{2}$$

ואם זה גדל ל-100,000 שיחות בחודש (חברה גדולה):

$$\text{Monthly Cost} = 100,000 \times 0.044 = \$4,400/\text{month} = \$52,800/\text{year}\tag{3}$$

אסטרטגיות חיסכון:

- שימוש במודלים זולים יותר למשימות פשוטות (GPT-3.5 במקום GPT-4)
- אופטימיזציה של prompts להיות קצרים וממוקדים
- Caching - שמירת תשובות לשאלות נפוצות
- שימוש במודלים self-hosted (למשל Llama 3) לנפחים גדולים

4.6 חוסר שקיפות (Black Box)

LLMs הם "קופסה שחורה". כשהם נותנים תשובה, אי אפשר לדעת בדיוק למה הם הגיעו למסקנה הזו. אין "שרשרת הוכחה".

בעיה עסקית:

- **רגולציה:** בתחומים מוסדרים (בנקאות, בריאות), לפעמים נדרש להסביר החלטות. "כי ה-AI אמר" לא מספיק.
- **אמון:** מנהלים מתקשים לסמוך על מערכת שלא יכולה להסביר את ההיגיון שלה.
- **Bias:** קשה לזהות ולתקן הטיות כשלא רואים את תהליך החשיבה.

פתרונות חלקיים:

- שימוש ב-Chain-of-Thought prompting [7] - לבקש מהמודל להסביר את הצעדים שלו
- שילוב מערכות explainable AI משלימות
- בקרה אנושית במקרים קריטיים

7 נוסחאות מנהליות להערכת LLMs

1.7 מדד ROI של יישום AI

לפני השקעה בכלי AI, חשוב לחשב את התשואה הצפויה על ההשקעה [8], [9].

נוסחת ROI בסיסית:

$$(4) \quad ROI = \frac{\text{עלות מנוי IA} - (\text{עלות שעת עבודה} \times \text{שעות נחסכות})}{\text{עלות מנוי IA}} \times 100\%$$

דוגמה מעשית:

צוות תמיכת לקוחות בן 5 אנשים משתמש ב-Claude לטיפול בפניות שגרתיות.

נתונים:

- כל נציג מטפל ב-30 פניות ביום
- לפני AI: זמן ממוצע לפנייה = 15 דקות
- עם AI (סיוע בכתיבה, חיפוש מידע, תבניות): זמן ממוצע = 10 דקות
- חיסכון לפנייה: 5 דקות
- ימי עבודה בחודש: 22
- עלות שעת עבודה ממוצעת: 100 ש"ח
- עלות מנוי Claude Pro לכל נציג: \$20/חודש = 75 ש"ח

חישוב:

שעות נחסכות בחודש:

$$\begin{aligned} \text{ימים} \times 22 \times \text{שעות} \times \frac{5}{60} \times \text{פניות/יום} \times 30 \times \text{נציגים} \times 5 &= \text{שעות נחסכות} \\ &= 5 \times 30 \times 0.0833 \times 22 \\ &= 275 \text{ שעות} \end{aligned} \quad (5)$$

ערך החיסכון:

$$(6) \quad \text{ש"ח} = 275 \times 100 = 27,500 \quad \text{ערך}$$

עלות מנוי:

$$(7) \quad \text{ש"ח} = 5 \times 75 = 375 \quad \text{עלות}$$

ROI:

$$(8) \quad ROI = \frac{27,500 - 375}{375} \times 100\% = 7,233\%$$

משמעות: על כל שקל שהושקע ב-AI, החברה חוסכת 72 שקל. זו תשואה עצומה.
נוסחה מורחבת (כוללת עלויות נוספות):

$$(9) \quad ROI = \frac{\text{עלות כוללת} - \text{תועלת כוללת}}{\text{עלות כוללת}} \times 100\%$$

כאשר:

- **תועלת כוללת** = חיסכון בשעות + הגדלת מכירות + שיפור שביעות רצון (בערך כסף)
- **עלות כוללת** = מנויים + הטמעה + הדרכה + תחזוקה

2.7 נוסחת עלות טוקנים

להבנת העלות התפעולית של שימוש ב-API:

$$(10) \quad \text{עלות שיחה} = \left(\frac{\text{טוקני קלט}}{1,000,000} \times \text{מחיר קלט} \right) + \left(\frac{\text{טוקני פלט}}{1,000,000} \times \text{מחיר פלט} \right)$$

דוגמה:

סוכן AI לניתוח משוב לקוחות משתמש ב-GPT-4 Turbo:

- מחיר קלט: \$10 למיליון טוקנים

- מחיר פלט: \$30 למיליון טוקנים

- כל ניתוח כולל: 3,000 טוקני קלט + 1,200 טוקני פלט

- נפח: 5,000 ניתוחים בחודש

עלות לניתוח בודד:

$$\begin{aligned} \text{Cost}_{\text{single}} &= \left(\frac{3,000}{1,000,000} \times 10 \right) + \left(\frac{1,200}{1,000,000} \times 30 \right) \\ &= 0.03 + 0.036 \\ &= \$0.066 \end{aligned} \quad (11)$$

עלות חודשית:

$$(12) \quad \text{Cost}_{\text{monthly}} = 5,000 \times 0.066 = \$330$$

שימוש מעשי: נוסחה זו מאפשרת למנהלים לחזות עלויות על בסיס נפח העסקאות הצפוי ולהחליט בין מודלים שונים.

3.7 Break-Even Analysis

מתי כדאי לעבור משירות cloud לפתרון self-hosted?

$$(13) \quad \text{נקודת איזון (חודשים)} = \frac{\text{עלות הקמה חד-פעמית}}{\text{חיסכון חודשי}}$$

דוגמה:

חברה משתמשת ב-GPT-4 API ומשלמת \$5,000/חודש. היא שוקלת להקים שרת פנימי עם Llama 3 70B.

עלויות self-hosted:

- שרת + GPU: \$15,000

- הקמה ותצורה: \$5,000

- עלות חודשית (חשמל, תחזוקה, כ"א): \$2,000

חיסכון חודשי:

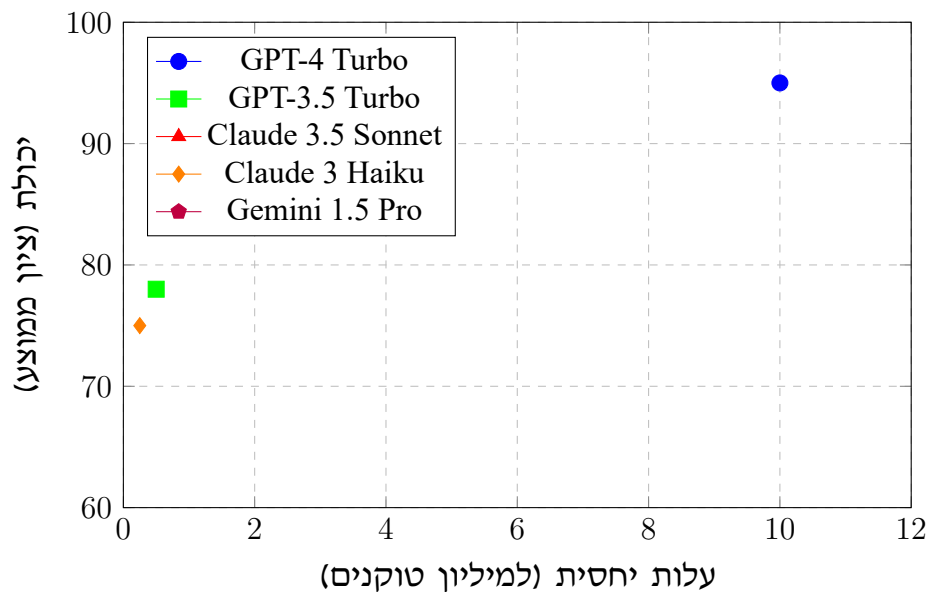
$$(14) \quad \text{חיסכון} = 5,000 - 2,000 = \$3,000$$

נקודת איזון:

$$(15) \quad \text{Break-even} = \frac{15,000 + 5,000}{3,000} = 6.67 \text{ חודשים}$$

משמעות: תוך כ-7 חודשים ההשקעה תשתלם, ומשם והלאה החברה תחסוך \$3,000/חודש.

8 השוואת מודלים: תרשים עלות מול יכולות



איור 2: השוואת מודלים מובילים - עלות מול יכולות

תרשים 2 מציג את הטרייד-אוף בין עלות ליכולות:

- **GPT-4 Turbo:** היקר והמסוגל ביותר - מתאים למשימות קריטיות ומורכבות

- **Claude 3.5 Sonnet:** איזון מצוין - 29% מהיכולת ב-3% מהמחיר

- **Gemini 1.5 Pro:** חלופה חזקה במחיר נוח

- **GPT-3.5 ו-Haiku:** למשימות פשוטות בנפח גבוה

אסטרטגיה מומלצת: שימוש במודלים שונים לצרכים שונים. משימות פשוטות (סיכום, עריכה) במודלים זולים; משימות מורכבות (ניתוח, החלטות) במודלים מתקדמים.

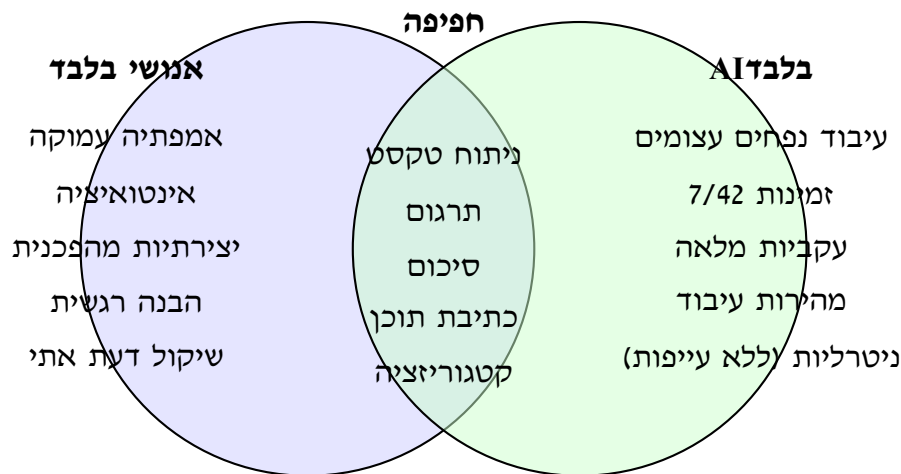
9 תרשים Venn: חפיפה בין יכולות אנושיות ו-AI

תרשים 3 ממחיש היכן כדאי להשתמש ב-AI, והיכן האדם עדיין בלתי תחליפי:

אזור החפיפה - משימות שבהן AI יכול לסייע או אפילו להחליף אדם:

- עיבוד ואנליזת טקסט

- תרגום והתאמה לשונית



איור 3: חפיפה בין יכולות אנושיות ויכולות IA

- סיכום מסמכים ארוכים
- כתיבת תוכן סטנדרטי
- מיון וקטגוריזציה

אזור אנושי בלבד - משימות שבהן האדם עדיין חיוני:

- החלטות אסטרטגיות מורכבות
- מצבים הדורשים אמפתיה אמיתית
- יצירתיות פורצת דרך (לא הרכבה של דפוסים)
- שיקול דעת אתי ומוסרי
- הבנה עמוקה של הקשר ארגוני ותרבותי

אזור AI בלבד - משימות שבהן AI עדיף על אדם:

- עיבוד נפחי מידע עצומים
- זמינות מתמדת ללא הפסקה
- עקביות מוחלטת בביצוע
- מהירות ותגובה מיידית
- ניטרליות (אין השפעת עייפות, רגשות חולפים)

המסקנה המנהלית: הגישה האופטימלית היא **שיתוף פעולה**, לא תחרות. AI כעוזר שמשחרר את האדם ממשימות שגרתיות, ומאפשר לו להתמקד ביתרונות הייחודיים שלו.

01 דוגמאות מעשיות: LLMs בעבודה

1.01 מנהלת שיווק: תכנון ויצירת קמפיין

תרחיש:

רינה, מנהלת שיווק בחברת SaaS, צריכה להשיק קמפיין למוצר חדש. לפניה:

- בניית אסטרטגיה

- כתיבת תוכן לערוצים שונים
- יצירת וריאציות למבחני A/B

השימוש ב-ChatGPT:

שלב 1 - סיעור מוחות אסטרטגי:

"אני משיקה מוצר SaaS לניהול פרויקטים לצוותי שיווק. קהל יעד: מנהלי שיווק B2B. הערך המרכזי: חיסכון של 10 שעות שבועיות באוטומציה. תן לי 5 זוויות שונות לקמפיין."

ChatGPT מציע:

1. "עשה יותר עם פחות" - דגש על יעילות
2. "מזמן לאסטרטגיה" - שחרור מדיווחים לחשיבה
3. "צוות קטן, תוצאות גדולות" - יתרון לסטארטאפים
4. "ROI ברור מיום ראשון"
5. "כל הכלים במקום אחד" - פשטות

שלב 2 - פיתוח זווית נבחרת:

רינה בחרה בזווית 2. היא ממשיכה:

"תפתח את זווית 'מזמן לאסטרטגיה'. כתוב לי 3 כותרות ל-LinkedIn Ads, 2 פוסטים קצרים, ודף נחיתה בסגנון חברי-מקצועי."

תוך דקה וחצי, רינה מקבלת תוכן מלא. היא משקיעה עוד 20 דקות בעריכה ובהתאמה למותג.

שלב 3 - וריאציות ל-A/B Testing:

"צור עוד 5 וריאציות של הכותרת הראשונה - אחת עם מספרים, אחת עם שאלה, אחת עם אמוג'י, אחת קצרה מאוד, ואחת ארוכה ומפורטת."

תוצאה:

- זמן שנחסך: 4-5 שעות של כתיבה וחשיבה
- איכות: טובה מאוד (לאחר עריכה קלה)
- יצירתיות: זוויות שרינה לא חשבה עליהן

2.01 סמנכ"ל כספים: ניתוח דוחות מורכבים

תרחיש:

יוסי, CFO של חברה ציבורית, מקבל דוח רבעוני של 80 עמודים מחברת ביקורת. הוא צריך:

- להבין מגמות עיקריות

- לזהות חריגות
- להכין סיכום להנהלה

השימוש ב-Claude (יכולת העלאת קבצים):

יוסי מעלה את ה-PDF ל-Claude ושואל:

"נתח את הדוח הזה וספק: 1. 3 המגמות החיוביות העיקריות 2. 3 נקודות הדאגה העיקריות 3. חריגות משמעותיות מול רבעון קודם 4. המלצות ראשוניות"

Claude קורא את כל הדוח (למרות אורכו) ותוך 2 דקות מספק:

מגמות חיוביות:

1. צמיחה של 21% בהכנסות חוזרות (ARR)
2. שיפור של 5% ב-Gross Margin
3. ירידה של 81% בעלויות רכישת לקוח (CAC)

נקודות דאגה:

1. עלייה של 22% ב-churn rate בסגמנט BMS
2. הארכת מחזור המכירה מ-45 ל-63 ימים
3. עלייה בהוצאות R&D ללא השקה מקבילה

חריגות:

- סעיף "הוצאות שונות" קפץ פי 3 - דורש הבהרה
- נכס בלתי מוחשי חדש בסכום חריג

המלצות:

- לחקור את עליית ה-churn - אפשרי שינוי בתחרות או בטיב שירות
 - לברר את הארכת מחזור המכירה - האם בעיה במכירות או בכלכלה הכללית
 - לדרוש פירוט על "הוצאות שונות"
- יוסי עובר על הניתוח, משווה למסמכים נוספים, ומעמיק בנקודות הקריטיות. אבל הסינון הראשוני חסך לו שעה וחצי של קריאה צפופה.

שימוש נוסף - שאלות המשך:

יוסי ממשיך לשאול:

"על בסיס הדוח, האם נוכל לעמוד ביעדי ה-EBITDA לסוף השנה?"

Claude מנתח את הנתונים ומספק תשובה מנומקת עם הנחות.

3.01 מנהלת משאבי אנוש: סינון קורות חיים

תרחיש:

מיכל, מנהלת גיוס, קיבלה 150 קורות חיים לתפקיד Product Manager. דרישות התפקיד:

- ניסיון של 3-5 שנים בניהול מוצר B2B SaaS

- רקע טכני - יתרון למי שעבד כמפתח

- ניסיון בעבודה עם צוותים מבוזרים

- אנגלית שוטפת

גישה מסורתית: מיכל הייתה צריכה לקרוא כל קו"ח ידנית - זמן: 5-6 שעות.

גישה AI:

מיכל כותבת סקריפט Python פשוט (בעזרת ChatGPT) שקורא את כל הקבצים ושולח כל קו"ח ל-GPT-4 עם ה-prompt:

"דרג קורות חיים אלה לתפקיד reganaM tcudorP ב-B2B SaaS. דרישות: - 3-5 שנות ניסיון PM - רקע טכני (יתרון) - עבודה עם צוותים מבוזרים - אנגלית שוטפת

דרג: TIF GNORTS / TIF DOOG / EBYAM / TIF ON ונמק בקצרה."

המערכת מעבדת את 150 הקורות חיים תוך 10 דקות ומחזירה:

- 12 מועמדים STRONG FIT

- 28 מועמדים GOOD FIT

- 45 מועמדים MAYBE

- 65 מועמדים NO FIT

מיכל קוראת רק את 12 ה-STRONG FIT ואת 28 ה-GOOD FIT - סך הכל 40 קורות חיים - זמן: שעה.

תוצאה: חיסכון של 4-5 שעות, תוך שמירה על איכות הסינון.

הערה חשובה: מיכל לא סומכת באופן עיוור על ה-AI. היא עדיין קוראת בעצמה את הקורות חיים הרלוונטיים. ה-AI משמש ככלי סינון ראשוני, לא כקובע סופי.

11 תרגילים

1.11 תרגילים תיאורטיים

תרגיל 1.11. חישוב ROI של הטמעת כלי AI

הנך מנהל/ת מחלקת תוכן בחברת e-commerce. הצוות שלך (8 כותבים) מייצר 40 מאמרים בחודש. כל מאמר לוקח כיום 4 שעות עבודה.

אתה שוקל להטמיע ChatGPT Plus (\$20/חודש לכל כותב) שלדעתך יקצר את הזמן ל-2.5 שעות למאמר.

נתונים:

- עלות שעת עבודה ממוצעת: 120 ש"ח

- ימי עבודה בחודש: 22

שאלות:

- (א) חשב את ה-ROI החודשי
- (ב) כמה זמן יעבור עד שההשקעה תשתלם (בהנחה שיש גם עלות הדרכה חד-פעמית של 5,000 ש"ח)?
- (ג) האם תמליץ על ההטמעה? נמק.

תרגיל 2.11. זיהוי משימות מתאימות ובלתי מתאימות ל-LLM

בחן את המשימות הבאות בארגון שלך וסווג כל אחת:

- א. מתאימה מאוד ל-LLM (יכול להחליף אדם לחלוטין)
- ב. מתאימה חלקית (יכול לסייע לאדם)
- ג. לא מתאימה (האדם חיוני)

משימות:

1. כתיבת מדיניות פרטיות לאתר
2. קבלת החלטה על פיטורי עובד
3. תרגום חוזה מאנגלית לעברית
4. בניית אסטרטגיה עסקית ל-3 שנים
5. סיכום פרוטוקול ישיבה
6. ניהול משא ומתן עם לקוח כועס
7. ניתוח משוב לקוחות מסקר (NPS)
8. בחירת ספק אסטרטגי לטווח ארוך
9. יצירת תבניות מייל לתמיכת לקוחות
10. ראיון עומק עם מועמד לתפקיד בכיר

נמק את הבחירות שלך.

תרגיל 3.11. השוואה בין GPT-4 ל-Claude לצורכי הארגון

הנך CTO של חברת FinTech. אתה צריך לבחור מודל LLM מרכזי לארגון.

תרחישי שימוש מרכזיים:

- ניתוח מסמכים משפטיים ופיננסיים (דוחות, חוזים)
- תמיכת לקוחות אוטומטית
- סיוע למפתחים בכתיבת קוד
- יצירת תוכן שיווקי

קריטריונים:

- עלות (נפח גבוה - 50 מיליון טוקנים/חודש)
- דיוק במסמכים ארוכים
- יכולת קוד
- תמיכה בעברית

- אבטחה ופרטיות

בנה טבלת השוואה ובחר מודל. נמק.

תרגיל 4.11. ניתוח כישלון AI - מה השתבש?

חברת ביטוח הטמיעה סוכן AI לתמיכת לקוחות. אחרי חודש, היא גילתה שהסוכן:

- נתן מידע שגוי על כיסוי ביטוחי ב-51% מהמקרים
- "המציא" פוליסות שלא קיימות
- לא הצליח להבין שאלות בעברית עם מונחים ביטוחיים

שאלות:

- (א) מה הסיבות האפשריות לכישלון?
- (ב) איזה חולשות של LLMs באו לידי ביטוי?
- (ג) איך היית ממליץ לתקן את המערכת?
- (ד) האם היית ממליץ להפסיק את הפרויקט לחלוטין? למה?

תרגיל 5.11. בניית Business Case ליישום LLM

הנך מנהל/ת מחלקת מכירות עם 20 נציגי מכירות. אתה רוצה להטמיע סוכן AI שיסייע להם במשימות הבאות:

- כתיבת הצעות מחיר מותאמות אישית
- מענה על שאלות טכניות של לקוחות (באמצעות RAG על מסמכי המוצר)
- סיכום שיחות עם לקוחות ומעקב אוטומטי

כתוב Business Case שכולל:

- (א) **בעיה עסקית:** מה הכאב הנוכחי?
- (ב) **פתרון מוצע:** מה בדיוק תטמיע?
- (ג) **תועלת:** מה יהיו היתרונות המדידים?
- (ד) **עלויות:** כמה זה יעלה? (כולל טכנולוגיה, הטמעה, הדרכה)
- (ה) **ROI:** תוך כמה זמן ההשקעה תשתלם?
- (ו) **סיכונים:** מה עלול להשתבש?
- (ז) **המלצה:** האם כדאי להתקדם?

2.11 תרגילי קוד Python

תרגיל 6.11. חישוב עלות שימוש חודשית ב-API

כתוב תוכנית Python שמקבלת:

- מספר שיחות/פעולות חודשיות
- ממוצע טוקני קלט לפעולה
- ממוצע טוקני פלט לפעולה

- מחיר קלט למיליון טוקנים

- מחיר פלט למיליון טוקנים

התוכנית צריכה לחשב ולהדפיס:

(א) עלות לפעולה בודדת

(ב) עלות חודשית כוללת

(ג) עלות שנתית

(ד) השוואה: כמה היה עולה באותו נפח עם מודל אחר (שהמשתמש מזין מחירים שלו)

דוגמת הרצה:

```
Enter monthly operations: 50000
Enter avg input tokens: 2000
Enter avg output tokens: 800
Enter input price ($/1M tokens): 10
Enter output price ($/1M tokens): 30
```

=== Cost Analysis ===

Cost per operation: \$0.044

Monthly cost: \$2,200.00

Annual cost: \$26,400.00

Compare with another model? (y/n): y

Enter input price (\$/1M tokens): 0.5

Enter output price (\$/1M tokens): 1.5

Alternative model monthly cost: \$112.00

You would save: \$2,088.00/month (94.9%)

21 סיכום הפרק

במהלך פרק זה עברנו מסע מקיף בעולם מודלי השפה הגדולים:

למדנו מהם LLMs:

- מכונות השלמת דפוסים שאומנו על כמויות עצומות של טקסט

- לא מסדי נתונים, אלא מודלים סטטיסטיים מורכבים שיוצרים תוכן חדש

הכרנו את שני הכוחות העל:

- **תקשורת טבעית:** סוף סוף, מכונות שמבינות אותנו

- **עיבוד לוגיקה מורכבת:** לא רק ממשק נוח, אלא חשיבה אמיתית

זיהינו נקודות חוזק:

- יצירתיות והפקת רעיונות

- הבנת הקשר ונואנסים

- גמישות והתאמה

- יכולת רב-לשונית

למדנו על נקודות חולשה קריטיות:

- הזיות - המצאת מידע שנשמע מהימן

- חוסר עדכניות - מוגבל לתאריך אימון

- עלויות - לא זניח בקנה מידה

- חוסר שקיפות - קופסה שחורה

רכשנו כלים מנהליים:

- נוסחת ROI להערכת השקעה

- נוסחת עלות טוקנים לתכנון תקציב

- ניתוח break-even להשוואת אלטרנטיבות

ראינו דוגמאות מהשטח:

- מנהלת שיווק שחוסכת שעות ביצירת תוכן

- CFO שמנתח דוחות מורכבים במהירות

- מנהלת HR שמסננת קורות חיים ביעילות

המסר המרכזי:

LLMs הם לא קסם, והם לא מושלמים. אבל כשמבינים את היכולות והמגבלות שלהם, ומיישמים אותם בצורה מושכלת - הם כלי עסקי עוצמתי שמשנה משחק.

ההצלחה לא תלויה בטכנולוגיה בלבד, אלא באופן שבו מנהלים מבינים, מתכננים ומיישמים אותה. פרק זה סיפק לכם את היסודות - בפרקים הבאים נצלול עמוק יותר לאקוסיסטם, לארכיטקטורה, וליישום מעשי.

מקורות והמלצות לקריאה נוספת

1. Attention Is All You Need - מאמר היסוד על ארכיטקטורת Transformer

2. OpenAI GPT-4 Technical Report - תיעוד רשמי של GPT-4

3. Anthropic Claude Documentation - מדריך מקיף למודלי Claude

4. The Economics of Large Language Models - ניתוח עלויות ו-ROI

5. Prompt Engineering Guide - מדריך מעמיק לכתיבת prompts

פתרונות מלאים לתרגילים

פתרון תרגיל 1: חישוב ROI

נתונים:

- 8 כותבים
- 40 מאמרים/חודש
- זמן נוכחי: 4 שעות/מאמר
- זמן עתידי: 2.5 שעות/מאמר
- עלות שעה: 120 ש"ח
- עלות ChatGPT Plus: \$20/חודש = 75 ש"ח

חישוב:

סך שעות נוכחי:

$$(16) \quad 40 \times 4 = 160 \text{ שעות/חודש}$$

סך שעות עתידי:

$$(17) \quad 40 \times 2.5 = 100 \text{ שעות/חודש}$$

שעות נחסכות:

$$(18) \quad 160 - 100 = 60 \text{ שעות/חודש}$$

ערך החיסכון:

$$(19) \quad 60 \times 120 = 7,200 \text{ ש"ח/חודש}$$

עלות מנוי:

$$(20) \quad 8 \times 75 = 600 \text{ ש"ח/חודש}$$

ROI חודשי:

$$(21) \quad \frac{7,200 - 600}{600} \times 100\% = 1,100\%$$

זמן החזר השקעה:

חיסכון נטו חודשי:

$$(22) \quad 7,200 - 600 = 6,600 \text{ ש"ח}$$

עלות הדרכה:

$$(23) \quad 5,000 \text{ ש"ח}$$

זמן החזר:

$$(24) \quad \frac{5,000}{6,600} = 0.76 \text{ ימים} \approx 23 \text{ חודשים}$$

המלצה: בהחלט כדאי! ה-ROI עצום, וההשקעה מתשלמת תוך פחות מחודש.

פתרון תרגיל 2: זיהוי משימות

1. **כתיבת מדיניות פרטיות:** ב' (מתאים חלקית) - AI יכול לכתוב טיוטה מצוינת, אבל עורך דין צריך לאשר.
2. **החלטה על פיטורים:** ג' (לא מתאים) - דורש שיקול דעת אנושי, אמפתיה, הבנה של הקשר ארגוני.

3. **תרגום חוזה:** ב' (מתאים חלקית) - תרגום ראשוני מצוין, אבל חוזה דורש עריכה משפטית אנושית.
4. **אסטרטגיה ל-3 שנים:** ב'-ג' - AI יכול לסייע בניתוח וברעיונות, אבל החלטה סופית דורשת אדם.
5. **סיכום פרוטוקול:** א' (מתאים מאוד) - משימה מובנית שה-AI עושה מצוין.
6. **משא ומתן עם לקוח כועס:** ג' (לא מתאים) - דורש אמפתיה אמיתית ושיקול דעת דינמי.
7. **ניתוח סקר NPS:** א'-ב' (מתאים מאוד) - AI מצוין בזיהוי דפוסים ומגמות.
8. **בחירת ספק אסטרטגי:** ב'-ג' - AI יכול לנתח, אבל החלטה כזו דורשת שיקולים רבים שאדם מבין טוב יותר.
9. **תבניות מייל:** א' (מתאים מאוד) - משימה שה-AI עושה מצוין.
10. **ראיון לתפקיד בכיר:** ג' (לא מתאים) - דורש הבנה עמוקה של תרבות ארגונית ושיקול דעת אנושי.

פתרון תרגיל 6: קוד nohtyP לחישוב עלויות

להלן קוד מלא ומתועד:

Listing 1: ילדומל API תוילוע ןובשחמ

```

1  """
2  LLM API Cost Calculator
3  Calculates monthly and annual costs for LLM API usage
4  """
5
6  def calculate_cost(operations, input_tokens, output_tokens,
7                    input_price, output_price):
8
9      """
10     Calculate cost per operation and total costs
11
12     Args:
13         operations: Number of monthly operations
14         input_tokens: Average input tokens per operation
15         output_tokens: Average output tokens per operation
16         input_price: Price per 1M input tokens ($)
17         output_price: Price per 1M output tokens ($)
18
19     Returns:
20         dict: Cost breakdown
21     """
22     # Cost per operation
23     cost_per_op = (

```

```

23         (input_tokens / 1_000_000) * input_price +
24         (output_tokens / 1_000_000) * output_price
25     )
26
27     # Total costs
28     monthly_cost = operations * cost_per_op
29     annual_cost = monthly_cost * 12
30
31     return {
32         'cost_per_operation': cost_per_op,
33         'monthly_cost': monthly_cost,
34         'annual_cost': annual_cost
35     }
36
37 def compare_models(operations, input_tokens, output_tokens,
38                   modell1_prices, modell2_prices):
39     """
40     Compare costs between two models
41
42     Args:
43         operations: Number of monthly operations
44         input_tokens: Average input tokens
45         output_tokens: Average output tokens
46         modell1_prices: (input_price, output_price) for model 1
47         modell2_prices: (input_price, output_price) for model 2
48
49     Returns:
50         dict: Comparison results
51     """
52     cost1 = calculate_cost(operations, input_tokens, output_tokens,
53                           modell1_prices[0], modell1_prices[1])
54     cost2 = calculate_cost(operations, input_tokens, output_tokens,
55                           modell2_prices[0], modell2_prices[1])
56
57     savings = cost1['monthly_cost'] - cost2['monthly_cost']
58     savings_pct = (savings / cost1['monthly_cost']) * 100
59
60     return {
61         'modell1_monthly': cost1['monthly_cost'],
62         'modell2_monthly': cost2['monthly_cost'],
63         'monthly_savings': savings,

```

```

64         'savings_percentage': savings_pct
65     }
66
67 def main():
68     """Main interactive calculator"""
69     print("=== LLM API Cost Calculator ===\n")
70
71     # Get user input
72     operations = int(input("Enter monthly operations: "))
73     input_tokens = int(input("Enter avg input tokens: "))
74     output_tokens = int(input("Enter avg output tokens: "))
75     input_price = float(input("Enter input price ($/1M tokens): "))
76     output_price = float(input("Enter output price ($/1M tokens): "))
77
78     # Calculate costs
79     costs = calculate_cost(operations, input_tokens, output_tokens,
80                           input_price, output_price)
81
82     # Display results
83     print("\n=== Cost Analysis ===")
84     print(f"Cost per operation: ${costs['cost_per_operation']:.4f}")
85     print(f"Monthly cost: ${costs['monthly_cost']:, .2f}")
86     print(f"Annual cost: ${costs['annual_cost']:, .2f}")
87
88     # Compare with another model?
89     compare = input("\nCompare with another model? (y/n): ")
90     if compare.lower() == 'y':
91         alt_input_price = float(
92             input("Enter input price ($/1M tokens): ")
93         )
94         alt_output_price = float(
95             input("Enter output price ($/1M tokens): ")
96         )
97
98         comparison = compare_models(
99             operations, input_tokens, output_tokens,
100             (input_price, output_price),
101             (alt_input_price, alt_output_price)
102         )
103
104     print(f"\nAlternative model monthly cost: ")

```



```

105         f"${comparison['model2_monthly']:, .2f}")
106     print(f"You would save: "
107           f"${comparison['monthly_savings']:, .2f}/month "
108           f"({comparison['savings_percentage']:.1f}%)")
109
110 if __name__ == "__main__":
111     main()

```

הרצת הקוד:

- שמור את הקוד בקובץ `yp.rotaluclac_tsoc_mll`
- הרץ: `yp.rotaluclac_tsoc_mll nohtyp`
- עקוב אחר ההנחיות האינטראקטיביות
- תרגיל מורחב: הרחב את הקוד כך שיתמודך ב:

1. שמירת תוצאות לקובץ CSV
2. ויזואליזציה של השוואת עלויות (גרף)
3. חישוב break-even עבור מעבר בין מודלים
4. תמיכה בהשוואה של יותר משני מודלים

סוף הפרק הראשון

בפרק הבא נצלול לאקוסיסטם הבינה המלאכותית - מפת הכלים, הספקים והטכנולוגיות שמנהל מודרני צריך להכיר.

מקורות

- 1 A. Vaswani et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- 2 OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.
- 3 Anthropic, "Claude 3 model card," 2024. [Online]. Available: <https://www.anthropic.com/claude>
- 4 T. Brown et al., "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, 1877–1901, 2020.
- 5 L. Huang et al., "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, 2024, arXiv:2311.05232. doi: [10.1145/3703155](https://doi.org/10.1145/3703155)

- 6 P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive nlp tasks,” *Advances in Neural Information Processing Systems*, vol. 33, 9459–9474, 2020.
- 7 J. Wei et al., “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in Neural Information Processing Systems*, vol. 35, 24824–24837, 2022.
- 8 E. Brynjolfsson, D. Li, and L. R. Raymond, “Generative ai at work,” *National Bureau of Economic Research*, 2023.
- 9 G. Almousa, R. Lovelace, V. L. Ziegler, et al., “The economic implications of large language model selection on earnings and return on investment: A decision theoretic model,” *arXiv preprint arXiv:2405.17637*, 2024.