

הנחיות לפרויקט סיום - חיזוי מחירי מניות באמצעות רשתות נוירונים חוזרות (LSTM ו-GRU)

מבוא:

פרויקט זה מהווה הזדמנות ליישם את הידע והכלים שלמדתם בקורס Data Science על בעיית חיזוי מעשית ופופולרית בתחום סדרות עתיות (Time Series) - חיזוי מחירי מניות. תתמקדו בשימוש בשני ארכיטקטורות חזקות של רשתות נוירונים חוזרות (RNNs): LSTM (Long Short-Term Memory) ו-GRU (Gated Recurrent Unit). מטרת הפרויקט היא לעבור את כל השלבים של פרויקט Data Science טיפוסי: איסוף וניתוח נתונים, עיבוד מקדים, בניית מודלים, אימון, הערכה והשוואה, וניתוח התוצאות. וכן לבחון איזה פיצ'רים חיונים למודל ומה התרומה של כול פיצ'רים .

מטרות הפרויקט:

1. לאסוף נתוני מניות היסטוריים ממקור חיצוני.
2. לאסוף עוד נתונים מחוץ למניה סצפצית , כגון ריביות , דווח תעסוקה , אבטלה , ועוד..
3. לבצע ניתוח נתונים אקספלורטורי (EDA) וויזואליזציה של הנתונים.
4. להכין את הנתונים לשימוש במודלי סדרות עתיות (חלונות זמן, סקאלינג).
5. Xgboost , LightBoost.
6. לבנות, לאמן ולהעריך מודל חיזוי מבוסס LSTM.
7. לבנות, לאמן ולהעריך מודל חיזוי מבוסס GRU.
8. להשוות את ביצועי שני המודלים באמצעות מדדי הערכה כמותיים (כמו MSE, RMSE).
9. לבחון האם הוספת מאפיינים נוספים (מעבר למחיר הסגירה) מסייעת לשיפור ביצועי המודל.
10. לתעד ולדווח על התהליך, הממצאים והמסקנות.
11. ולהשבות ביצועי MSE R Squer MSA בין המודלים .

שלבי הפרויקט:

שלב 1: איסוף וטעינת נתונים

* מקור נתונים: בחרו מניה ספציפית (לדוגמה, מניה מ-NASDAQ, NYSE, או כל בורסה אחרת הזמינה דרך API) וטווח תאריכים מספק (מומלץ מספר שנים, לפחות 3-5, תלוי בתדירות הנתונים שבחרתם - יומית היא הנפוצה ביותר).
* כלי: השתמשו בספריות Python כמו yfinance או pandas\_datareader כדי להוריד נתוני מניות היסטוריים.
* נתונים: ודאו שאתם מורידים את הנתונים הסטנדרטיים הכוללים: Date, Open, High, Low, Close, Adjusted Close, Volume.
* טעינה: טענו את הנתונים ל-DataFrame של pandas.

שלב 2: ניתוח נתונים אקספלורטורי (EDA) וויזואליזציה

* סקירה ראשונית: בדקו את מבנה הנתונים (.info(), .head(), .describe()), בדקו קיום ערכים חסרים.
* ויזואליזציה:
  + ציירו גרף קו (Line Plot) של מחיר הסגירה (Close) לאורך זמן. האם יש מגמה (טרנד)? האם יש עונתיות? תנודתיות?
  + ציירו גרף של נפח המסחר (Volume) לאורך זמן. האם יש קשר בין נפח המסחר לתנודות מחיר משמעותיות?
  + הציגו גרפים של Open, High, Low, Close באותו ציר זמן כדי לראות את התנועה היומית (אפשר להשתמש ב- Candlestick Chart אם רוצים להרחיב).
  + אם הוספתם פיצ'רים נוספים (ראו שלב 4), בצעו להם ויזואליזציה ובדקו קשרים בינם לבין מחיר הסגירה (למשל, scatter plots, correlation matrix).
* מסקנות מ-EDA: סכמו את הממצאים העיקריים מהניתוח והוויזואליזציה. האם יש תובנות לגבי הנתונים שיכולות להשפיע על מודל החיזוי?

שלב 3: עיבוד מקדים של הנתונים

* טיפול בערכים חסרים (אם קיימים): בחרו אסטרטגיה (השלמה, מחיקה) ונמקו אותה. לרוב, בנתוני מניות היסטוריים מדויקים אין ערכים חסרים רבים.
* בחירת פיצ'רים ומטרה: החליטו אילו עמודות ישמשו כפיצ'רים (X) ואילו כמטרת החיזוי (y). ברוב המקרים, המטרה תהיה לחזות את מחיר הסגירה ביום הבא. הפיצ'רים הבסיסיים יהיו מחיר הסגירה ב-N הימים האחרונים (כאשר N הוא גודל חלון הזמן). ניתן להרחיב לשלב 4.
* סקאלינג (Scaling): שלב קריטי לרשתות נוירונים. יש להשתמש בסקיילר (לדוגמה, MinMaxScaler או StandardScaler מ-sklearn.preprocessing) כדי לנרמל או למדר את הנתונים. חשוב מאוד לבצע סקאלינג *לאחר* פיצול הנתונים (ראו בהמשך), ולהתאים את הסקיילר *רק* על נתוני האימון, ואז להשתמש בו כדי לבצע טרנספורמציה גם על נתוני הוולידציה (אם יש) והטסט. יש לבצע סקאלינג הן על הפיצ'רים והן על עמודת המטרה.
* פיצול נתונים (Train/Test Split): מכיוון שמדובר בסדרת עתית, הפיצול חייב להיות כרונולוגי. חלקו את הנתונים לחלק אימון (Train) וחלק בדיקה (Test) על פי תאריך. מומלץ להקצות כ-70-80% מהנתונים לאימון והשאר לבדיקה. ניתן גם להוסיף חלק ולידציה (Validation) מתוך נתוני האימון לצורך כוונון היפרפרמטרים (כ-10-15% מנתוני האימון המקוריים).
* יצירת חלונות זמן (Time Windows / Sequences): רשתות RNN (ובפרט LSTM ו-GRU) מקבלות כקלט סדרות של נתונים. עליכם להמיר את הנתונים המקוריים למבנה של סדרות קלט (X) ונקודות מטרה מתאימות (y).
  + הגדירו גודל חלון זמן (N) – כמה ימים אחורה (לדוגמה, 60 ימים) ישמשו כדי לחזות את היום הבא.
  + צרו זוגות (Xt​,yt​) כאשר Xt​ היא סדרת הפיצ'רים מהזמן t−N+1 עד t, ו-yt​ היא המטרה בזמן t+1.
  + צורת הנתונים לאחר יצירת החלונות צריכה להיות: (מספר דוגמאות, גודל חלון, מספר פיצ'רים). לדוגמה, אם השתמשתם רק במחיר סגירה, גודל החלון הוא 60, ומספר דוגמאות האימון הוא 1000, אז צורת נתוני האימון תהיה (1000, 60, 1).

שלב 4 (אופציונלי / מתקדם): הנדסת מאפיינים (Feature Engineering)

* כדי לבחון האם פיצ'רים נוספים מסייעים, צרו מאפיינים חדשים מהנתונים הקיימים *לפני* הסקאלינג ויצירת החלונות:
  + ממוצעים נעים: Simple Moving Average (SMA), Exponential Moving Average (EMA) על מחיר הסגירה או הנפח בטווחים שונים (לדוגמה, 10 ימים, 50 ימים, 200 ימים).
  + אינדיקטורים טכניים: (מתקדם יותר) RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence), Bollinger Bands, וכו'.
  + מאפייני זמן: יום בשבוע, חודש, שנה (ייתכן שפחות רלוונטי בחיזוי יום-אחרי-יום אך יכול להוסיף הקשר).
  + תנודות מחיר: שינוי יומי במחיר ((Close - Open) / Open), שינוי יומי בנפח.
* בצעו את הניתוח והוויזואליזציה (שלב 2) מחדש עם הפיצ'רים החדשים.
* השוו מודלים שאומנו עם הפיצ'רים הבסיסיים בלבד מול מודלים שאומנו עם הפיצ'רים המורחבים (ראו שלב 8).

שלב 5: בניית מודלים (LSTM ו-GRU)

* כלי: השתמשו בספריית למידה עמוקה כמו TensorFlow/Keras או PyTorch. Keras בדרך כלל פשוטה יותר לבניית מודלים סדרתיים.
* ארכיטקטורה:
  + מודל LSTM: בנו מודל סדרתי (Sequential). שכבת הקלט צריכה להתאים לצורת הנתונים (גודל חלון, מספר פיצ'רים). הוסיפו שכבת LSTM אחת או יותר (LSTM). שימו לב לפרמטר return\_sequences=True אם אתם מחברים מספר שכבות LSTM ברצף. בסוף, הוסיפו שכבת Dense אחת עם יחידה אחת (כדי לחזות ערך יחיד - מחיר הסגירה ביום הבא). השתמשו בפונקציית הפעלה מתאימה בשכבת הפלט (לרוב Linear או None עבור בעיות רגרסיה).
  + מודל GRU: בנו מודל מקביל למודל ה-LSTM, אך השתמשו בשכבת GRU (GRU) במקום LSTM.
* היפרפרמטרים: נסו ערכים שונים עבור מספר היחידות בשכבות ה-RNN, מספר השכבות, שימוש בשכבות Dropout (עוזר למנוע Overfitting), גודל ה-Batch, מספר Epochs. התחילו עם ארכיטקטורה פשוטה (לדוגמה, שכבת LSTM/GRU אחת עם 50 יחידות) והתקדמו משם.
* קומפילציה: קמפלו את המודלים.
  + Loss Function: Mean Squared Error (MSE) הוא הבחירה הסטנדרטית לבעיות רגרסיה כמו זו.
  + Optimizer: Adam הוא אופטימייזר נפוץ ויעיל.
  + Metrics: הוסיפו את MSE ו-RMSE (Root Mean Squared Error) כמדדים למעקב בזמן האימון וההערכה.

שלב 6: אימון המודלים

* אימון: אמנו כל מודל בנפרד באמצעות הפונקציה model.fit().
  + העבירו את נתוני האימון (X\_train, y\_train).
  + הגדירו epochs ו-batch\_size.
  + אם יש לכם נתוני ולידציה, העבירו אותם לפרמטר validation\_data.
  + מומלץ להשתמש ב-Callbacks כמו EarlyStopping כדי לעצור את האימון אם ביצועי הוולידציה מפסיקים להשתפר, ו-ModelCheckpoint כדי לשמור את המודל הטוב ביותר על בסיס הוולידציה.
* מעקב: עקבו אחרי ערך ה-Loss ומדדי ההערכה (MSE, RMSE) הן על נתוני האימון והן על נתוני הוולידציה (אם קיימים) לאורך ה-Epochs. ציירו גרפים של ערכי ה-Loss והמדדים כדי לראות את תהליך הלמידה והתכנסות.

שלב 7: הערכת ביצועי המודלים

* חיזוי על נתוני הבדיקה: השתמשו במודלים שאומנו כדי לבצע חיזויים על נתוני הבדיקה (X\_test). הפונקציה היא model.predict(). שימו לב שהפלט הוא בפורמט ה-Scaled.
* הסרת סקאלינג (Inverse Transform): מכיוון שאימנתם על נתונים שעברו סקאלינג, התחזיות של המודל הן גם בטווח המסוקל. כדי להשוות אותן למחירי המניה המקוריים ולחשב מדדי הערכה בטווח המקורי, יש לבצע Inverse Transform על התחזיות באמצעות אותו סקיילר ששימש לסקאלינג עמודת המטרה.
* חישוב מדדים: חשבו את MSE ו-RMSE (ובאופן אופציונלי MAE - Mean Absolute Error) על התחזיות שחזרו לסקאלה המקורית בהשוואה למחירי המניה האמיתיים בסט הבדיקה (שגם עברו Inverse Transform אם עמודת המטרה המקורית סוקלה).
* ויזואליזציה של החיזויים: ציירו גרף קו המציג את מחירי המניה האמיתיים בסט הבדיקה ואת מחירי החיזוי של כל אחד מהמודלים (LSTM ו-GRU) על אותו גרף. גרף זה קריטי להבנה ויזואלית של עד כמה החיזוי קרוב למציאות.

שלב 8: השוואת מודלים וניתוח תוצאות

* השוואה כמותית: הציגו טבלה מסכמת של מדדי ההערכה (MSE, RMSE) עבור מודל ה-LSTM ומודל ה-GRU על גבי סט הבדיקה. איזה מודל השיג ביצועים טובים יותר (ערכי מדדים נמוכים יותר)?
* השוואה איכותית: התייחסו לגרף החיזויים מול הנתונים האמיתיים (משלב 7). איזה מודל נראה ויזואלית קרוב יותר לנתונים האמיתיים? האם יש תקופות בהן מודל אחד טוב יותר מהשני?
* השפעת פיצ'רים (אם רלוונטי - שלב 4): אם הוספתם פיצ'רים נוספים, השוו את מדדי ההערכה של המודלים (LSTM ו-GRU) שאומנו עם הפיצ'רים המורחבים אל מול המודלים שאומנו עם הפיצ'רים הבסיסיים בלבד. האם הוספת הפיצ'רים שיפרה את הביצועים? באילו פיצ'רים נראה שהייתה ההשפעה הגדולה ביותר (ייתכן שניתוח זה דורש ניסויים נוספים, אך לפחות התייחסות ראשונית)?
* דיון:
  + מדוע מודל מסוים הצליח טוב יותר מהשני (אם בכלל)? התייחסו להבדלים התיאורטיים בין LSTM ו-GRU. GRU לרוב פשוט יותר, מהיר יותר לאימון, ועם פחות פרמטרים, אך LSTM יכול ללכוד תלויות ארוכות טווח בצורה אפקטיבית יותר במקרים מסוימים.
  + האם הוספת פיצ'רים אכן הייתה מועילה? למה כן/לא?
  + מהן המגבלות של המודלים שבניתם בהקשר של חיזוי מניות? (לדוגמה: אינם מתחשבים באירועים חדשותיים, שינויים מקרו-כלכליים, שינויים בחוק ורגולציה וכו'. הם מתבססים אך ורק על נתונים היסטוריים).
  + כיצד ניתן לשפר את המודלים בעתיד? (לדוגמה: הוספת פיצ'רים נוספים, שימוש בארכיטקטורות מודלים מורכבות יותר, כוונון היפרפרמטרים מעמיק יותר, חיזוי לטווח ארוך יותר, שימוש בנתונים לא מספריים כמו חדשות).

שלב 9: סיכום ומסקנות

* סכמו את ממצאי הפרויקט, כולל השוואת המודלים וההשפעה של הפיצ'רים.
* ציינו את המסקנות העיקריות לגבי היכולת לחזות מחירי מניות באמצעות LSTM ו-GRU על הנתונים שבחרתם.
* ציינו את האתגרים בהם נתקלתם וכיצד התמודדתם איתם.

תוצרים נדרשים:

1. קוד Python: קוד מלא ומסודר (במחברת Jupyter Notebook או קבצי .py) המכיל את כל שלבי הפרויקט: איסוף נתונים, EDA, Preprocessing, בניית מודלים, אימון, הערכה והשוואה. הקפידו על קוד קריא, מתועד (עם הערות) ומאורגן בפונקציות או תאים נפרדים לכל שלב.
2. דו"ח פרויקט מסכם: מסמך (PDF) המפרט את כל שלבי הפרויקט באופן מובנה. הדו"ח צריך לכלול:
   * מבוא (מטרות הפרויקט, המניה וטווח התאריכים שנבחרו).
   * תיאור הנתונים וממצאי ה-EDA (כולל ויזואליזציות רלוונטיות).
   * פירוט תהליך עיבוד הנתונים המקדים (פיצול, סקאלינג, יצירת חלונות, פיצ'רים שנבחרו/נוספו).
   * תיאור ארכיטקטורת המודלים (LSTM ו-GRU) והיפרפרמטרים שנבחרו.
   * תיאור תהליך האימון (epochs, batch size, loss function, optimizer).
   * תוצאות הערכת המודלים (טבלה של מדדים, גרף השוואת חיזויים מול נתונים אמיתיים).
   * ניתוח והשוואה של המודלים ומסקנות לגבי השפעת הפיצ'רים.
   * מסקנות סופיות והצעות לעבודה עתידית.
   * רפרנסים (אם נעשה שימוש במאמרים, קוד חיצוני וכו').