

# מיני פרויקט שחמט - חיזוי תוצאות משחקים

## מסמך מטרות ותכנון מפורט

### המטרה הכללית

פיתוח פרויקט קטן ומתוכנן היטב שמשמש כהכנה מעשית לפרויקט הגמר בבינה מלאכותית, תוך שילוב למידת כלים טכניים חיוניים שנלמדים במקביל בקורסים אקדמיים.

### המטרות הספציפיות

#### מעשית Machine Learning למידת A: מטרה

- ואופטימיזציה, hyperparameters, הבנת עקרונות עבודה - **Random Forest** התנסות עם אלגוריתם
- Random Forest מול Logistic Regression ו Decision Trees - **השוואה בין אלגוריתמים**
- **מלא** - מניקוי נתונים ועד הערכת מודל ML **תהליך**
- **הבנת עקרונות חיזוי** - features, target variables, train/test split

#### SQL-שליטה מעשית B: מטרה

- בהתאם לקורס "מסדי נתונים" שנלמד במקביל - **PostgreSQL** עבודה עם
- **יצירת מסד נתונים מובנה** - טבלאות עם קשרים נכונים (שחקנים, משחקים, תוצאות)
- פונקציות אגרגציה, JOIN, GROUP BY - **שאלות מתקדמות**
- **PostgreSQL** פונקציות - תרגול החומר הנלמד בקורס

#### APIs עבודה עם C: מטרה

- rate limiting ומגבלות REST APIs הבנת **Chess.com/Lichess API** חיבור ל
- **JSON** עיבוד - המרה לפורמט מתאים למסד נתונים
- **ניהול נתונים** - איסוף הדרגתי וניקוי

### הקשר לפרויקט הגמר

(NBA מערכת המלצות טקטיות למאמני) AIAC פרויקט זה משמש כאב טיפוס מוקטן לפרויקט הגמר

- **גישה דומה:** חיזוי תוצאות על בסיס נתונים היסטוריים
- **כלים זהים:** Python, SQL, API integration, ML algorithms
- **עקרונות דומים:** feature engineering, model evaluation, data pipeline
- **מורכבות מותאמת:** פרויקט פשוט יותר לצורך למידה ותרגול

## הבעיה לפתרון

חיזוי תוצאות משחקי שחמט (ניצחון/לבן/שחור/תיקו) על בסיס:

- ELO rating) דירוגי השחקנים
- היסטוריית מפגשים
- סטטיסטיקות אישיות

## מקור הנתונים

- **Chess.com API** או **Lichess API** (חינמיים ונגישים)
- **היקף התחלת:** 500-1,000 משחקים (בגודל מתאים ללמידה)
- **איכות נתונים:** משחקים של שחקנים מדורגים (rating 1200+)

## הארכיטקטורה הטכנית

Chess API → Python Data Processing → PostgreSQL Database → ML Pipeline → Predictions

## השלבים המתוכננים

### שלב 1: הקמת תשתית (שבוע 1)

- והכנת סביבה PostgreSQL התקנת
- Chess API-בדיקת חיבור ל
- יצירת מבנה מסד נתונים בסיסי

### שלב 2: איסוף וניקוי נתונים (שבוע 2)

- API לחילוץ נתונים מ scripts כתיבת
- הזנה למסד נתונים
- ניקוי וולידציה של הנתונים

### (שבוע 3) **SQL** שלב 3: עבודה עם

- features כתיבת שאילתות לחילוץ
- views I-aggregations יצירת
- מהקורס PostgreSQL תרגול פונקציות

### שלב 4: **Machine Learning** (שבוע 4)

- לאימון datasets הכנת

- ואלגוריתמים נוספים Random Forest יישום
  - השוואת ביצועים והערכת מודלים
- 

## מדדי הצלחה

### מדדים טכניים

- (מתקבל על הדעת לשחמט baseline) **דיוק חיזוי**: מעל 60%
- **איכות קוד**: מבנה נקי ומתועד
- שאלות מהירות ומתקדמות: **SQL יעילות**

### מדדי למידה

- ותוצאות hyperparameters יכולת להסביר: **Random Forest הבנת**
  - כתיבת שאלות מורכבות בביטחון: **SQL-שליטה ב**
  - הבנת עקרונות וטיפול בשגיאות: **API עבודה עם**
- 

## מגבלות והיזהרויות

### גודל והיקף

- **לא להיסחף**: הפרויקט צריך להישאר פשוט ומתאים לזמן הזמין
- **משחקים בלבד**: מספיק ללמידה, לא יותר מדי לביצוע 500-1000
- deep learning נוספים, לא 1-2 + Random Forest: **אלגוריתמים בסיסיים**

### דגשים חשובים

- **הוא מרכזי**: לא רק לאחסון, אלא גם לתרגול הקורס **SQL**
  - **הוא מרכזי**: חובה להבין איך האלגוריתמים עובדים **ML**
  - **תיעוד ולמידה**: מטרה ללמוד, לא רק לסיים
- 

## התוצרים הצפויים

### מוצר טכני

- עובד ומובנה PostgreSQL מסד נתונים
- עד חיזוי API מלא מ Pipeline
- מאומן עם הערכת ביצועים ML מודל

### מוצר אקדמי

- תיעוד מפורט של התהליך

- ניתוח השוואתי של אלגוריתמים
- מתקדמות SQL דוגמאות שאילתות

## ערך מוסף

- ביטחון טכני לפרויקט הגמר
- SQL כלים עובדים לתרגול
- ML pipeline הבנה מעמיקה של



## מסגרת זמן

### סך הכל: 3-4 שבועות

- זמן סביר שלא יפריע ללימודים
- מספיק ללמידה מעמיקה
- יספק תשתית טובה לפרויקט הגמר

מטרה ללמוד ולהתקדם, לא ליצור מוצר מושלם - "Better done than perfect": **עקרון מנחה**

---



## סיכום

פרויקט זה הוא **גשר מעשי** בין הלמידה התיאורטית בקורסים לבין הפרויקט הגדול הקרוב. הוא מאפשר תרגול מבוקר של כלים חיוניים תוך שמירה על היקף סביר ומטרות ברורות.

המטרה היא לבנות **בסיס טכני חזק וביטחון עצמי** שיאפשרו להתמודד בהצלחה עם פרויקט הגמר המאתגר יותר.