

פרויקט גמר 5 יחידות לימוד

Deep Learning

שם העבודה: Forecasting apartment prices

שם התלמיד: מאור דרורי

שם בית הספר: מקיף י"א ראשונים

שם המנחה: דינה קראוס

שם החלופה: למידת מכונה

תאריך הגשה: 21.6.2024



תוכן עניינים

3.....	מבוא
5.....	מבנה/ארכיטקטורה
23.....	מדריך למפתח
27.....	מדריך למשתמש
30.....	רפלקציה / סיכום אישי
31.....	ביבליוגרפיה
32.....	נספחים

מבוא

רוכשי ומוכרי הדירות רבים מסתמכים על חיזויים בשביל להעריך מחיר של דירה מסוימת בעתיד.

דוגמא אחת לכך היא כאשר משקיע(קונה הדירה) מנסה לדעת ולהעריך האם שווה לו לקנות את הדירה עכשיו או בעתיד. יכול להיות שהמשקיע אראה לפי מחירי דירות קודמות שמחירי הדירות ממשיכות לעלות אזי הוא יבין ששווה לו לקנות את הדירה עכשיו מאשר בעתיד.

השנה אנחנו, תלמידי מגמת הנדסת תוכנה בבית ספר ראשונים עשינו פרויקט בתחום שנקרא Machine learning או בעברית – למידת מכונה.

למידת מכונה היא תת תחום של ענף מחקר שנקרא בינה מלאכותית החותר להקנות למחשב יכולת חשיבה ויכולת ביטוי הקרובות לדרכי החשיבה והביטוי האנושית בעזרת פיתוח אלגוריתמים המיועדים לאפשר למחשב ללמוד מתוך ניסיונות ונתונים. במילים אחרות המחשב מקבל תחזיות או החלטות מבלי להיות מתוכנת מפורשות לעשות זאת.

למידה עמוקה היא תת תחום של למידת מכונה העוסקת באלגוריתמים הנקראים רשתות נוירונים, וזאת בהשראת המבנה והתפקוד של המוח הבנוי מתאי עצב(נוירונים) המקושרים זה לזה.

בפרויקט שלי בהתמחות זו בחרתי לבנות מודל שמטרתו חיזוי מחירי דירות על ידי הענקת נכסי דירות רבים שנמכרו בעבר ומהם הוא מנתח את המידע ומנסה לחזות מחירי דירות בעתיד.

להלן הסבר קצר שלב אחר שלב כיצד זה עובד:

1. נתוני עיבוד מוקדם: איסוף הדירות לקובץ Excel מסוג csv וסידורם לטורים על פי התכונות המשפיעות ביותר על מחיר הדירה, וכמו כן, עמודה של מחיר הדירה.
 2. אימון המודל(train) + וולידציה(validation): לאחר מכן המודל מאומן על ידי קריאת הנתונים מקובץ ה Excel. הוא לומד ליצור קשרים בין תכונות הדירה לבין מחיר הדירה. תהליך זה חוזר על עצמו פעמים רבות והמודל משתפר בהבנת הקשר בין תכונות הדירה לבין מחיר הדירה עם כל איטרציה. לאחר שהמודל הפנים את הקשר, אנו בודקים אותו על דירות שהוא לא ראה בשביל לראות שהמודל אכן למד.
 3. בחינה(test): לאחר האימון והוולידציה, אנו בוחנים את המודל על דירות שהוא כמו בוולידציה, לא ראה בעבר, אך הפעם, המטרה של הטסט לעומת האימון והוולידציה שאנו בוחנים את המודל ולא לומדים מהתוצאות, כלומר, אנו לא מסיקים מסקנות איך אפשר לשפר את המודל, אנו רוצים לראות מה מצבו על דירות שלא ראינו כלל.
- בחרתי בפרויקט זה מכיוון שאבי הוא מטווח דירות שעוסק בתחום הזה בחיי היום יום שלו. תחום זה עניין וסקרן אותי והחלטתי לקחת את ההזדמנות ולעשות בתחום הזה פרויקט. כמו כן, יש בו שימוש לחלק גדול מאוכלוסיית העולם. הפרויקט שלי יכול לעזור לרוכשי ומוכרי דירות רבים להעריך מחיר דירה עתידית בשביל צרכיהם.
- במהלך עשיית הפרויקט נתקלתי בלא מעט אתגרים. ראשית הייתי צריך ללמוד על אופן מחירי הדירות ומגמת המחירים לאורך השנים. בנוסף, הייתי צריך ללמוד את תחום למידת המכונה, שלא הכרתי מספיק טוב לפני בשביל להכין פרויקט שכזה. בנוסף לכך, היה לי קשה לבחור נושא לפרויקט. ידעתי שאני רוצה לבחור נושא לפרויקט הקשור לתחום הדירות אך לא ידעתי איזה נושא במדויק אני צריך לבחור בתחום הזה. אך ככל שנכנסתי יותר לעומק והידע שלי גדל גם בתחום מחירי הדירות וגם בתחום למידת המכונה, ידעתי איזה נושא במדויק לבחור בשביל שאעניין אותי ויהווה אתגר במהלך הפרויקט.

מושגים בלמידת מכונה

Loss – הפסד מייצג ערך מספרי המכמת את הפער בין התפוקות החזויות של מודל לבין התפוקות האמיתיות או הצפויות. הוא מודד את השגיאה או העלות של תחזיות המודל ומשמש כמדריך לעדכון הפרמטרים של המודל במהלך תהליך האימון, במטרה למזער את ההפסד ולשפר את ביצועי המודל.

Dataset – מערך נתונים מתייחס לאוסף של נקודות נתונים או דוגמאות המשמשות לאימון, הערכה ובדיקה של מודלים של למידת מכונה. מערכי נתונים חיוניים לאימון מודלים, ומספקים את המידע הדרוש למודל כדי ללמוד דפוסים ולבצע תחזיות.

Train – אימון הוא תהליך של אופטימיזציה איטרטיבית של מודל למידת מכונה באמצעות מערך נתונים מסומן. במהלך האימון, המודל לומד מנתוני הקלט, מתאים את הפרמטרים שלו וממזער את פונקציית ההפסד על ידי עדכון איטרטיבי של המשקלים וההטיות של המודל. המטרה היא לאפשר למודל לבצע תחזיות מדויקות על נתונים בלתי נראים.

Validation – אימות הוא תהליך הערכת הביצועים וההכללה של המודל על מערך נתונים נפרד, הנקרא לעיתים קרובות מערך האימות. הוא משמש במהלך שלב האימון כדי לכוון הפרמטרים היפר, לבחור את הדגם הטוב ביותר, או למנוע התאמת יתר. ערכת האימות מסייעת במעקב אחר ביצועי המודל על נתונים שונים ממערכי ההדרכה והבדיקות.

Test – בדיקה כוללת הערכת ביצועי המודל המאומן על מערך נתונים נפרד שלא נעשה בו שימוש בשלב ההדרכה. מערך הנתונים של הבדיקה מכיל דוגמאות ידועות, המאפשר להשוות את התחזיות של המודל אל מול האמת הבסיסית. הוא מספק אומדן של ההכללה והביצועים של המודל על נתונים בלתי נראים.

Epoch – מספר הפעמים שהמכונה מבצעת מחדש את תהליך הלמידה.

Overfitting - מצב בו המודל מותאם יותר מדי ל-dataset אותו הוא לומד, ולכן הוא פחות יצליח בחיזוי מחירי הדירות (או במקרים אחרים, תמונות). הדבר קורה כאשר המודל נקבע על ידי יותר פרמטרים מאשר הנתונים מצדיקים.

Underfitting - מצב בו המודל לא מותאם בכלל ל-dataset אותו הוא לומד, ולכן הוא פחות יצליח בחיזוי מחירי הדירות (או במקרים אחרים, תמונות). הדבר קורה כאשר המודל פשוט מידי כדי לייצג כראוי את מבנה הנתונים, לדוגמה, בעקבות מיעוט של פרמטרים המגדירים את המודל. דוגמא לכך היא למשל לנסות להשתמש במודל לינארי ולנסות לחזות מחיר של מנייה, שהרי הוא בעל התנהגות לא לינארית.

מושגים נוספים:

R^2 – מדד סטטיסטי שמראה את מידת ההתאמה בין הערכים החזויים של המודל לערכים האמיתיים. הוא נע בין 0 ל-1, כאשר שאיפה לאחד מצביע על כך שהמודל מסביר את השונות בנתונים.

RMSE – מדד להערכת שגיאת התחזיות של המודל. הוא מספק מידע על גודל השגיאות הצפויות של התחזיות, כאשר ערך נמוך מצביע על תחזיות מדויקות יותר.

בפרויקט שלי השתמשתי במודל **Deep learning regressor model** הוא סוג של רשת נוירונים אשר עוזר בשביל בעיות רגרסיה כאשר הוא לומד בעצמו דפוסים מורכבים מהנתונים ומטרתו לחזות ערכים מספריים (לדוגמה במקרה שלי חוזה מחירי דירות). כמו כן, השתמשתי במודל של linear regression אך ורק בשביל השוואה והסקת מסקנות בין שני המודלים.

מבנה\ארכיטקטורה של הפרויקט

שלב איסוף הכנה וניתוח נתונים:

לצורך בניית מבנה הנתונים (Dataset), לקחתי מידע על 4 שכונות בראשון לציון: פרס נובל, נאות אשלים, נאות שקמה, רמבם בין השנים 2009 – 2023, מידע על תכונות כל דירה ומחירי הדירה מאתר הנדל"ן הממשלתי – gov נדל"ן. להלן קישור לאתר: [/https://www.nadlan.gov.il](https://www.nadlan.gov.il)

בשביל שאוכל להשתמש במידע זה לצורכי הפרויקט שלי, המרתי את הדאטה שאני צריך לקובץ Excel מסוג csv וסידרתי אותו כך שאוכל להשתמש בו. הדאטה סט שלי מסודר לפי תאריכים – 2009 עד 2023 ובנוי מ – 9 עמודות אשר 8 העמודות הראשונות מייצגות את התכונות של כל דירה והעמודה ה 9 מייצגת את מחיר הדירה אותו אנו מנסים לחזות(שמות כל העמודות כתובות בשורה הראשונה). בנוסף, בדאטה סט שלי יש 6538 שורות אשר כל שורה מייצגת דירה בעלי מאפיינים שונים(לא כולל השורה הראשונה).

להלן תכונות הדירה בדאטה סט והסבר קצר עליהם:

Month – החודש שבו נמכר הדירה

Year – השנה שבה נמכרה הדירה

Neighborhood – השכונה שבו נמכר הדירה

Street – הרחוב שבו נמכר הדירה

NumberOfRooms – כמות החדרים בדירה שנמכרה

Floor – מספר הקומה של הדירה שנמכרה

Size – מטר רבוע(מ"ר) של הדירה שנמכרה (גודל הדירה)

הערה: לא החשבתי את העמודה Date כתכונה כי עמודה זו שימשה רק בשביל יצירת שני עמודות חדשות שלא הופיעו באתר בו לקחתי את הדאטה סט (Month, Year).

הערך אותו אני רוצה לחזות בדאטה סט:

Apartment_Price – מייצג את מחיר הדירה שנמכרה

חילקתי את הדאטה סט שלי לפי השנים שבהם נמכרו דירות על 4 השכונות שבחרתי באופן הבא:
Train: 2009 - 2019

Validation: 2020 - 2021

Test: 2022 - 2023

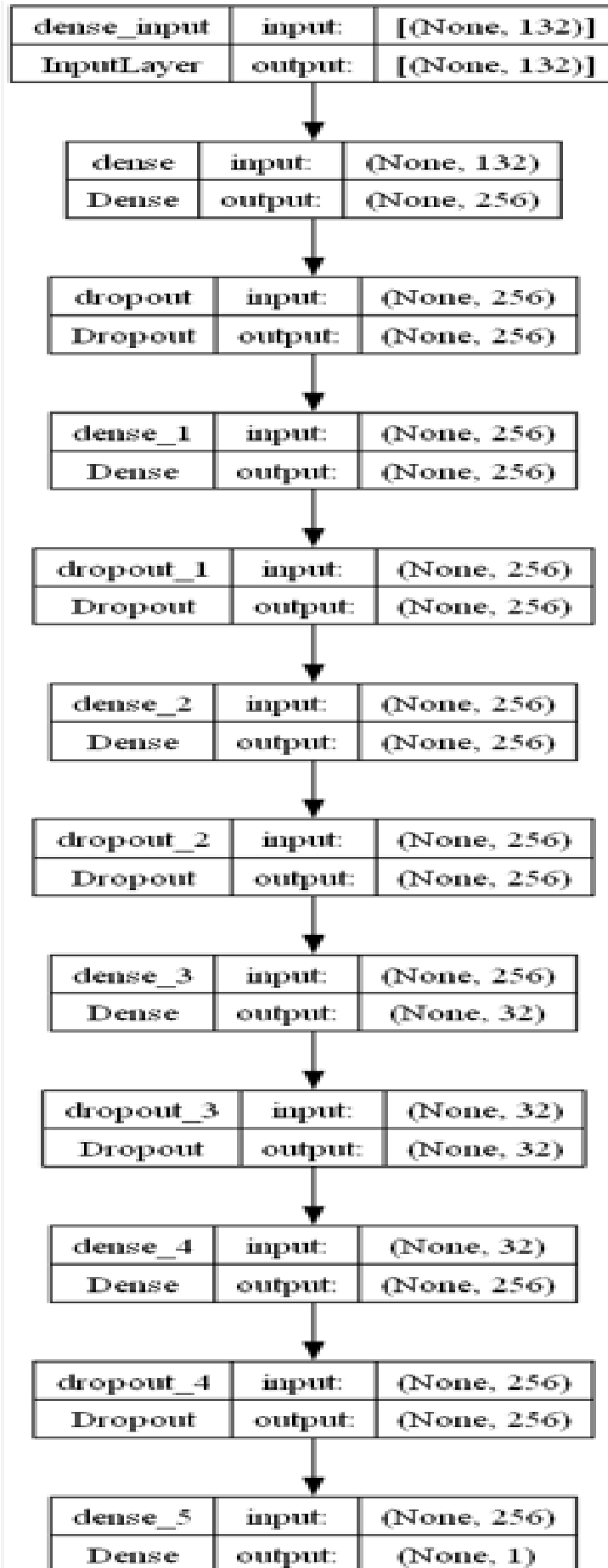
תחילה פיצלתי את הדאטה סט שלי כך בשביל שאוכל למצוא את הארכיטקטורה הטובה ביותר למודל שלי. בשביל שאוכל למצוא את הארכיטקטורה הטובה ביותר למודל שלי אני הרצתי סוגי ארכיטקטורות שונות על ה Train ו Validation ללא החלק של ה Test. החלק של הטסט (Test) שמרתי אותו ולא השתמשתי בו רק אחרי שמצאתי את הארכיטקטורה למודל הכי טוב שלי כי הטסט אמור להיות נתונים אשר המכונה לא ראתה מעולם. לאחר שמצאתי את הארכיטקטורה הטובה ביותר למודל שלי אני מחלק שוב את הדאטה סט שלי ל Train (2009 – 2021) ו test (2022 – 2023).

בפרויקט שלי, נרמלתי את הנתונים כדי לעבד מראש את התכונות לפני שאני מזין אותם למודל שלי. בשביל לנרמל את הנתונים השתמשתי ב – StandardScaler מהספרייה sklearn.preprocessing.

שלב בנייה ואימון המודל:

כמו שכבר ציינתי, אני רציתי קודם למצוא את המודל הכי טוב, כלומר, המודל שמביא לי את התוצאה של ה rmse הכי נמוך. לשם כך, הפרדתי ל train, validation, test, "התעלמתי" מהדאטה של ה test והרצתי סוגי ארכיטקטורות שונות על הטריין והוולידציה. לא החשבתי את הטסט כי הוא אמור להיות נתונים שהמודל לא ראה. הוולידציה הפכה להיות כטסט ובכך הרצתי סוגי ארכיטקטורות שונות עד שמצאתי את המודל בעל התוצאה של rmse של הוולידציה הכי נמוך. המודל בעל התוצאה של rmse של הוולידציה הכי נמוך הכי המודל שהפיק לי את התוצאות הכי טובות ואני אוכל להשתמש איתו ולבדוק אותו על הטסט כאשר עכשיו חילקתי ל train ו test כאשר הטריין מכיל את הנתונים של הטריין והוולידציה הקודמים והטסט זה נתונים (דירות) שהמודל נבחן עליהם בפעם הראשונה.

תיאור גרפי של המודל שעליו בוצע האימון:



הסבר על סוגי השכבות השונים ברשת:

שכבת Dense

שכבה זו היא שכבה ברשת נוירונים שכל נוירון בשכבה הזו מקבל קלט מכל הנוירונים מהשכבה הקודמת.

להלן הסבר על כל פרמטר בשכבת Dense:

החלק הראשון הוא החלק שבו קובעים את כמות הנוירונים בשכבה

activation – בחלק הזה קובעים איזה פונקציית אקטיבציה להשתמש בשכבה

input shape – מגדירים פה גודל הממדים של הקלט שהשכבה צריכה. אין צורך להגדיר שוב גם בשאר השכבות חוץ מהשכבה הראשונה.

kernel_regularizer – שיטה המיושמת על המטריצה של המשקלים. כדי למנוע overfitting הוספתי penalty לפונקציית ה loss.

שכבת Dropout

מטרתה של שכבה זו היא מניעת overfitting. השכבה מגדירה באופן אקראי חלק מהנוירונים של השכבה הקודמת להיות 0 בכל עדכון בזמן האימון, מה שעוזר למודל ליישם מה שהוא למד בשלב האימון לדוגמאות שהוא עוד לא נתקל בהם.

להלן הסבר על הפרמטר בשכבת Dropout:

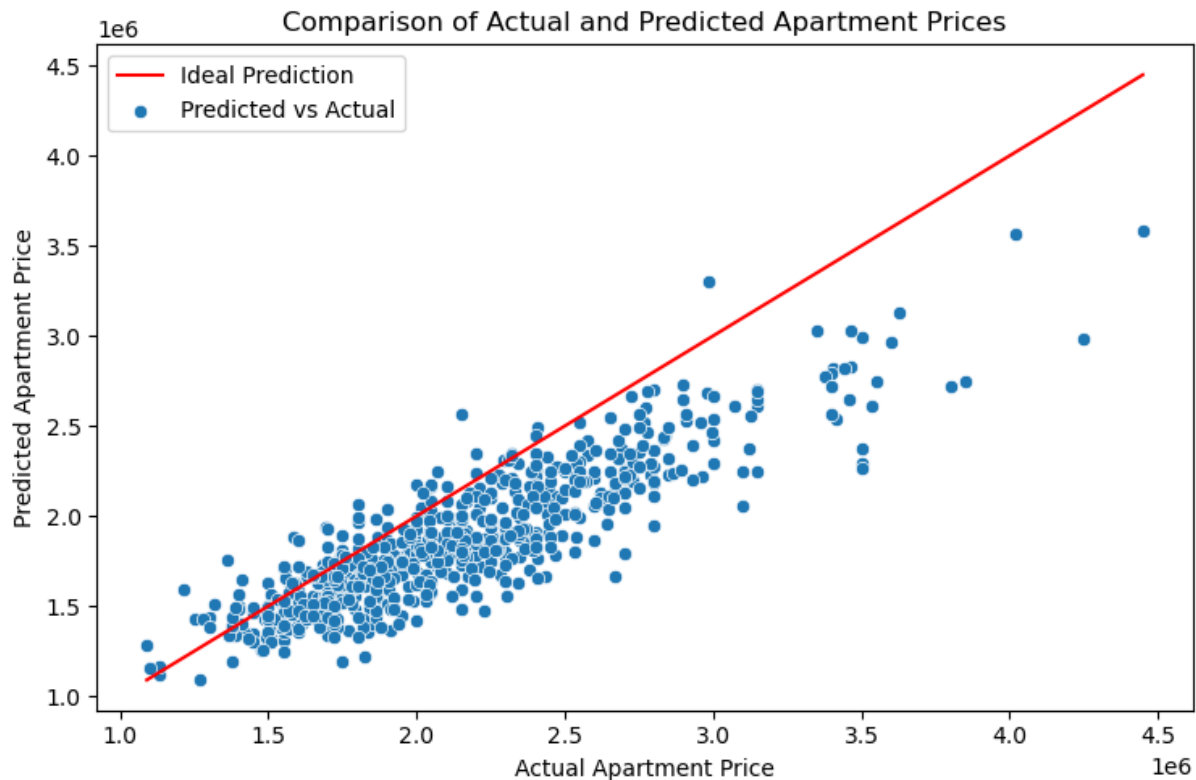
בפרמטר של השכבה הזו מגדירים כמה נוירונים אנו רוצים להיפטר מהם באופן רנדומלי, לדוגמא – 0.1 הכוונה 10% מהנוירונים בשכבה זו אנו נפטרים מהם באופן רנדומלי.

השכבה האחרונה היא שכבת Dense עם רק נוירון אחד שמייצג את שכבת הפלט

דוחות וגרפים המתארים את תוצאות שלב האימון

בכל אימון של מודל חדש, נוצר עבורנו מספר גרפים ודוחות, הגרפים והדוחות הללו נשמרים בתוך תיקיית ההרצה (D:\project_visual_studio_code). הגרפים מספקים מידע רלוונטי לגבי ביצועי המודל והשתפרותו במהלך האימון. בעזרת גרפים אלו, אנו יודעים איפה המודל צריך חיזוק על מנת לשפר את מבנה הנתונים ובעזרת כך לשפר את תוצאות האימון בהרצות הבאות.

כאשר אנו מריצים את המודל, אנו נקבל תוצאות טיפה שונות כל פעם, אך להלן דוגמא של הרצה של



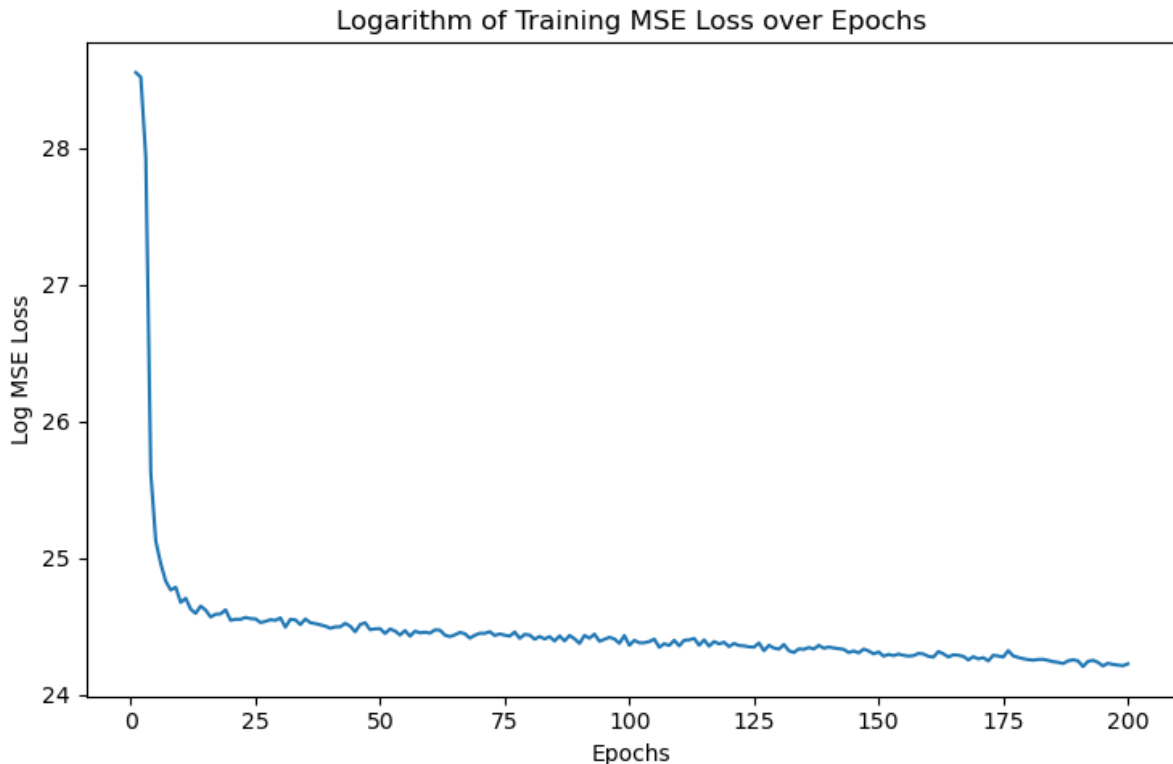
התוצאות של המודל הכי טוב שלי:

גרף זה מציג לנו את מחירי הדירות שהמודל ניסה לחזות כפונקציה של המחירים האמיתיים בשוק.

הקו המוצג לנו בגרף המתאר את החיזויים המושלמים, כלומר, נקודות הנמצאים בדיוק על הקו מבטאים דיוק של 100% של המודל לחיזוי מחיר הדירה.

כל נקודה בגרף מייצג מחיר אמיתי לעומת מחיר חזוי.

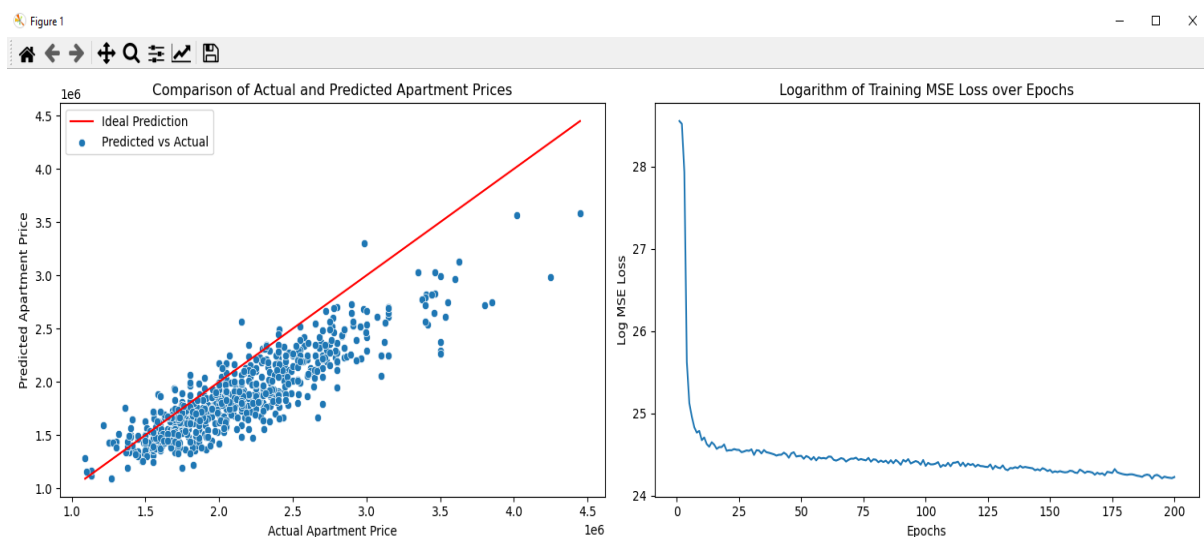
ככל שהנקודות קרובות יותר לקו, החיזוי של המודל מדויק יותר. נקודות רחוקות יותר מהקו מייצגות דיוק נמוך יותר של המודל לחיזוי מחירי הדירות.



גרף זה מייצג את הלוגריתם של האימון של ה MSE loss כפונקציה של כמות האיטרציות (Epochs), כלומר, ה loss של האימון מתייחס לערך ה loss (במקרה שלנו זה הלוגריתם של ה MSE loss) שמחושב על פי נתוני האימון בכל איטרציה במהלך תהליך האימון. פונקציית ה loss מודדת עד כמה התחזיות של המודל תואמות את המחירים האמיתיים ואימון של ה loss פוחת ככל שהמודל לומד מהנתונים של האימון.

ציר ה x מייצג את כמות האיטרציות וציר ה y מייצג את הלוגריתם של ה MSE loss, לא ה MSE loss עצמו (החלטתי לעשות את הלוגריתם של ה MSE loss בשביל לראות ויזואלית טוב יותר את ערכי ה loss).

להלן התמונה המקורית שמוצגת על המסך בסיום ההרצה (הפרדתי לשתי תמונות לצורך הסבר נוח



יותר וקל יותר להבנה:

```
Train: rmse: 152951.0, r2 score: 0.89
Test:  rmse: 360009.0, r2 score: 0.51
```

כאן ניתן לראות את התוצאות גם של האימון (train) וגם של המבחן (test).
התוצאות הנ"ל מבטאות את מדדי הביצועים של המודל עבור הדאטה סט של train ועבור הדאטה סט של test.

תוצאות ההרצה:

:Train

Root Mean Squared Error (RMSE): 152,951

R-squared (R^2): 0.89

:Test

Root Mean Squared Error (RMSE): 360,009

R-squared (R^2): 0.51

מדדים אלו מצביעים על כך שהמודל מתפקד היטב בנתוני האימון עם ערך R^2 גבוה (0.89) וערך RMSE נמוך יחסית, אך יש לו RMSE גבוה משמעותית ו R^2 נמוך יותר (0.51) ב test, דבר המצביע על overfitting (כנראה ההתאמת היתר הזאת נובעת מכמות נתונים קטנה יחסית בדאטה סט).

בסיום הרצת המודל, נוצר גם כן קובץ אקסל (result_model_df) ובו נתונים על כל דירה(בדיוק אותם נתונים שיש בקובץ הדאטה סט) ובנוסף ישנו עמודה נוספת של המחירים החזויים(predicted_price) של ה train וכמו כן של ה test.

predicted_price	Apartment_Price	Size	Floor	NumberOfRooms	Street	Neighborhood	Year	Month	Date
745860.2	714000	79	6	3	Herzl	Rambam	2009	1	01/01/2009
391877.16	470000	36	3	2	DavidAlroi	Rambam	2009	1	01/01/2009
892985.56	930000	65	2	3	Kapah	NeotShikma	2009	1	05/01/2009
894522.44	1000000	80	6	4	Tarmav	Rambam	2009	1	05/01/2009
1358820.5	1450000	104	9	4	HaNagid	NeotShikma	2009	1	07/01/2009
955278.3	1050000	72	8	3	HaRavNeriya	NeotShikma	2009	1	08/01/2009
1436646.4	1370000	131	4	6	DavidHaReuveni	Rambam	2009	1	08/01/2009
797249.9	710000	87	1	3.5	Abarbanel	Rambam	2009	1	08/01/2009
1279155.4	1390000	98	4	4	Alkabets	NeotShikma	2009	1	12/01/2009
911225.7	838000	65	3	3	HaRashba	NeotShikma	2009	1	14/01/2009

התמונה הנ"ל מציגה דוגמאות של כמה דירות וחיזוי המחיר על כל דירה בשלב ה train.

predicted_price	Apartment_Price	Size	Floor	NumberOfRooms	Street	Neighborhood	Year	Month	Date
1628902.6	1580000	72	3	3	YehudaLeib	Rambam	2023	12	05/12/2023
1827945.4	2400000	106	2	4	Givati	Rambam	2023	12	07/12/2023
1700425.1	1840000	43	4	2	HaTof	NeotAshalim	2023	12	08/12/2023
2036084.4	1900000	67	5	3	HaOrgan	NeotAshalim	2023	12	10/12/2023
2367008.8	2605000	142	4	5	ItamarBenAvi	Rambam	2023	12	13/12/2023
2395957.8	2760000	94	9	4	Alkabets	NeotShikma	2023	12	13/12/2023
2321875.8	2850000	95	4	4	HaKinor	NeotAshalim	2023	12	13/12/2023
1490254.5	1390000	46	4	2	HaAhimSuleiman	Rambam	2023	12	21/12/2023
2280551.2	2400000	117	6	4	Vitkin	Rambam	2023	12	21/12/2023
1385298.8	1300000	50	1	2	Rothschild	Rambam	2023	12	25/12/2023

התמונה הנ"ל מציגה דוגמאות של כמה דירות וחיזוי המחיר על כל דירה בשלב ה test.

החיזויים של המודל במהלך ה train:

בקובץ האקסל ניתן לראות זאת בשורות 2 – 5875.

החיזויים של המודל במהלך ה test:

בקובץ האקסל ניתן לראות זאת בשורות 5876 – 6538.

ratio predicted and real	predicted_price_filter_outliers	Apartment_Price	Size	Floor	NumberOfRooms	Street	Neighborhood	Year	Month	Date
1.0448111	745995.1253	714000	79	6	3	Herzl	Rambam	2009	1	01/01/2009
0.466270479	219147.1253	470000	36	3	2	DavidAlroi	Rambam	2009	1	01/01/2009
1.017336694	946123.1253	930000	65	2	3	Kapah	NeotShikma	2009	1	05/01/2009
0.801419125	801419.1253	1000000	80	6	4	Tarmav	Rambam	2009	1	05/01/2009
1.014782845	1471435.125	1450000	104	9	4	HaNagid	NeotShikma	2009	1	07/01/2009
1.073321072	1126987.125	1050000	72	8	3	HaRavNeriya	NeotShikma	2009	1	08/01/2009
0.997473814	1366539.125	1370000	131	4	6	DavidHaReuveni	Rambam	2009	1	08/01/2009
0.961457923	682635.1253	710000	87	1	3.5	Abarbanel	Rambam	2009	1	08/01/2009
0.959823831	1334155.125	1390000	98	4	4	Alkabets	NeotShikma	2009	1	12/01/2009
1.116958383	936011.1253	838000	65	3	3	HaRashba	NeotShikma	2009	1	14/01/2009

ניתן לראות בתמונה הנ"ל דוגמא של כמה שורות מקובץ אקסל אשר כל שורה מייצגת דירה למעט 2 הטורים האחרונים. בטור אחד לפני האחרון(predicted_proce_fillter_outliers) השתמשתי בספרייה sklearn בשביל להשתמש במודל מובנה בספרייה של Linear regression וכך יצרתי עמודה נוספת שבעזרת המודל היא חוזה כמה הדירה עלתה. בעמודה האחרונה ישנו היחס בין העמודה שהמודל חזה(predicted_proce_fillter_outliers) לבין העמודה המייצג את המחיר המקורי של הדירה(Apartment_Price) אשר כתוב באתר ממנו לקחתי את הדאטה סט. העמודה הזאת עוזרת לי לשם סינון הדירות עם מחירים לא הגיוניים. אני ואבא שלי אשר עובד בתחום, עברנו על הדירות אשר היחס גדול/קטן מאוד ובדקנו האם אנו צריכים לסנן אותם מהדאטה סט בשביל נתונים נכונים או לא.

```
{'base_model': {'train': {'rmse': 166758.74260427145, 'r2': 0.8722721668370298}, 'test': {'rmse': 325273.3305200072, 'r2': 0.6017989034114082}}}
```

להלן תוצאות מודל ה linear regression אשר השתמשתי אך ורק לשם השוואה למודל ה deep learning regressor model. בתמונה תוכלו לראות את תוצאות המודל אשר מאוחסנים במילונית – תוצאות ה train ו test.

תוצאות ההרצה:

:Train

Root Mean Squared Error (RMSE): 166,758

R-squared (R^2): 0.87

:Test

Root Mean Squared Error (RMSE): 325,273

R-squared (R^2): 0.6

הערה: שלב זה שימושי בשביל השוואת תוצאות המודל הראשי שבו אני משתמש (deep learning regressor model) למודל הבסיסי (linear regression).

```
{'dl_reg': {'train': {'rmse': 137285.59533652227, 'r2': 0.9031932742013201}, 'val': {'rmse': 198752.52630668192, 'r2': 0.812160318437024}}}
```

בתמונה הזאת אפשר לראות את התוצאות של המודל deep learning regressor model כאשר הוא מפוצל ל train ו validation (לא כולל test). התוצאות מופיעות במילונית ומחולקות ל train ו validation.

תוצאות ההרצה:

:Train

Root Mean Squared Error (RMSE): 137,285

R-squared (R^2): 0.9

:Validation

Root Mean Squared Error (RMSE): 198,752

R-squared (R^2): 0.81

הערה: שלב זה שימושי בשביל למצוא את הארכיטקטורה הכי טובה למודל שלי (deep learning regressor model). מצאתי את הארכיטקטורה הכי טובה בכך שהרצתי סוגי ארכיטקטורות שונות ותוצאת ה rmse הכי נמוכה מפיקה לי את הארכיטקטורה הכי טובה. לאחר מציאת הארכיטקטורה הכי טובה אוכל לבחון את המודל על נתונים (דירות) שהוא עדיין לא ראה (test).

דוח הכולל ריכוז כל ה Hyper parameters (מציג את הארכיטקטורה של המודל
הכי טוב שלי)

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	34048
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 256)	65792
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 32)	8224
dropout_3 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 256)	8448
dropout_4 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 1)	257

Total params: 182561 (713.13 KB)

Trainable params: 182561 (713.13 KB)

Non-trainable params: 0 (0.00 Byte)

תיעוד כל השינויים שנעשו במודל וב - Hyper parameters לשיפור תוצאות
האימון(הרצתי המון סוגי ארכיטקטורות שונות - להלן דוגמא של שני סוגי
ארכיטקטורות שהשתמשתי בעבר)

ארכיטקטורה 1:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	34,048
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

Total params: 75,265 (294.00 KB)
Trainable params: 75,265 (294.00 KB)
Non-trainable params: 0 (0.00 B)

ניתן לראות שישנם 3 שכבות:

שכבה ראשונה – 256 ניוירונים

שכבה שנייה - 128 ניוירונים

שכבה שלישית – 64 ניוירונים

תוצאות המודל (validation | train):

```
'dl_reg': {'train': {'rmse': 154852.32472749002, 'r2': 0.8768339402066218}, 'val': {'rmse': 209232.1534532667, 'r2': 0.7918296473404709}}}
```

תוצאות ההרצה:

:Train

Root Mean Squared Error (RMSE): 154,852

R-squared (R²): 0.87

:Validation

Root Mean Squared Error (RMSE): 209,232

R-squared (R²): 0.79

ארכיטקטורה 2:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	2,128
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 16)	272
dropout_1 (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 16)	272
dropout_2 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 16)	272
dropout_3 (Dropout)	(None, 16)	0
dense_4 (Dense)	(None, 16)	272
dropout_4 (Dropout)	(None, 16)	0
dense_5 (Dense)	(None, 1)	17

Trainable params: 3,233 (12.63 KB)
Non-trainable params: 0 (0.00 B)

ניתן לראות שישנם 5 שכבות:

שכבה ראשונה – 16 ניוונים

שכבה שנייה - 16 ניוונים

שכבה שלישית – 16 ניוונים

שכבה רביעית – 16 ניוונים

שכבה חמישית - 16 ניוונים

תוצאות המודל (validation | train):

```
'dl_reg': {'train': {'rmse': 177181.228283184, 'r2': 0.8387532370045584}, 'val': {'rmse': 240485.58905486387, 'r2': 0.7249952840354266}}
```

תוצאות ההרצה:

:Train

Root Mean Squared Error (RMSE): 177,181

R-squared (R^2): 0.83

:Validation

Root Mean Squared Error (RMSE): 240,485

R-squared (R^2): 0.72

ניתן לראות שהתוצאות של ארכיטקטורה 1 וגם ארכיטקטורה 2 (התוצאות בשלב שפיצלתי ל validation ו train) מפיגות תוצאות פחות טובות מהתוצאות של המודל הכי טוב שלי.

תיעוד והסבר של ייעול ההתכנסות (Optimization)

אופטימיזרים (optimizer) הם אלגוריתמים המשמשים להתאמת המשקלים (weights) של רשת הנוירונים כדי למזער את פונקציית ה Loss במהלך האימון. המטרה העיקרית של אופטימיזר היא למצוא סט של משקלים שמביאים לביצועים הטובים ביותר של המודל על נתוני האימון ובאופן אידאלי גם על הנתונים שהמודל עוד לא ראה (לא מכיר).

לפרויקט שלי אני בחרתי להשתמש באופטימיזר Adam שהפיק לי ביצועי מודל טובים ועזר לי למצוא ביעילות את המשקלים האופטימליים עבור הפרויקט שלי.

תיעוד ההתמודדות עם הטיה ושונות (שגיאת אימון ושגיאת מבחן)

בפרויקט שלי השתמשתי בפונקציית loss - Mean Squared Error (MSE).

MSE היא סוג של פונקציית loss המחשבת את הממוצע של ריבועי השגיאות כאשר השגיאה עבור כל נקודת נתונים היא ההפרש בין הערך בפועל לערך החזוי. פונקציית loss מסוג MSE משמשת במיוחד עבור בעיות רגרסיה כאשר המטרה זה לחזות ערכים מתמשכים כמו מחירי דירות בפרויקט שלי.

לסיכום, החלטתי להשתמש ב MSE כפונקציית loss כי המודל שלי שמטרתו חיזוי מחירי דירות יכול להשיג ביצועים מאוזנים ומוטבים, תוך התמקדות במזעור טעויות חיזוי משמעותיות ומתן מדד ברור להצלחה.

שלב היישום:

תיאור והסבר כיצד היישום משתמש במודל

כאשר היישום מריץ את המודול app.py הוא יצטרך לבחור מבין שלושת האופציות:

1. האם הוא רוצה לבנות מודל משלו
2. האם הוא רוצה לראות את המודל הכי טוב עם התוצאות שלו
3. האם הוא רוצה לראות קובץ אקסל המראה יחס של דירות לא תקינות(מבחינת המחיר)
4. האם הוא רוצה לראות את התוצאות של המודל הבסיסי – Linear regression
5. האם הוא רוצה לראות את התוצאות של המודל הכי טוב בשלב פיצול הנתונים לאימון וולידציה
6. האם הוא לא רוצה להריץ כלל/שוב את הפרויקט

הערה: התקשורת עם המשתמש תתבצע דרך ה TERMINAL

להלן איך זה נראה בסביבת עבודה ב visual studio code ב TERMINAL

```
Choose an option:
1. Enter layer parameters
2. Get the best model
3. Detect outliers by linear regression
4. Get linear regression model results
5. Get the best model results when it's split into train and validation
6. Exit
Enter your choice: █
```

במידה והיישום בוחר באופציה 1:

היישום יתבקש להזין את מספר השכבות למודל שהוא רוצה לבנות (Enter the number of layers) (you want to configure) – 3/4/5 ולאחר מכן, בהתאם למספר השכבות, המשתמש יתבקש להזין את מספר הנוירונים שהוא מחליט לבחור לכל שכבה (Enter the number of parameters for layer 1\2\3\4\5). לאחר שמשתמש בחר את מספר הנוירונים לכל שכבה, הוא יתבקש להזין את גודל ה learning rate למודל (Please enter the learning rate) ולאחר מכן הוא יתבקש להזין את מספר האיטרציות (Please enter the number of epoch). לבסוף יופיע תיאור גרפי של המחיר שהמודל חזה לעומת המחיר האמיתי, תיאור גרפי של פונקציית ה loss ותוצאות המודל שיצרת.

במידה והיישום בוחר באופציה 2:

תחילה יוצג לפניך הפרמטרים שהשתמשתי בשביל ליצור את הארכיטקטורה הכי טובה למודל. לאחר מכן הפרויקט ירוץ למשך זמן מסוים ולבסוף יופיע תיאור גרפי של המחיר שהמודל חזה לעומת המחיר האמיתי, תיאור גרפי של פונקציית ה loss ותוצאות המודל.

במידה והיישום בוחר באופציה 3:

היווצר קובץ במיקום בו החלטת לשמור אותו בשם validation_price_v2 המכיל את ה dataset עם היחס של הדירות שמודל בסיסי חזה לעומת המחירים האמיתיים וכך נוכל לזהות את הדירות החריגות.

במידה והיישום בוחר באופציה 4:

יוצג תוצאות המודל הבסיסי – Linear regression.

במידה והיישום בוחר באופציה 5:

יוצג תוצאות המודל הכי טוב שלי בשלב ה validation ו train.

במידה והיישום בוחר באופציה 6:

יוצג כפלט במסך ה TERMINAL ההודעה – "Exiting the program" והרצת הפרויקט תסתיים.

הערה: לאחר בחירת אחת האופציות, תוכל לבחור שוב את אחד האופציות שתרצה. בנוסף, חשוב לי להדגיש במידה ויש גרפים באופציה שבחרת, הגרפים יופיעו בסיום ההרצה במסך שלך, במידה ויש גם קבצים תצטרך לשנות את מיקומם והם יהיו במקום ששמרת אותם ושאר הדברים(תוצאות המודל וכו) יופיעו במסך ה TERMINAL.

תיאור קוד הקלט הקולט את ה – DATA (דאטה סט) שעליו יבוצע החיזוי והתאמתו למבנה נתונים המתאים לחיזוי

```
df = pd.read_csv(r'D:\project_visual_studio_code\Final_Project_Apartment_Prices.csv', encoding='unicode_escape')
```

להלן הסבר על כל חלק בשורת הקוד הנ"ל:

df - משתנה הקולט את התוצאה של הפונקציה pd.read_csv. משתנה זה מייצג DataFrame שהוא מבנה נתונים בספריית Pandas במשמש לאחסון וטיפול בנתונים בעלי טבלאות.

pd.read_csv - פונקציה מספריית Pandas שמטרתה לקרוא קובץ של ערכים מופרדים בפסיקים לתוך Data Frame.

Path (נתיב) של קובץ האקסל שממנו אני קורא את הנתונים. ה r שלפני המחרוזת אומרת לפייטון לפרש סלאש אחורי (\) כתווים מילוליים. במילים אחרות, זה עוזר יותר טוב לקרוא את הנתונים.

'encoding='unicode_escape' - זה מציין את סוג הקידוד המשמש לקריאת הקובץ.

לסיכום, שורת הקוד הנ"ל קוראת את הנתונים מקובץ אקסל מסוג csv, מטפלת בכל קידוד תווים מיוחד ומאחסנת את הנתונים ב DataFrame לשימוש נוסף במשימת ניתוח נתונים או מניפולציה של נתונים.

```
model_df=encoded_df
masked_train=model_df.loc[:, 'Year']<=2021
masked_test=model_df.loc[:, 'Year']>=2022
train,test=model_df.loc[mask_train,:],model_df.loc[mask_test,:]
```

להלן הסבר על כל חלק בשורות הקוד הנ"ל:

model_df = encoded_df - שורה זו מקצה את ה DataFrame encoded_df למשתנה חדש בשם model_df. זה יוצר DataFrame נפרד שישמש ל train ו test.

masked_train = model_df.loc[:, 'Year'] <= 2021 - בשורה הזאת, לפי העמודה של השנה שיש לי בדאטה סט ששמור במשתנה model_df, אני מכיל במשתנה masked_train את כל הדירות עד שנת 2021 כולל (2009-2021).

masked_test = model_df.loc[:, 'Year'] >= 2022 - בשורה הזאת, לפי העמודה של השנה שיש לי בדאטה סט ששמור במשתנה model_df, אני מכיל במשתנה masked_test את כל הדירות בין אחרי שנת 2022 כולל (2022-2023).

train, test = model_df.loc[mask_train, :], model_df.loc[mask_test, :] - בשורה זו, אני מפצל את הנתונים ששמורים במשתנים masked_train ו masked_test למערך נתונים ששמם של המשתנים train ו test. model_df.loc[mask_train, :] חלק זה בוחר את כל השורות (דירות) כאשר התנאי של המשתנה masked_train מתקיים (כל הדירות בין השנים 2009-2021). model_df.loc[mask_test, :] חלק זה בוחר את כל השורות (דירות) כאשר התנאי של המשתנה masked_test מתקיים (כל הדירות בין השנים 2022 - 2023).

לסיכום, המטרה של 4 שורות הקוד הנ"ל היא לפצל את מערך הנתונים ל train ו test.

מדריך למפתח

הפרויקט שלי מחולק ל modules אשר כל אחד מהם מכיל פעולות שימושיות לתחומים שונים. חלוקה זו מארגנת את השימוש בקוד, הופכת את הפרויקט לקל יותר להבנה, ומאפשרת לבצע שינויים לקוד בקלות.

הערה: קיימים הסברים נוספים באנגלית בפרויקט עצמו אשר כתובים בהערות בנוסף להסברים בעברית בעמודים הבאים.

שם הקובץ ותפקידו:

app.py – הקובץ הראשי שממנו מריצים את הפרויקט. מכיל בתוכו את ממשק המשתמש.
model.py – מכיל בתוכו את הקוד לבניית המודל הרצוי deep learning regressor model.
graphs.py – מכיל בתוכו את הקוד ליצירת הגרפים.
linear_regression_for_outliers.py – מכיל בתוכו את הקוד לזיהוי דירות שגויות.
linear_regression_model.py – מכיל מודל בסיסי linear regression.
best_model_train_validation.py – מכיל את הארכיטקטורה הכי טובה של המודל deep learning regressor model בשלב פיצול הנתונים ל train ו validation.
Final_Project_Apartment_Prices.csv – קובץ זה מכיל את מאגר הנתונים (Data set)

תתי הקבצים הנוצרים דרך הקבצים הנ"ל:

model_plot.png – בקובץ graphs.py נוצר קובץ חדש אשר יוצר תמונה של המודל.
validation_price_v2.csv – בקובץ linear_regression_for_outliers.py נוצר קובץ חדש אשר יוצר קובץ אקסל (Excel) לסינון הדירות הלא רצויות.
result_model_df – בקובץ model.py נוצר קובץ חדש אשר יוצר קובץ אקסל המציג את הדירות והנתונים עליהם ויוצר עמודה חדשה של מחירי הדירות שהוא חזה בשלב ה test ו train.

הערה: בכל קובץ ישנו פעולה אחת (הפעולה היא תפקידו של הקובץ) למעט הקבצים model.py, app.py אז אני הסביר רק על הפעולות בקבצים הללו.

הפעולות בקובץ model.py ותפקידם:

`model(layer_params, learning_rate, epoch_number)` – הפונקציה הראשית בקובץ בשביל להריץ את כל המודל. הוא מייבא את הנתונים, מעבד את הנתונים, בונה את המודל, מאמן (train) ובודק (test) את המודל ומעריך את התוצאות.

`import_dataset()` – מכיל את הנתונים (Data set)

`data_processing(df)` – מבצעת עיבוד מקדים של הנתונים כולל שימוש בשיטה one hot-encoding, פיצול הנתונים ל train ו test וביצוע scaling על התכונות.

`deep_learning_reg(scaled_x_train, y_train, scaled_x_test, y_test, layer_params, learning_rate)` – יוצר באופן דינמי (בהתאם לערכים שמוזנים לו) מודל deep learning regressor `model`.

`fit_predict_model(model, scaled_x_train, y_train, scaled_x_test, y_test, df, masked_train, masked_test, epoch_number)` – מתאים את המודל על נתוני ה train, מבצע תחזיות על ה train וגם על ה test, מחשב את תוצאות ה RMSE ו R2 ויוצר הגרף המראה את תוצאות מחירי הדירות של ה train ו test.

הפעולות בקובץ app.py ותפקידם:

`get_layer_parameters()` – הפונקציה מבקש מהמשתמש להזין את מספר השכבות, כמות הנוירונים לכל שכבה, קצב הלמידה וכמות האיטרציות. במידה והמשתמש הכניס ערך שגוי, מוצגת הודעה מתאימה לכך. הפונקציה מחזירה את הערכים שהוזנו בשביל להגדיר את המודל.

`main()` – הפונקציה הראשית בפרויקט אשר משמשת לתקשורת עם המשתמש. תקשורת עם המשתמש מבוצעת דרך ה TERMINAL וניתנת לו אפשרויות שונות בהתאם למה שהוא מחליט לענות.

הסבר על המשתנים החשובים בפרויקט:

`parameters` – מכיל את כמות הנוירונים שהמשתמש בחר

`learning_rate` – מכיל את קצב הלמידה שהמשתמש בחר

`epoch_number` – מכיל את כמות האיטרציות שהמשתמש בחר

`choice` – מכיל את הקלט שהמשתמש בחר להזין (יכול להזין אך ורק אחד מהמספרים 1-6)

`layer_params`: רשימה המכילה את מספר הנוירונים עבור כל שכבה.

`learning_rate`: קצב למידה עבור האופטימיזר.

`epoch_number`: מספר האיטרציות לאימון המודל.

`df` – מכיל את ה Dataset

`encoded_df` – מומר בו שלושה עמודות לעמודות בינאריות, מומר בו כל ערך בוליאני למספרים שלמים והורדת העמודה Date.

`scaled_x_train`: תכונות מותאמות לאימון.

`y_train`: משתנה יעד לאימון.

`scaled_x_test`: תכונות מותאמות לבדיקה.

y_test: משתנה יעד לבדיקה.

masked_train: מכיל את כל הדירות עד שנת 2021 כולל עבור האימון.

masked_test: מכיל את כל הדירות שגדולים משנת 2021 (2022 – 2023 לפי הדאטה סט שיש לי) עבור הבדיקה.

input_shape: מספר התכונות בשכבה הראשונה (שכבת הקלט) של הרשת נוירונים.

model – מכיל את כל המודל.

opt – מכיל את האופטימיזר "Adam" שמוגדר אם קצב למידה אשר מגדירים לו.

results – מילונית שמאחסנת את התוצאות ה RMSE ו R2.

history - מכיל האימון והתאמתו למודל.

y_prediction_train – חוזה את מחירי הדירות בשלב האימון.

y_prediction_test – חוזה את מחירי הדירות בשלב הבדיקה.

result_model_df – מכיל את תוצאות החיזוי של האימון והבדיקה.

price_compare – מכיל את מחירי דירות האמיתיות ומחירי הדירות שהמודל חזה (מיועד בשביל הצגה גרפית).

x,y – מייצגים את הקו המגמה האידאלי.

loss – לוגריתם של ערכי ה loss של האימון לפי כמות האיטרציות (ציר y).

epochs – טווח הערכים של כמות האיטרציות שנועדו ליצירת הגרף (ציר x).

plt.show(): מציג את הגרף הרצוי.

reg – משומש בשביל להתאים את המודל וליצור חיזויים.

y_score – מכיל את החיזויים שנוצרו על פי מודל בסיסי של רגרסיה לינארית.

price_validation_df – מכיל את המחירים המקוריים, המחירים שהמודל חזה והיחס שנוצר ביניהם.

Lasso – זה מחלקה שמשומשת לבצע רגרסיה שזה סוג של מודל של רגרסיה לינארית.

summary – מכילה את תוצאות ה RMSE ו R2.

x_train – מכיל את כל התכונות של הדירה לאימון

y_val – מכיל את משתנה היעד לוולידציה (מחיר הדירה)

x_val - מכיל את כל התכונות של הדירה לוולידציה

scaled_x_val – מכיל תכונות מותאמות לוולידציה.

y_prediction_val – חוזה את מחירי הדירות בשלב הוולידציה.

להלן הסבר קצר על כל ספרייה בה השתמשתי לצורכי הפרויקט שלי:

- Seaborn – ספרייה המשמשת לויזואליזציה של הנתונים המבוסס על ספריית Matplotlib
- TensorFlow – ספרייה המציעה מערכת אקולוגית מקיפה לבניית ואימון מודלים של למידת מכונה למידה עמוקה
- NumPy – ספרייה לנרמול המספקת תמיכה במערכים, מטריצות ואוסף גדול של פונקציות מתמטיות
- Pandas – ספריית מניפולציה וניתוח נתונים המספקת מבני נתונים כמו DataFrames כדי לבצע מניפולציה יעילה של נתונים מובנים
- Matplotlib – ספרייה ליצירה והצגה ויזואלית של הדמיות מונפשות, גרפים ותרשמים
- Sklearn – ספריית למידת מכונה המספקת כלים פשוטים ויעילים לניתוח נתונים ולמידת מכונה כולל אלגוריתמים לרגרסיה
- Datetime – מאפשר מניפולציה של תאריכים וזמנים

מדריך למשתמש

תרשים מסכים המתאר את היררכיית המסכים עם הסברים

קישור להתקנת סביבת העבודה - visual studio code ב Windows, Linux, Mac:

<https://code.visualstudio.com/download>

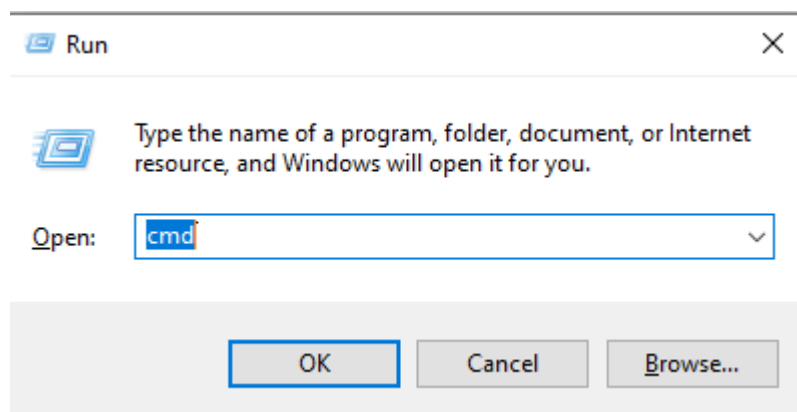
להלן תמונה של המסך של הורדת סביבת העבודה:



תבחר את המערכת שאתה משתמש ואז תלחץ על המלבן הכחול בשביל להוריד את סביבת העבודה.

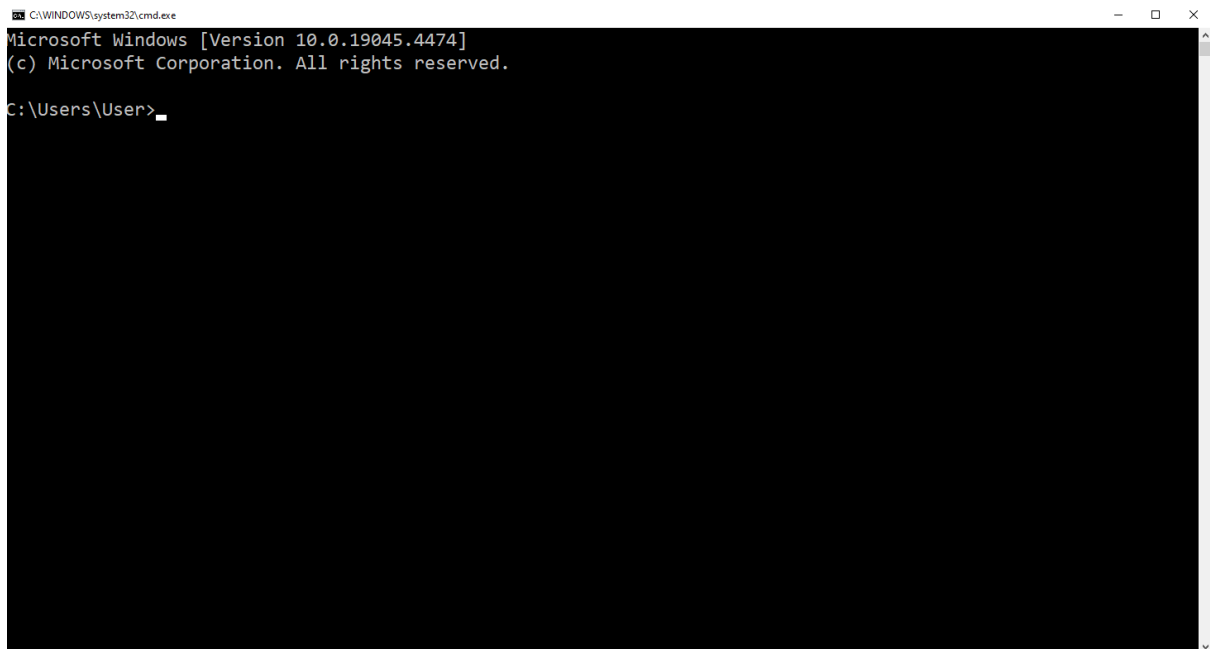
התקנת הספריות מתקיימת ב CMD. בשביל להיכנס ל CMD אפשר להשתמש בקיצור במקלדת - windows ולחיצה על המקש R בו זמנית ואז הפתח חלונית ששם נכתוב cmd.

להלן היררכיית המסכים למעבר ל CMD:



לאחר שימוש בקיצור במקלדת - windows ולחיצה על המקש R בו זמנית הפתח החלון המופיע למעלה. לאחר פתיחת החלון נצטרך לכתוב cmd(כמו שמופיע בתמונה)

לאחר מכן הפתח לנו ה CMD. תמונה המראה איך נראה ה CMD:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.4474]
(c) Microsoft Corporation. All rights reserved.
C:\Users\User>
```

ב CMD נצטרך להתקין את הספריות בהם אני משתמש לצורכי הפרויקט.

התקנת הספריות תמומש על ידי כתיבת הפקודות בדרך הבאה:

הערה: צד שמאל למקף אלו הם הפקודות שאנו צריכים להקליד ב CMD וצד ימין אלו שמות הספריות שאנו רוצים להתקין.

pip install seaborn – Seaborn

pip install tensorflow - TensorFlow

pip install numpy - NumPy

pip install pandas - Pandas

pip install matplotlib - Matplotlib

pip install scikit-learn - Sklearn

פעולות נדרשות לשם הרצת התוכנית:

לפני הרצת התוכנית, תצטרך לשנות את מיקומי הקבצים הבאים לפי ההוראות שכתובים מטה.

1. ראשית, יש לשמור את קובץ האקסל(קובץ הדאטה סט - Dataset) ב Path(מיקום) שנוח לך.

לאחר שמירת קובץ הדאטה סט, יש לשנות בקוד של הפרויקט את מיקום הדאטה סט בו שמרת אותו(זה מופיע במודול model.py):

```
df = pd.read_csv(r'D:\project_visual_studio_code\Final_Project_Apartment_Prices.csv', encoding='unicode_escape')
```

הערה: החלק במחרוזת לפני הסלאש הראשון זה מיקום הכונן. אחרי הסלאש הראשון זה מיקום הספרייה בה תרצה לשמור. לאחר הסלאש השני אתה כותב את שם הקובץ ומסיים עם סוג הקובץ(במקרה שלנו סוג הקובץ הוא csv).

בחלק של D:\project_visual_studio_code אתה מחליף למיקום בו שמרת את הדאטה סט. במידה ותרצה לשנות את שם הקובץ תוכל לעשות זאת בכך שתחליף במקום Final_Project_Apartment_prices את השם שבחרת.

2. תצטרך גם כן לשנות את מיקום הקובץ שמציג תיאור גרפי של המודל(זה מופיע במודול graphs.py):

```
plot_model(model, to_file="D:/project_visual_studio_code/model_plot.png", show_shapes=True, show_layer_names=True)
```

שינוי מיקום הקובץ יתבצע כך שתבחר את המיקום בו תרצה שהקובץ ישמר ותחליף את המיקום שבחרת במקום המחרוזת - D:/project_visual_studio_code. במידה ותרצה לשנות את שם הקובץ תוכל להחליף את השם שבחרת במקום model_plot המופיע בקטע קוד הנ"ל.

3. בנוסף, בחלק שבו יצרתי אקסל שבו מופיעים תוצאות החיזוי של מחירי הדירות של ה train ו test(חיזוי כל הדירות), תצטרך לשנות את מיקום הקובץ שבו תרצה שהוא יופיע(זה מופיע במודול graphs.py):

```
result_model_df.to_csv("D:/project_visual_studio_code/result_model_df.csv")
```

בתחילת המחרוזת(לאחר הגרשיים), במקום D:/project_visual_studio_code תשנה למיקום בו תרצה לשמור את קובץ התוצאות. במידה ותרצה לשנות את שם הקובץ תוכל להחליף את השם שבחרת במקום result_model_df המופיע בקטע קוד הנ"ל.

4. שינוי מיקום יצירת קובץ האקסל לזיהוי דירות חריגות תתבצע באופן המתואר מטה(זה מופיע במודול linear_regression_for_outliers.py):

```
price_validation_df.to_csv("validation_price_v2.csv")
```

תוכל לבחור את מיקום הקובץ שבו הקובץ הנ"ל יופיע בכך שתעתיק את המיקום לתחילת המחרוזת(לאחר הגרשיים) ואז את שם הקובץ(במקרה הזה validation_price_v2)

לדוגמא:

```
price_validation_df.to_csv("C:/project_visual_studio_code/validation_price_v2.csv")
```

רפלקציה / סיכום אישי

במהלך פרויקט הגמר שלי בהתמחות למידת מכונה עברתי מלא קשיים בדרך אך למדתי והצלחתי להתגבר עליהם בכל מידי דרכים כאשר כל קושי שאני מתגבר עליו כך הרגשתי מאין סיפוק אישי וקבלת מוטיבציה לפרויקט. למדתי מלא מהפרויקט וגם מכתובת הספר של הפרויקט ואני מרגיש שאני מצליח להשתפר ולחזק ולחדד את הכישורים שלי.

לו הייתי עושה פרויקט זה מחדש הייתי מתכנן טוב יותר את לוחות הזמנים שלי – לא הייתי דוחה את רוב העבודה לסוף ופורס את זמן העבודה שלי לכך אורך השנה. כאשר נתקלתי בבעיה, לא היה לי הרבה זמן לתוקן, מה שהוביל ללחץ בכתובת הפרויקט.

לסיכום, רוצה לומר תודה לבית ספר שלי שבחר לאפשר לנו ללמוד נושא מעניין וחשוב זה ולבצע את ההתמחות שלנו בתחום הזה. אני מאמין שתחום זה הינו תחום בעל עתיד רב ולכן לעסוק בו כבר בבית הספר זו זכות גדולה בשבילי, אני מאמין שידע זה כוח, וכאשר אני מבין יותר ויותר בנושא זה – אני מאמין שזה חשוב מאוד.

ביבליוגרפיה

אתר הנדל"ן הממשלתי – gov נדל"ן: [/https://www.nadlan.gov.il](https://www.nadlan.gov.il)

נספחים

חשוב לי להדגיש שהמודל linear regression מפיק אותם תוצאות כמו המודל הכי טוב שלי deep learning regressor model או אפילו תוצאות טובות יותר מכיוון שהוא מודל מורכב יותר שמצריך להשתמש בכמות גדולה של דאטה סט. בנוסף, ישנו overfitting בחלק הסופי בו פיצלתי את המודל הכי טוב שלי ל train ו test – כאשר היישום בוחר אופציה מספר 2 (ניתן לשים לב כי ישנם הבדלים גדולים בין התוצאות של ה train ו test של ה r^2 ו rmse), אשר גרם לחיזויים לא טובים והסיבה לכך היא כמו שצוין קודם, עקב הדאטה סט היחסית קטן שיש לי, בנוסף לכך, החיזויים הלא טובים נבעו גם בגלל מיעוט בתכונות (features) - כמות תכונות גדולה יותר כנראה הייתה עוזרת למודל ללמוד טוב יותר ולהפיק תוצאות טובות יותר.