פרויקט בבינה מלאכותית

חיזוי מחירי דירות בניו-יורק

מגישים :

אמיר אביבי - *********

איתי ישראלוב - *******

מנחה :

דרור סימון

תוכן עניינים

1. מבוא

מאז ומתמיד, שוק הנדל"ן (Real Estate) היווה מוקד עניין גדול בחברה האנושית.
האופי הדואלי (קורת גג + נכס פיזי בשטח) של נכסים אלו, גורמים לכך שאופי המשקיעים בענף הוא רחב מאוד, החל ב"אנשים פשוטים" המחפשים נכס למגורים וכלה במשקיעים ממולחים, אשר שואפים למקסם את ההחזר על ההשקעה בנכס (ROI-Return On Investment).
בנוסף, מקצועות רבים כגון מתווכים, שמאים ועורכי דין וכו' מתמתחים ספציפית בענף הנדל"ן לאור הביקוש ההולך וגובר בענף זה.
הקושי לאמוד את שווי הנכס, הן מצד המוכר והן מצד הקונה הוא לב ליבו של המשא ומתן כמעט בכל עסקת מכירה.

1.1. הצגת הבעיה

עקב הסיבות שנמנו מעלה, מחירי נכסי נדל"ן הינם מגוונים מאוד. מחירים אלו מושפעים מגורמים רבים כגון: מיקום הנכס, מחירי נכסים דומים באזור, קירבה של נקודות עניין לנכס ועוד רבים אחרים. אין ספק שכלי יעיל לאומדן מחיר נכס נדלן, הוא בעל ערך רב הן לקונים פוטנציאלים והן למוכרים פוטנציאלים.

הפרויקט יעסוק בשערוך מחירי נכסי הנדל"ן בעיר ניו-יורק שבארה"ב, שהיא אחת מן הערים המבוקשת ביותר בעולם, אם לא הכי מבוקשת, מבחינה נדל"נית.

1.2. מטרת הפרויקט

מטרת הפרויקט היא לספק כלי אלגוריתמי יעיל לאומדן מחירי נכסי הנדל"ן בניו-יורק.

1.3. תיאור הפתרון המוצע

במסגרת הפרויקט בחרנו להתמודד עם הבעיה כבעיית למידה. מכיוון שמדובר במשא ומתן בין בני-אדם, מחירו של נכס כלשהו נקבע באופן מובהק בעזרת דוגמאות דומות ונוספות המקשרות בין הנכס המדובר לנכסים אחרים דומים וקשורים. אופי זה של הבעיה הביא אותנו להניח כי פתרון, המבוסס על אלגוריתם למידה יהיה הכלי היעיל ביותר לבעיה זו.

השלבים בפתרון שלנו:

1. <u>איסוף נתונים:</u> את החלק הארי של הנתונים הפקנו ממאגר הנתונים הרשמי של עיריית ניו-יורק, אותו ניתן למצוא בקישור הבא.
2. <u>חקר הנתונים:</u> לאחר איסוף הנתונים, הלכנו וחקרנו מהו בעצם כל פיצ'ר המיוצג בנתונים, מה ניתן ללמוד מערך כזה או אחר בפיצ'ר מסוים (לדוגמה, לכל נכס משויכת מחלקת מס הנקבעת בין היתר ע"י אופי השימוש בנכס) ובאופן כללי, מה התובנות שניתן להסיק מהנתונים ע"מ להבין טוב יותר את הענף(נדל"ן) בו אנו מתעסקים בפרויקט.
3. <u>עיבוד מקדים (Pre-Processing) של הנתונים:</u> בשלב זה חקרנו את אופי הנתונים שאספנו. היכן חסרים ערכים, האם החוסרים מובנים, האם כל הערכים הגיוניים, אם לא מדוע וכו'.
4. <u>בניית מודלים ראשונית:</u> בשלב זה בדקנו מספר מודלים ואלגוריתמים שונים עבור בעיות רגרסיה ניתחנו את תוצאותיהם וערכנו ביניהם השוואה. המטרה של שלב זה הייתה ליצור בסיס השוואתי לשלב המודלים המאוחר יותר. דוגמא לאלגוריתם בו השתמשנו הוא $Random\ Forest\ Regression$.
5. <u>יצירת תכונות חדשות (Feature Generation):</u> בשלב זה השתמשנו ב – Google API ע"מ לייצר תכונות נוספות על הנכסים שברשותנו. בשלב הראשון הוצאנו את קווי האורך והגובה של כל נכס בעזרת Google Geocoding ולאחר מכן ייצרנו פי"צרים חדשים עבור הנכסים, לדוגמא: מספר בתי הקפה ברדיוס מסוים מהנכס, כמות האטרקציות ברדיוס מסוים מהנכס, האם קיימת סוכנות נדלן בקרבת הנכס וכו'. ניתחנו את

תוצאות המודלים מהשלב הקודם בשילוב הפיצ'רים החדשים שייצרנו כדי לבחון את תועלתם בחיזוי המחיר (חלק עזרו, חלק לא השפיעו וחלק אף פגעו ביכולות החיזוי).

6. בחינה סופית של המודלים עם הפיצ'רים החדשים: לאחר שייצרנו פיצ'רים חדשים בנתונים שלנו בעזרת השיטות לעיל, בחנו שוב את המודלים משלב 4 על הנתונים עם התוספות ע"מ לבדוק האם תוספות אלו עזרו בבעיית החיזוי.
7. סיכום וניתוח תוצאות: בשלב זה ביצענו ניתוח של התוצאות. ניסינו להבין מה היו אופי השגיאות, מדוע פיצ'רים מסוימים תרמו לניבוי ופיצ'רים אחרים לא, ומה היא הגישה הטובה ביותר לפתרון, מאלו שנתקלנו בהן במהלך הפרויקט.

1.4. גילוי נאות: מגבלות הפרויקט

הפרויקט שלנו השתמש באמצעים חינמיים לחלוטין. כנגזרת מהחלטה זו, הייתה לנו מסגרת תקציב שבה היה עלינו לעמוד. ההשלכות למגבלות אלו היו, בעיקר, הגבלה ושימוש מדוד מאוד ב Google API . כתוצאה מכך, בחרנו להתרכז במידע איכותי יותר, הגבלנו את כמות המידע שאספנו (למרות שמאגר הנתונים מכיל מידע משנת 2003 והלאה. הפרויקט מבוסס על נתוני 2019, ע"מ לעמוד במסגרת התקציב) והגבלנו את כמות הפיצ'רים הנוספים שחקרנו (ניתן להוציא מידע רב על כל נכס, אך כל חיפוש עולה כסף).

2. סביבת העבודה

בפרק זה נתאר את סביבת העבודה והכלים בהם השתמשנו בפרויקט ונציג כיצד ניתן לשחזר את הניסויים והתוצאות.

2.1. NYC Annualized Sales Database

הנתונים שלנו נלקחו מהאתר הרשמי של עיריית ניו יורק (קישור למידע). באתר זה ניתן למצוא פעילות של עסקאות נדלן בעיר משנת 2003 ואילך המסווגות לפי מיקום הנכס.

2.2. Google API

הלב והמגבלה העיקרית של הפרויקט הוא Google API.

ע"מ להשתמש בשירותי גוגל, נדרש להירשם באתר שירותי הענן של גוגל ולספק אמצעי חיוב עבור השירותים בהם משתמשים.

נכון ל 1.10.2020, בהרשמה הראשונית לאתר מקבלים 300 דולר לשימוש בתוך 90 הימים הראשונים.

מסגרת התקציב שלנו במהלך הפרויקט עמדה על 600 דולר, עם דד-ליין (עבור הקוד) של 90 יום.

בפרויקט שלנו השתמשנו בשני שירותים עיקריים שניתן לקבל דרך Google API.

2.2.1. Google Geocoding API

גוגל גיאוקודינג הוא התהליך שבהמרת כתובת, לדוגמה:

"1600 Amphitheatre Parkway, Mountain View, CA"

לקואורדינטות גאוגרפיות, כמו קווי אורך וגובה.

2.2.2 Google Places API

גוגל פלייסיס הוא שירות שמחזיר מידע לגבי מיקומים באמצעות בקשות HTTP. מיקומים מוגדרים בידי ה-API כמוסדות, מיקום גאוגרפי או נקודות עניין.

הבקשות האפשריות שניתן לבקש עבור כל מיקום נתונות בקישור הבא.

2.3 Python3

השפה בה נכתב הקוד. השימוש ההולך וגובר בשפה כשפה עיקרית לפיתוח פתרונות מבוססי ML הביא להימצאות של אינספור חבילות שבהן ניתן להיעזר במהלך כתיבת הקוד.

2.4. Jupyter Notebooks, Google Colab & Binder

מרבית קטעי הקוד נכתבו במחברות פייתון. השתמשנו במחברות מכיוון שניתן להעביר בהם בצורה ברורה ומחושית את התהליך המחשבתי שעברנו במהלך העבודה על הפרויקט. את כל המחברות של הפרויקט ניתן להריץ באמצעות Binder שהינו כלי חינמי המאפשר יצירת סביבה מותאמת לפי הגדרת ראשי הפרויקט (אנחנו).

2.5. GitHub

הפרויקט כולו חשוף לצפייה ציבורית בGitHub, בו השתמשנו ככלי עבור בקרת גרסאות בפרויקט.
את הקישור לפרויקט ניתן למצוא בקישור הבא.

2.6 שחזור תוצאות

בכדי לשחזר את הפרויקט ניתן להיכנס לעמוד הגיט-האב של הפרויקט, וללחוץ על הקישור להרצת הבינדר (ניתן להיכנס גם ע"י לחיצה על הקישור מטה). יפתח חלון חדש בו יוקם דוקר רלוונטי שיכיל את כל החבילות בהם השתמשנו בפרויקט, עם כל קבצי המקור, בהיררכיה הנכונה.

launch binder

*אם ברצונך לשחזר גם את תהליך יצירת הפיצ'רים, יש להירשם לשירות הגוגל פלטפורם ולהוסיף את המפתח הייחודי שלך בקובץ api-key_geocoding.txt.

3. מהלך הפרויקט

בפרק זה את נפרט שלבי העבודה על הפרויקט באופן מפורט תוך כדי מתן דוגמאות ויזואליות מהפעלת הקוד עצמו.

3.1. איסוף הנתונים

בשלב זה חיפשנו נתונים שיהוו את בסיס הנתונים ללמידה עבור המודל. כידוע, לשלב זה חשיבות מכרעת בהצלחת הפרויקט, כיוון שאם הדוגמאות שעליהן יתבססו האלגוריתמים לא יהיו דוגמאות מייצגות ומורכבות, המסווגים שייווצרו בסופו של דבר לא יהיו מוצלחים.

בשלב הראשון חיפשנו נתונים ב-Kaggle, שהינו מנוע החיפוש העיקרי למציאת Datasets בתחומים שונים. לאחר חיפושים ובחינה של מספר מאגרי נתונים, מצאנו את המאגר הבא: https://www.kaggle.com/new-york-city/nyc-property-sales. מאגר זה עודכן לאחרונה בתאריך 09.2017 ומכיל נתונים רשמיים של העירייה על מכירות נכסים בניו-יורק באותה השנה.

מידע זה נראה לנו רלוונטי והתחלנו לחקור מאין הוא מגיע.

בתום החיפושים הגענו לאתר הרשמי של עיריית ניו-יורק, שם גילינו כי מאגר הנתונים בKaggle נגזר מנתונים רשמיים של העירייה. נתונים אלו נאספים ע"י העירייה כחלק מתהליך המיסוי. מכיוון שתהליך המיסוי הוא ציבורי נתוני המכירה הם ציבוריים וחשופים לציבור הרחב.

היות וזהו מידע רשמי מגורם מוסמך (בעל אינטרס – מיסוי, מה שמגביר את אמינות הנתונים) החלטנו שזהו מאגר המידע האמין והמדויק ביותר שנוכל להתבסס עליו ולכן נבחר כמאגר הנתונים של הפרויקט.

את המאגר המלא ניתן למצוא בקישור:

https://www1.nyc.gov/site/finance/taxes/property-annualized-sales-update.page

המאגר עצמו מחולק לפי שנים ולפי אזורים בעיר. ניתן גם למצוא טבלאות סיכומים לפי שנים\ אזורים \ מחלקות מיסוי.

אנו בחרנו להתמקד בכל הנתונים משנת 2019, את הפרויקט ניתן להרחיב תחת מסגרות תקציב מתאימות כך שיכיל גם נתונים משנים נוספות.

3.2. חקר הנתונים

חלק זה מקודד ומתואר במחברת הפרויקט: 00_Data_Understanding.ipynb

לאחר תהליך איסוף הנתונים התחלנו לחקור את הנתונים שהפקנו. מה אומר כל פיצ'ר, מה ההשלכות של ערך ספציפי בפיצ'ר מסוים וכו'.

תחילה נאמר שבמאגר הנתונים שלנו אספנו נתוני עסקאות נדלן עבור שנת 2019. הנתונים חולקו לפי 5 אזורים: מנהטן, ברונקס, ברוקלין, קווינס וסטייטן-איילנד. בשלב הראשון איחדנו את כל הנתונים מכל האזורים לבסיס נתונים אחד. בשלב השני ניתחנו את המידע- המאגר הכיל 83,920 עסקאות, כשכל אחת מהן מאופיינת ע"י 21 פיצ'רים, כאשר אחד מהם הוא פיצ'ר המטרה שלנו - מחיר המכירה.

הנתונים שלנו כללו 7 פיצ'רים קטגוריילים, 1 פיצ'ר תאריך, ו13 פיצ'רים מספריים (שלמים או ממשיים).

נתמקד תחילה בפיצ'ר המטרה שלנו – מחיר המכירה.

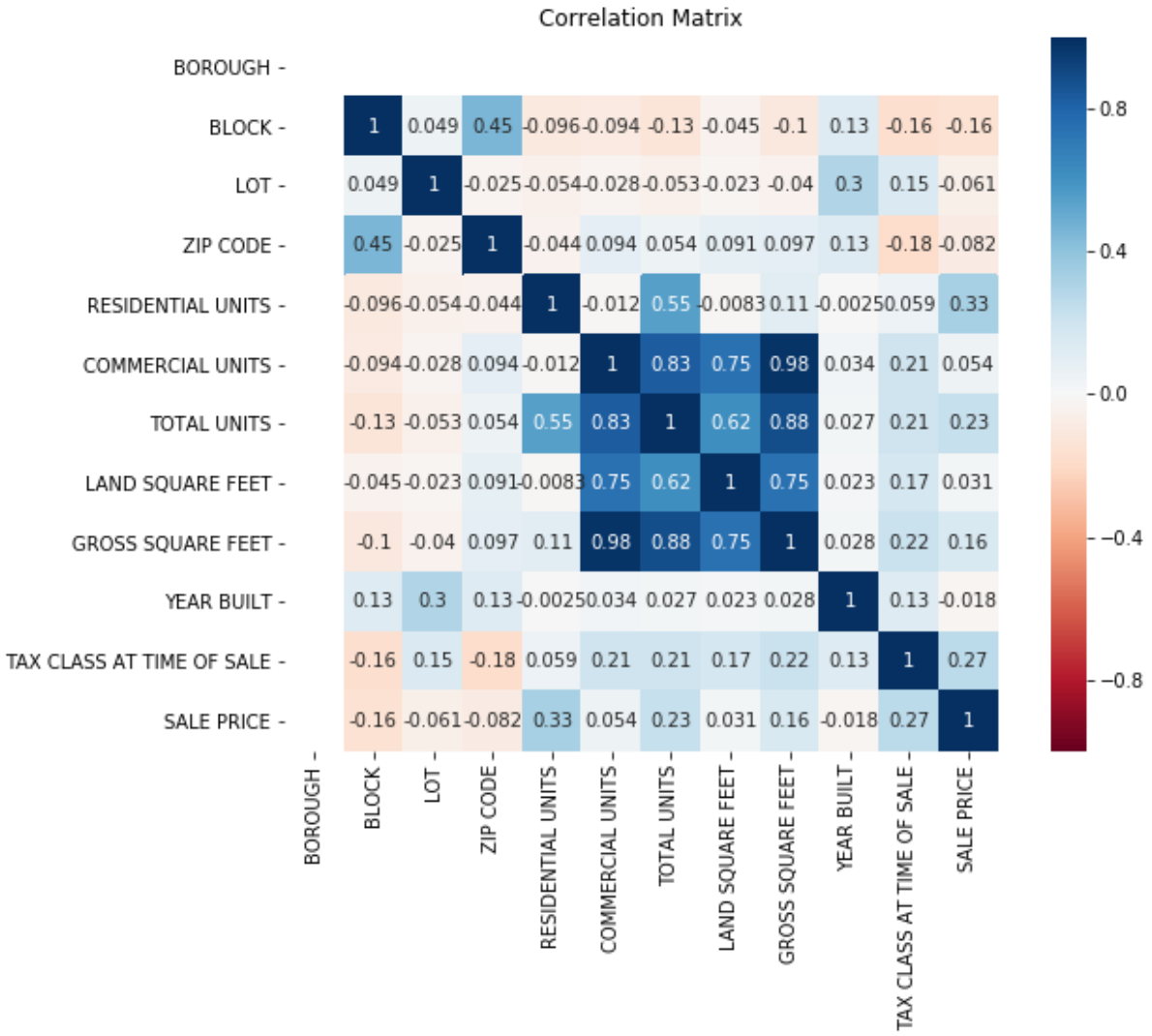
מצאנו כי 26,383 מהעסקאות נרשמו במחיר עסקה של 0$. ביררנו מה משמעות הדבר באתר העירייה וגילינו כי מחיר עסקה שכזה מייצג עסקאות נדלן שכללו מקרים חריגים כמו למשל: הורשה של הנכס או עסקה בה לא צוין מחיר המכירה אולם אלו נרשמו במאגר לצורך תיעוד. מכיוון שהפרויקט שלנו עוסק בחיזוי מחירי הנכסים, החלטנו להוריד את הנתונים הללו מהמאגר, מהסיבה שמטרת הפרויקט הינה לצפות מחירי נכסים ולא לסווג עסקאות

לירושה\מכירה.

נחזור למחיר המכירה בחלק העיבוד המקדים.

בחינה נוספת של המידע גילתה לנו כי ישנם שני פיצ׳רים (1. מספר הדירה 2. קיום של זכות שימוש בנכס למי שאיננו בעל הנכס, לדוגמא חלק ממסילת רכבת שנמצאת בשטח הנכס) שבהם הערכים המיוצגים חסרים במעל 75% מהמידע שנאסף.

בנוסף גילינו חוסרים מבניים בנתונים. חמישה פיצ׳רים המייצגים את שטח הנכס ותאור הנכס (כמות יחידות דיור, כמות יחידות מסחר, כמות היחידות הכוללת, שטח הנכס ברוטו ושטח הנכס נטו) היו חסרים באופן עקבי בחלק ניכר מהדוגמאות. לאחר ניתוח, גילינו שהפיצ׳רים הנ״ל מאוד קורלטיביים כפי שניתן לראות בגרף הנ״ל:



כעת בחנו את הערכים עצמם:

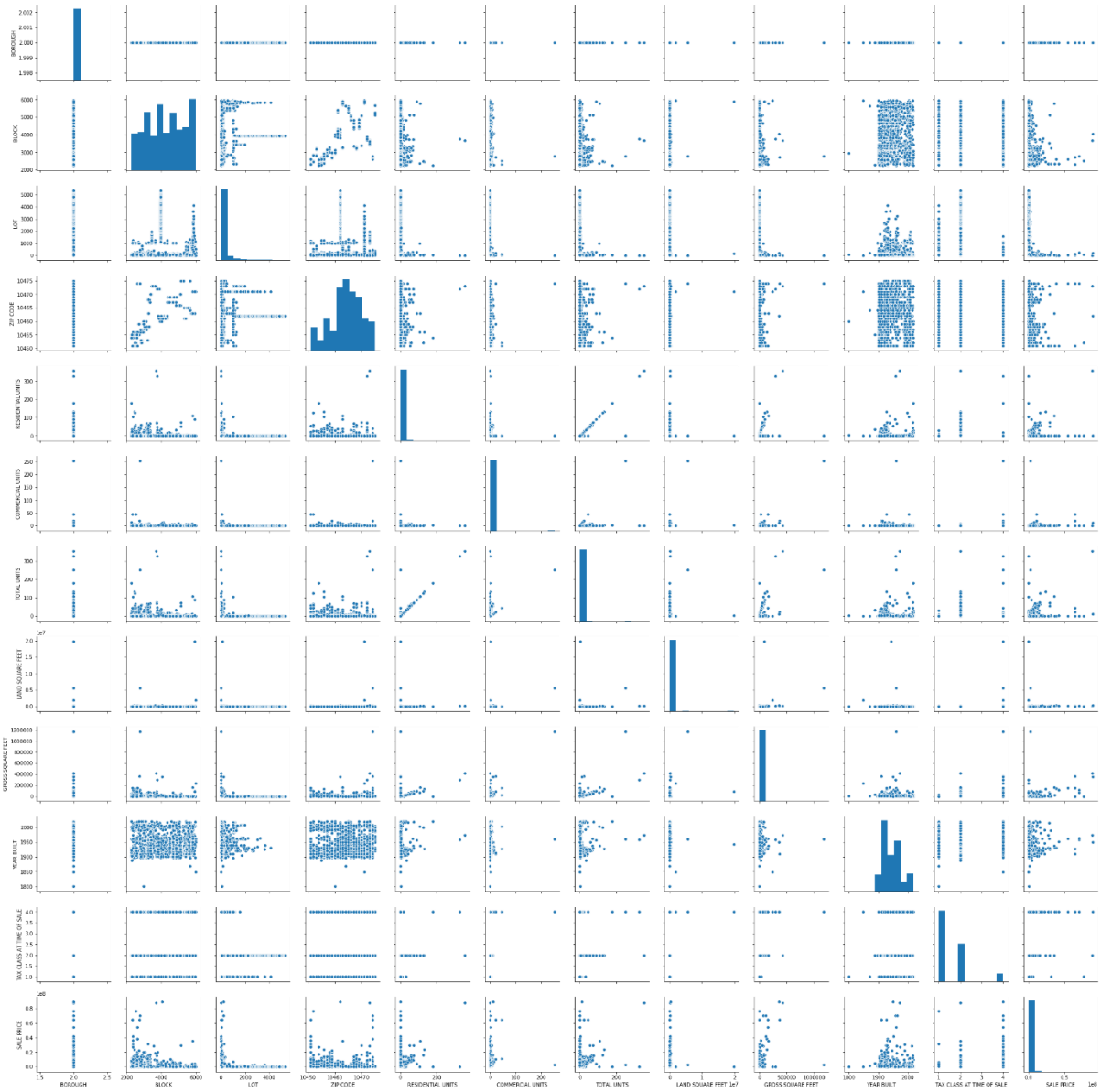
עבור ערכים מספריים - את הסטטיסטיקה שלהם (ממוצעים, סטייות תקן ואחוזונים)

עבור ערכים קטגוריילים – את טווח הערכים האפשרי והתפלגות הקטגורייות.

לאחר מכן בדקנו את הקשר בין הפיצ׳רים ע״י מטריקות הקורלציה וה mutual information.

לבסוף בחנו את ההתפלגות המשותפת בין כל שני פיצ׳רים ע״י ויזואליזציה, בה ניתן לראות פיצ׳ר אחד כפונקציה של השני.

ניתן לראות את הגרף בעמוד הבא.


BOROUGH
BLOCK
LOT
ZIP CODE
RESIDENTIAL UNITS
COMMERCIAL UNITS
TOTAL UNITS
LAND SQUARE FEET
GROSS SQUARE FEET
YEAR BUILT
TAX CLASS AT TIME OF SALE
SALE PRICE

3.3. עיבוד מקדים – Pre-Processing

חלק זה מקודד ומתואר במחברת הפרויקט: 01_Data_Preparation.ipynb

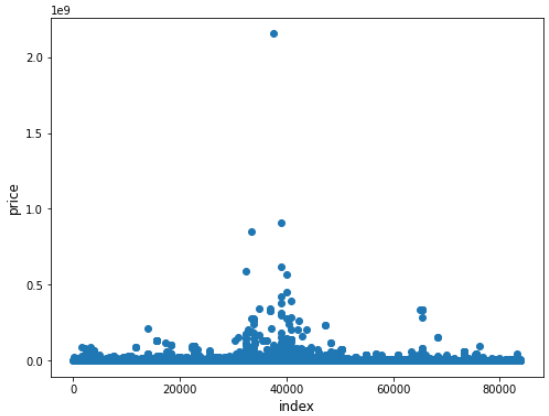
בחלק זה התמקדנו במשימות העיבוד המקדים שכללו:

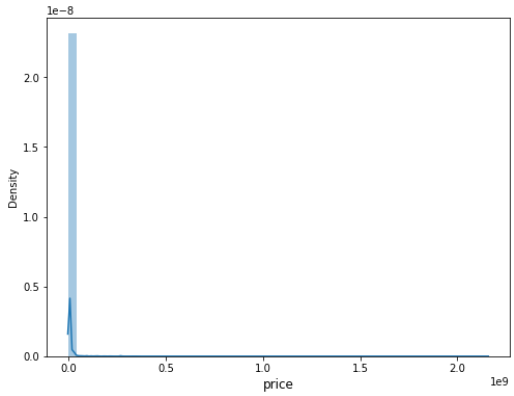
- ❖ השלמת\הסרת ערכים חסרים
- ❖ זיהוי והחרגת מקרי קצה(outliers)
- ❖ "החלקה של הנתונים" – התמקדות בעיקר ולא בטפל מבחינת הנתונים
- ❖ טרנספורמציה של הנתונים לנתונים מספריים
- ❖ סקיילינג

***בחרנו להציג בדו"ח רק חלק מתהליכי העיבוד המקדים שכן הם מכילים גרפים וויזואליזציה רבה וחוזרים על עצמם במקרים רבים, את המידע השלם ניתן למצוא במחברות הפרויקט.**

<u>מחיר המכירה:</u>

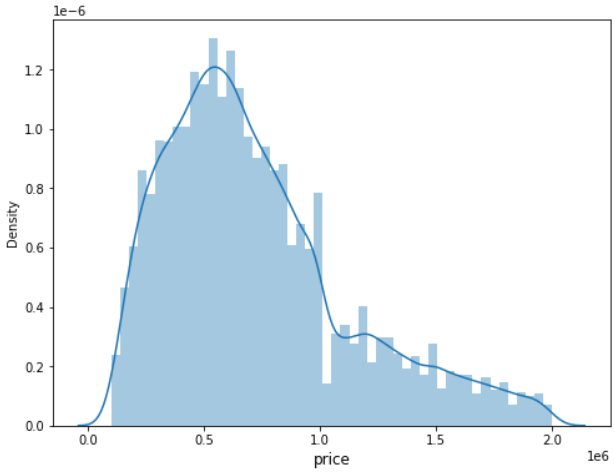
הפיצ'ר העיקרי בפרויקט הוא כמובן מחיר המכירה, נקודת הפתיחה שלנו הייתה:





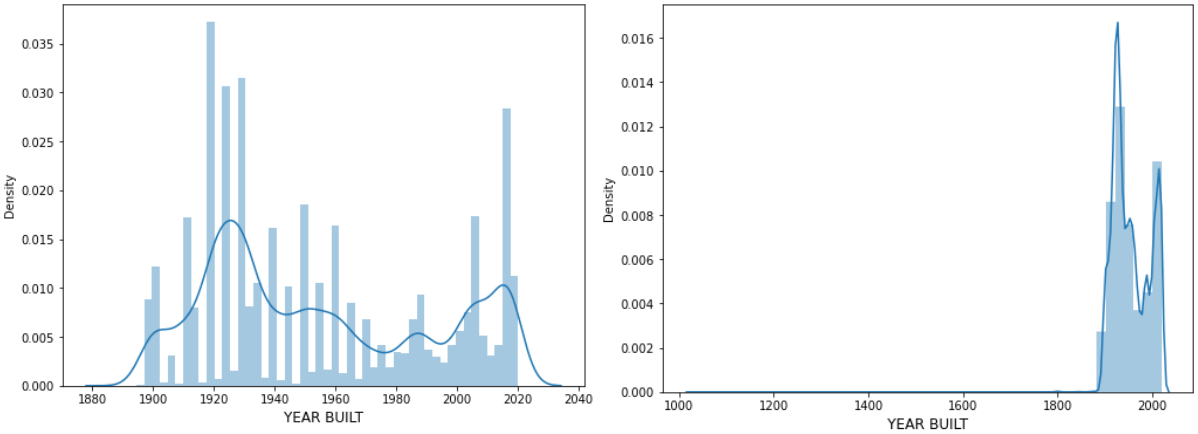
ניתן לראות כי הגרפים מוטים. תוצאות אלו התקבלו מכיוון שהיו לנו Outliers משמעותיים בנתונים, המידע מכיל "זנב" של מחירי עסקאות במחירים גבוהים מאוד, דבר זה מאפיל על שאר הנתונים.

לצורך הורדת ה-outliers - השתמשנו במטריקת z-score, שהשאירה אותנו עם כ 95% ממרכז כובד הנתונים. בנוסף הסרנו עסקאות שנערכו במחיר 0$ וכך נשארנו עם התפלגות המחיר הבאה:



דוגמא נוספת ל– Outlier detection & removal:

התפלגות שנת הבנייה לפני (ימין) ואחרי (שמאל) העיבוד - גם פה השתמשנו בz-score כשבחרנו להישאר עם הנתונים בטווח $mean \pm 1.96 \cdot std$ (מתרגם תחת הטרנספורמציה לנתונים בעלי z-score בטווח $(-1.96,1.96)$:



טיפול בערכים חסרים בכמויות גבוהות:

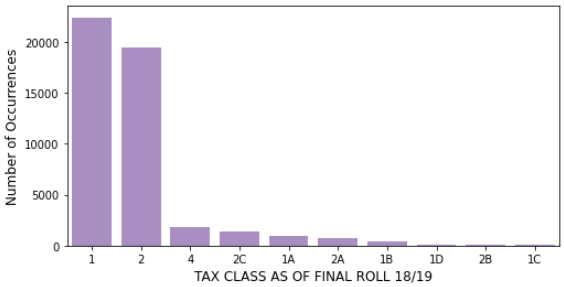
כפי שתיארנו בפרק הקודם, הפיצ'רים Easement ו Apartment number היו חסרים במעל 75% מהנתונים שלנו, כאשר Easement לא היה קיים בכלל ו Apartment number היה נוכח רק ב25% מהנתונים. בחרנו להוריד פיצ'רים אלו בעיקר כי לא היה להם ייצוג הולם.

Outliers מבניים:

כפי שתיארנו בפרק הקודם מצאנו כי חמשת הפיצ'רים שעוסקים בשטח ובמספר היחידות בנכס היו חסרים באופן מבני (קורלציה גבוהה בין פיצ'רים חסרים). ניסינו להשלימם בצורה סטטיסטית מתוך נתונים שנראו לנו רלוונטיים כמו zip-code, ושם השכונה, אך קיבלנו גיוון רחב מאוד הן בממוצע והן בסטיית התקן בין החסרים השונים.
החלטנו לוותר גם על שורות בעלות חסרים מבניים שכאלה.
חשוב להדגיש שהפרויקט עצמו השתמש בכמות נתונים מוגבלת מאוד (בגלל מגבלות התקציב ועלות ה- feature generation), לכן עצם הוויתור על נתונים במקרה של חוסרים פגע באופן מינורי לכל היותר במטרה אותה ניסינו להשיג.

טרנספורמציה של משתנים קטגוריילים

בחרנו להסב משתנים קטגוריילים למשתנים מספריים בעזרת Label encoding, כלומר לספק לכל קטגוריה מספר שלם שייצג אותה. בחרנו בשיטה זו מכיוון שהחלופה של one-hot encoding הייתה מובילה לגדילה משמעותית מידי בממד הפיצ'רים. הבחירה ב Label encoding, נעשתה לאחר בחינה של התפלגות הנתונים הקטגוריילים, לדוגמא:



3.4. בחינת מודלים ראשונית

חלק זה מקודד ומתואר במחברת הפרויקט: Modeling_02

לאחר עיבוד הנתונים ניגשנו לעבודת המידול.

3.4.1. המודלים הנבחרים

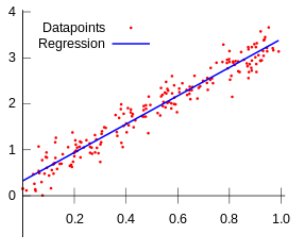
בחרנו לעבוד עם מודל פשוט, מודל מסוג ועדה ומודל למידה עמוקה לצורך גיוון.

כעת נציג את המודלים שבחרנו לעבוד איתם:

Linear Regression

רגרסיה ליניארית היא שיטה מתמטית למציאת הפרמטרים (מקדמים) של הקשר בין משתנה בלתי תלוי X למשתנה תלוי Y, **בהנחה שהקשר ביניהם הוא לינארי**, כלומר מהצורה $Y = \hat{a}X + \hat{b}$.

המודל מאפנן את המשקולו ת $\hat{a}$ ו$\hat{b}$ במטרה להביא למינימום את סכום השגיאות הריבועיות:



$$\min \sum_{i=1}^{n} \left(Y_i - \left(\hat{a}X_i + \hat{b}\right)\right)^2$$

Random Forest Regression

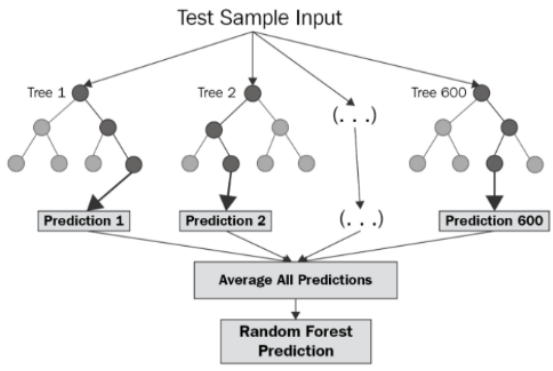
מסווג נוסף שבחרנו לעבוד איתו הוא מסווג מסוג ועדה, מסווג יער רנדומלי עבור רגרסיה.

ועדה זו בונה אוסף רגרסורים מסוג עצי רגרסיה אשר כל רגרסור בה מספק שיערוך והפרדיקט הסופי מתקבל ע"י ממוצע כלל השיערוכים של כל השותפים בועדה. יתרון של גישה זו היא שהיא שעל ידי שימוש ברגרסורים שונים ניתן להוריד את השפעת הרעש בצורה משמעותית ולקבל סיווג מדויק יותר. אלגוריתם היער הרנדומלי משלב בחירת תת קבוצות של דוגמאות ותת קבוצות של מאפיינים, באופן שבו כל עץ בועדה נבנה ממדגם של קבוצת הדוגמאות. בנוסף, בפיצול צומת פנימי בעץ נבחרת תת קבוצה של מאפיינים באופן רנדומלי, והתכונה שתיבחר לפיצול תיבחר מתוך קבוצה זו .

יער רדנומלי ידוע כאחד המודלים המדויקים שקיימים ולכן נבחר כאופציה למודל.

רשימת ההיפר-פרמטרים שנבחנו עבור מודל זה:

1. מספר המסווגים
2. עומק מרבי (עבור כל עץ רגרסיה)



Multi Layered Perceptron Regression

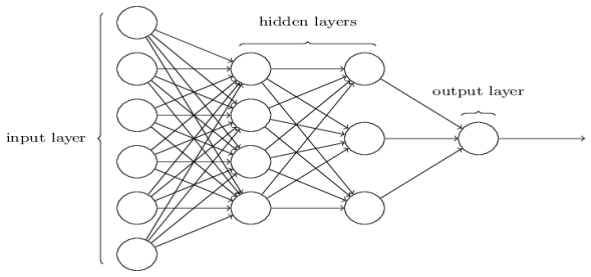
מסווג זה הוא מסווג מסוג למידה עמוקה.

בפרספטרון רב שכבתי כל נוירון בשכבה אחת מקושר לנוירונים בשכבה הבאה (ידוע גם בשם Fully-connected). כל נוירון מקבל ערך על ידי סכום כל הקישורים הנכנסים אליו, ובמקרים רבים הוא מפעיל גם פונקציית הפעלה (אקטיבציה) לא-ליניארית כדוגמת סיגמואיד או ReLU.
קיימות שיטות רבות לאימון של רשתות רב שכבתיות, הידועה והמוצלחת מביניהן היא מורד הגרדיאנט. בשיטה זו משווים את ערכי החיזוי של הרשת לתשובות הנכונות ומסיקים את הערך של פונקציית-שגיאה (עליהם נדבר בהמשך) מגדרת מראש. לאחר מכן, מזרימים את השגיאה לאחור ומעדכנים את המשקלים של הרשת בהתאם כדי להקטין את פונקציית השגיאה המחושבת במידה מסוימת. לאחר ביצוע של תהליך זה במשך מספר רב של פעמים באופן איטרטיבי, לרוב הרשת תתכנס לערך שגיאה נמוך מספיק, ברגע זה ניתן להגדיר שהרשת למדה. כיוון שרשתות נוירונים מקרבות פונקציות לא-ליניאריות, לא ניתן לפתור אותם באופן אנליטי ולכן משתמשים בשיטת האופטימיזציה של מורד הגרדיאנט. לשם כך מחשבים את הגרדיאנט של פונקציית השגיאה כפרמטר של המשקלים על הרשת, על ידי כלל השרשרת, ולאחר מכן כל משקל מוזז מעט לכיוון ההופכי לגרדיאנט, כלומר הקטנה מרבית של פונקציית השגיאה. זו היא גם הסיבה לכך שכל הרכיבים ברשתות עצביות - ובפרט פונקציית האקטיבציה - חייבים להיות גזירים .

בפרספטרון רב-שכבתי מסוג רגרסיה, השכבה האחרונה מחוברת לניורון פלט בודד בצורה לינארית ממושקלת, כך שכל ניורון בשכבה האחרונה תורם את חלקו לשיערוך הסופי.

רשימת ההיפר-פרמטרים שנבחנו עבור מודל זה:

1. ארכיטקטורת הרשת
2. פונקציית אקטיבציה
3. מספר האיטרציות המקסימלי



3.4.2. פונקציות השגיאה

בחרנו לעבוד עם 2 פונקציות שגיאה:

Mean squared log error

$$\mathrm{MSLE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (\log_e(1+y_i) - \log_e(1+\hat{y}_i))^2.$$

זוהי פונקציה שגיאה דומה לפונקציית השגיאה המוכרת MSE , רק שמכיוון שאנו מתעסקים במספרים גדולים בחרנו לעבוד עם המקבילה הלוגריתמית

אנחנו השתמשנו במינוס ה MSLE כדי לבסס יחס סדר שבו תוצאה גבוהה יותר, היא טובה יותר.

R^2 score

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

פונקציית שגיאה זו מספקת אומדן לאחוז השונות של המטרה (מחיר המכירה) שהוסבר ע״י המשתנים המסבירים (שאר הפיצ׳רים).

הסבר פרמטרים:

- $\hat{y}_i$ - שערוך הדוגמא ה- i
- y_i -המחיר האמיתי של הדוגמא ה- i
- $\bar{y} = \frac{1}{n}\Sigma y_i$

3.4.3. תוצאות

K – Fold cross validation :

השתמשנו בטכניקה זו ע״מ לקבל אומדן טוב יותר לשגיאות המסווגים ולהימנע מ-overfitting.

- Hyper Parameter Tuning

לכל מסווג בעל היפר פרמטרים הגדרנו היפר פרמטרים לכוונון ואת אוסף הערכים האפשרי לכל פרמטר. ביצענו מכפלה קרטזית כדי לקבל את כל אוסף כל אופציות ההיפר פרמטרים שאפשרנו והשווינו את תוצאות פונקציות השגיאות ($MSLE$ & R^2), כדי לבחור את אוסף ההיפר-פרמטרים הטוב ביותר.

זה נראה כך:

```
# Tuning hyper-parameters for neg_mean_squared_log_error

Best parameters set found on development set:

{'activation': 'relu', 'hidden_layer_sizes': (128,), 'max_iter': 10000}

Grid scores on development set:

-8.845 (+/-2.059) for {'activation': 'relu', 'hidden_layer_sizes': (128,), 'max_iter': 100}
-0.138 (+/-0.080) for {'activation': 'relu', 'hidden_layer_sizes': (128,), 'max_iter': 1000}
-0.094 (+/-0.106) for {'activation': 'relu', 'hidden_layer_sizes': (128,), 'max_iter': 10000}
-0.173 (+/-0.124) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256), 'max_iter': 100}
-0.168 (+/-0.118) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256), 'max_iter': 1000}
-0.155 (+/-0.114) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256), 'max_iter': 10000}
-0.169 (+/-0.123) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256, 512), 'max_iter': 100}
-0.176 (+/-0.131) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256, 512), 'max_iter': 1000}
-0.176 (+/-0.115) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256, 512), 'max_iter': 10000}

# Tuning hyper-parameters for r2

Best parameters set found on development set:

{'activation': 'relu', 'hidden_layer_sizes': (128,), 'max_iter': 10000}

Grid scores on development set:

-6.809 (+/-2.977) for {'activation': 'relu', 'hidden_layer_sizes': (128,), 'max_iter': 100}
-0.016 (+/-0.366) for {'activation': 'relu', 'hidden_layer_sizes': (128,), 'max_iter': 1000}
0.198 (+/-0.965) for {'activation': 'relu', 'hidden_layer_sizes': (128,), 'max_iter': 10000}
-0.166 (+/-0.582) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256), 'max_iter': 100}
-0.219 (+/-0.410) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256), 'max_iter': 1000}
-0.414 (+/-0.908) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256), 'max_iter': 10000}
-0.192 (+/-0.428) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256, 512), 'max_iter': 100}
-0.196 (+/-0.354) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256, 512), 'max_iter': 1000}
-0.182 (+/-0.453) for {'activation': 'relu', 'hidden_layer_sizes': (128, 256, 512), 'max_iter': 10000}
```

בנוסף, הגדרנו 2 מסווגים "טיפשים" שמסווגים קבוע שהוא הממוצע\ חציון ע"מ לייצר בסיס השוואתי לאלגוריתמים האחרים.

להלן התוצאות:

Model	**Neg MSLE**	R^2
Dummy Mean Reg	-0.15	-0.12
Dummy Median Reg	-0.15	-0.13
Linear Reg	-0.1	0.28
Random Forest Reg	-0.08	0.35
MLP Reg	-0.09	0.22

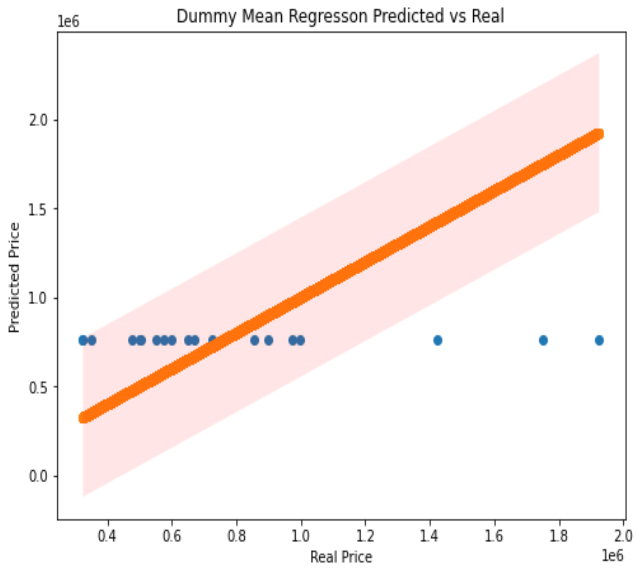
היפר פרמטרים נבחרים:

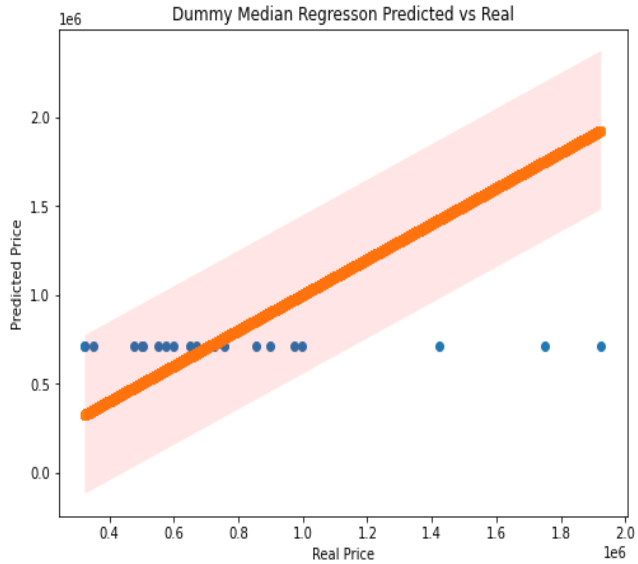
Model	**Max_depth**	$n\_estimators$	**activation**	**architecture**
Random Forest Reg	100	50	-	-
MLP Reg	-	-	relu	(128,)

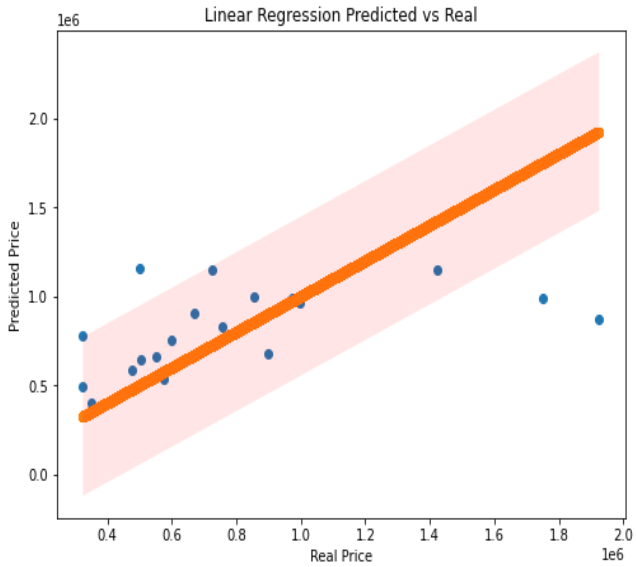
ניתן לראות כי האלגוריתם RandomForest הציג את התוצאה הטובה ביותר.

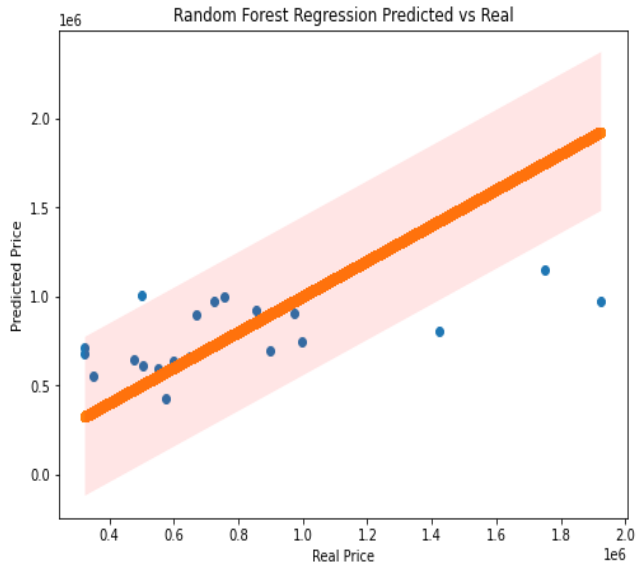
בשלב זה הגדרנו צורת הערכה וויזואלית לאלגוריתמים שלנו, אנו נציג את השיערוכים על ציר אחד, ואת המחיר האמיתי בציר השני, כלומר הישר $x = y$ מייצג את המודל המושלם. נצבע את סטיית התקן האמיתית (של המחיר האמיתי!) מסביב למודל האולטימטיבי וכן נבחן עבור אילו נקודות שגיאת השיערוך הייתה "סבירה" ועבור אילו נקודות הייתה "רעה".

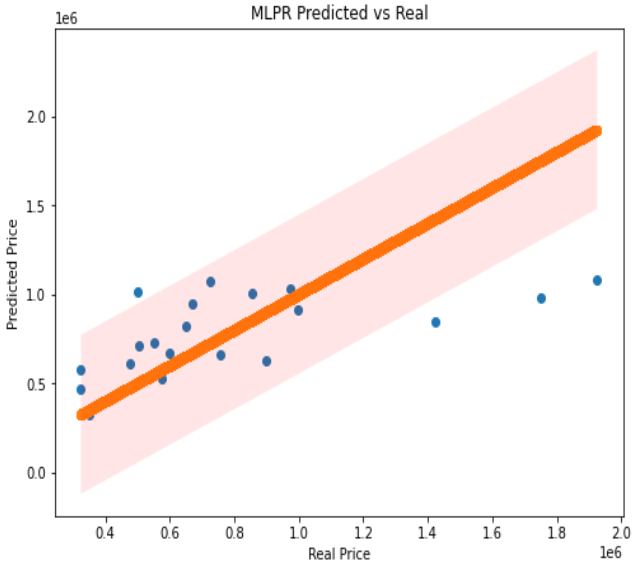
הגרפים מוצגים בעמוד הבא:


Dummy Mean Regresson Predicted vs Real
1e6
Predicted Price
Real Price
1e6


Dummy Median Regresson Predicted vs Real
1e6
Predicted Price
Real Price
1e6


Linear Regression Predicted vs Real
1e6
Predicted Price
Real Price
1e6


Random Forest Regression Predicted vs Real
1e6
Predicted Price
Real Price
1e6


MLPR Predicted vs Real
1e6
Predicted Price
Real Price
1e6

3.5. יצירת פיצ'רים - קווי אורך ורוחב

חלק זה מקודד ומתואר במחברת הפרויקט: 03_Feature_Generating_Lat_Lng.ipynb

לאחר שיערוך ראשוני של המודלים ניגשנו לחלק העיקרי של הפרויקט – יצירה של פיצ'רים חדשים. בשלב בראשון רצינו לדעת את קווי האורך והרוחב של הנקודות. לצורך השגת המטרה פנינו ל- Google Geocoding.
Google Geocoding הינו שירות (בתשלום) שניתן ע"י גוגל שמטרתו הפיכת כתובת לקווי אורך ורוחב.
התחלנו לחקור על הכלי ועל אופן השימוש בו, כתבנו קוד שמחלץ את המידע עבור שורה בטבלה ושמרנו אותה לצורך המשך העבודה.

3.6. בחינת מודלים שנייה

חלק זה מקודד ומתואר במחברת הפרויקט: Modeling_With_Lat_Lng_04 .

לאחר בניית הפיצ'רים החדשים של קווי האורך והגובה ניגשנו בחזרה לעבודת המידול, ע"מ לבדוק את השפעת הפיצ'רים החדשים על יכולות החיזוי.

תוספות בחלק זה לעומת המידול הראשוני

1. בחלק זה הוספנו מודל נוסף – KNN Regression . במודל זה בחרנו להשתמש **אך ורק** בפיצ'רים החדשים (קווי אורך ורוחב). המחשבה שעמדה מאחורי הוספת המודל הנ"ל הינה תהליך בחירת נכס מסוים בחיים האמיתיים. כלומר, מניסיוננו בשוק הנדל"ן אנו בעיקר בודקים את מחירי הנכסים בקרבת הנכס המבוקש כדי לשערך את מחירו. רצינו לבדוק את רמת הדיוק והאמינות של מודל זה, אותנו כולנו מיישמים הלכה למעשה בחיי היום-יום.

2. הוספנו שיטת הערכה ויזואלית לחיזוי. כעת משהשגנו את המיקום המדויק של הנכסים, יכולנו לצייר אותם על מפה. הוספנו ציורים שמראים את נקודות המבחן. נקודות כחולות הן נקודות בהן השגיאה קטנה מסטיית התקן, נקודות אדומות הן נקודות בהן השגיאה גדולה מסטיית התקן.

להלן התוצאות:

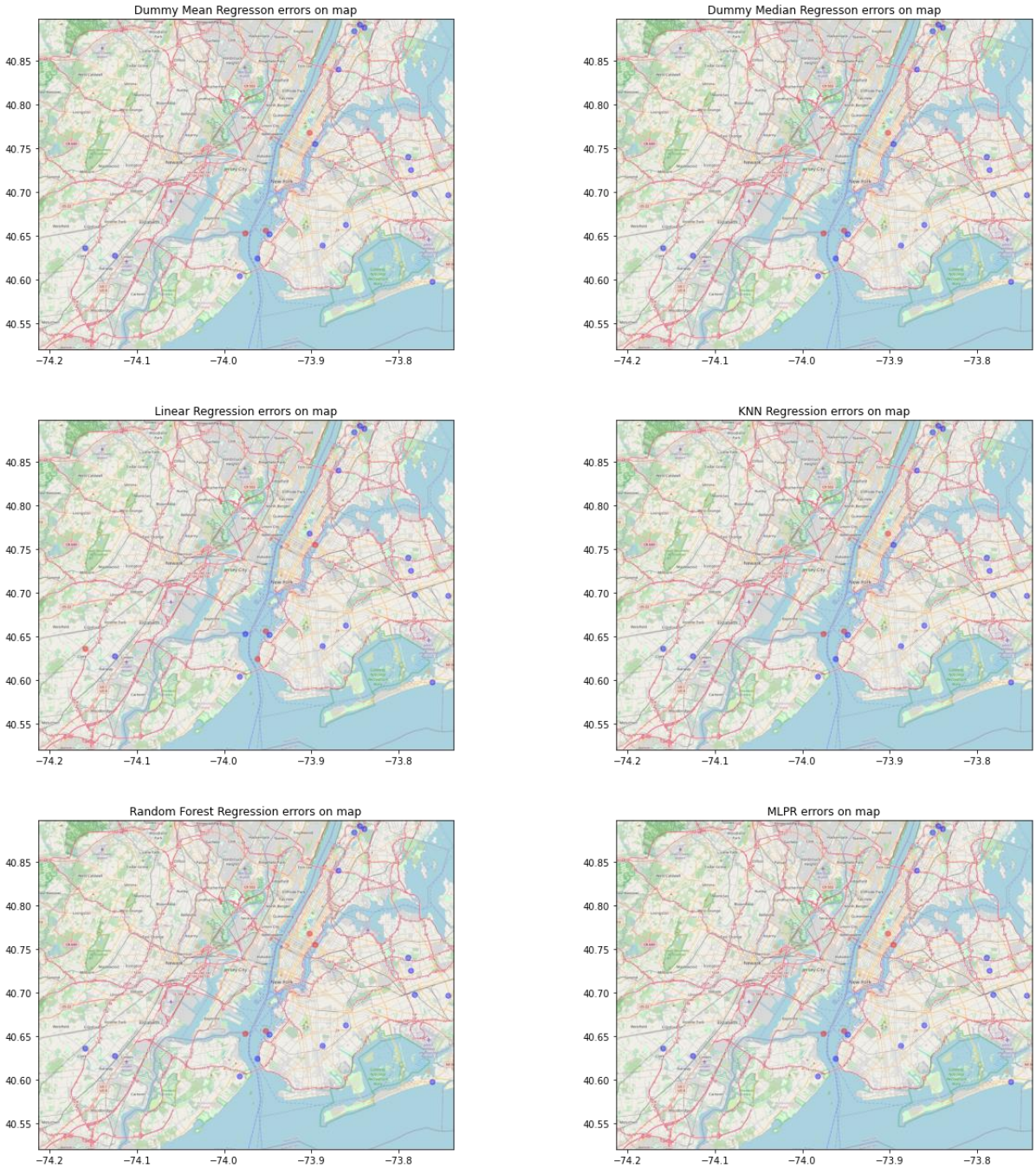
	With Old Features		With New Features	
Model	**Neg MSLE**	R^2	**Neg MSLE**	R^2
Dummy Mean Reg	-0.15	-0.12	-0.15	-0.12
Dummy Median Reg	-0.15	-0.13	-0.15	-0.13
KNN Reg	-	-	-0.1	0.24
Linear Reg	-0.1	0.28	-0.08	0.44
Random Forest Reg	-0.08	0.35	-0.07	0.411
MLP Reg	-0.09	0.22	-0.1	0.22

היפר פרמטרים נבחרים:

Model	**n_neighbors**	**metric**	**Max_depth**	***n_estimators***	**activation**	**architecture**
KNN	5	Manhattan	-	-	-	-
Random Forest Reg	-	-	100	100	-	-
MLP Reg	-	-	-	-	ReLu	(128,)

ניתן לראות שברוב המקרים חל שיפור לטובה בתוצאות מה שמעיד על כך שהוספת הפיצ'רים **תרמה** ליכולות החיזוי. בנוסף ניתן לראות שהשיטה המוצעת בעזרת KNN-Reg הניבה תוצאות מספקות, יחסית לחיזוי בעזרת שני פיצ'רים בודדים, מה שמעיד על כך ששיטה זו "פשוטה אבל עובדת".

כך זה נראה על המפה:



3.7. יצירת פיצ'רים - מאפיינים גאוגרפיים נוספים

כעת ניגשנו ליצור עוד פיצ'רים מעניינים הקשורים למיקום הגאוגרפי של הנכס שעשויים להשפיע על יכולות החיזוי. לצורך כך השתמשנו ב Google Places API.

יצרנו קרוב ל15 פיצ'רים שונים, הקוד והתיעוד שלהם נמצא במחברות notebook שנמצאות ב-GitHub.

<u>דוגמאות למחברות כאלה:</u>

notebook לקבלת מספר בתי הקפה ברדיוס 1000 מטר: realEstateProject_get_cofe_7

notebook לקבלת סוכנויות הנדל"ן ברדיוס 1000 מטר: realEstateProject_get_real_estate_9

notebook לקבלת מספר הקניונים ברדיוס 3000 מטר: realEstateProject_get_shopping_mall_11

notebook לקבלת מספר בתי הכנסת ברדיוס 3000 מטר: realEstateProject_get_synagogue_13

שירות זה של גוגל מאפשר קבלת מאפיינים גאוגרפים, לפי סוג חיפוש ע"י מתן קווי אורך ורוחב ורדיוס מהנקודה. ניתן להשיג מידע על סוגים שונים של מקומות כמו – מוסדות בנקאות, עו"ד, חנויות אלכוהול, קזינו ועוד רבים אחרים. את הרשימה המלאה על סוגי המקומות הניתנים לחיפוש ניתן למצוא בקישור הבא.

המאפיינים אותם בחרנו לחקור היו: בתי קפה, סוכנויות נסיעות, סוכנויות נדל"ן, אטרקציות תיירותיות, בתי כנסת וקניונים.

3.8. בחינת מודלים סופית

חלק זה מקודד ומתואר במחברת הפרויקט: Modeling_With_New_Features.ipynb_05

בחלק זה לקחנו את כל הפיצ'רים החדשים שייצרנו:

- כמות בתי הקפה ברדיוס 1 ק"מ
- האם קיימת סוכנות נסיעות ברדיוס 2 ק"מ
- מספר האטרקציות התיירותיות ברדיוס 1 ק"מ
- כמות בתי הכנסת בקרבת הנכס ברדיוס 3 ק"מ
- האם קיימת סוכנות נדל"ן ברדיוס 1 ק"מ
- מספר הקניונים ברדיוס 3 ק"מ מהנכס

צירפנו אותם לבסיס הנתונים והרצנו שוב את האלגוריתמים שלנו.

להלן התוצאות:

	With Old Features		With Lat,Lng		With Lat,Lng + Features	
Model	**Neg MSLE**	**R^2**	**Neg MSLE**	**R^2**	**Neg MSLE**	**R^2**
Dummy Mean Reg	-0.15	-0.12	-0.15	-0.12	-0.15	-0.12
Dummy Median Reg	-0.15	-0.13	-0.15	-0.13	-0.15	-0.13
KNN Reg	-	-	- 0.1	0.24	-	-
Linear Reg	-0.1	0.28	-0.08	0.44	-0.08	0.45
Random Forest Reg	-0.08	0.35	-0.07	0.411	-0.09	0.31
MLP Reg	-0.09	0.22	-0.1	0.22	-0.08	0.355

היפר פרמטרים נבחרים:

Model	n_neighbors	metric	Max_depth	$n\_estimators$	activation	architecture
KNN	5	Manhattan	-	-	-	-
Random Forest Reg	-	-	10	50	-	-
MLP Reg	-	-	-	-	ReLu	(128,)

*את הגרפים של המחיר נגד המחיר החזוי ואת המפות ניתן למצוא במחברת

רגרסיה לינארית - ניתן לראות כי חל שיפור קטן מאוד בתוצאות עבור רגרסיה לינארית.
יער רנדומלי – הוספת הפיצ'רים הפכה את המודל לפחות מוצלח.
פרספטרון רב שכבתי –חל שיפור משמעותי בתוצאות המודל.

4. סיכום, ניתוח תוצאות ומסקנות

לאחר ביצוע כל שלבי הפרויקט ניגשנו לבחון את התוצאות.

Mean\Median Regression:
ראוי לציין כי החזאים ה"טיפשים" של הממוצע\חציון מציגים תוצאות יפות מאוד. ניתן ללמוד מכך שלמרות סטיית התקן הגדולה עקב ההתעסקות במספרים גדולים (מבחינת סדר הגודל), המחיר הממוצע\חציון מספק שיערוך לא רע למחירי הדירות בעיר.

KNN:
מודל הKNN שלוקח בחשבון רק את מיקומי הנכסים הציג תוצאות יפות מאוד יחסית לפשטות המודל. מכך ניתן ללמוד כי שיטת השיערוך, המתבססת על המיקום היחסי של הנכס היא בעלת בסיס איתן, ועכשיו גם מבוסס.

מבחינת הפיצ'רים הנוספים שייצרנו למערכת:
הקפיצה הגדולה בתוצאות הגיעה לאחר שילוב האלמנט הגאוגרפי, קווי האורך והרוחב במערכת. בהמשך לפסקה הקודמת, ניתן לומר שכצפוי, חלק משמעותי ממחיר הנכס נקבע ע"י מיקומו ביחס לנכסים אחרים.
שילוב הפיצ'רים הנוספים שבוצע בעזרת Google Places אכן הציג שיפור בתוצאות רוב המודלים אולם, שיפור זה היה מינורי ביחס לשיפור שהושג ע"י שימוש בקווי הגובה\אורך.
קיימים שני הסברים אפשריים לתוצאה זו, האחד- הפיצ'רים שבחרנו להוסיף בחלקם השפיעו לטובה ובחלקם לרעה, כך שבסה"כ היה שיפור יחסית זניח. השני- השפעת הפיצ'רים הללו באה לידי ביטוי באמצעות השימוש בקווי רוחב\אורך, ומעצם השימוש במיקום הגאוגרפי, מתגלה רוב המידע החבוי על הנכס.

המודלים:
לאחר התבוננות בתוצאות, אפילו בפרויקט בסקאלה קטנה כמו שלנו, ניתן לראות כי שימוש במודל מורכב יותר מניב תוצאות טובות יותר.
מבחינתנו ניתן להעריך כי "עוצמת המודלים" משולה ליחס הסדר הבא:

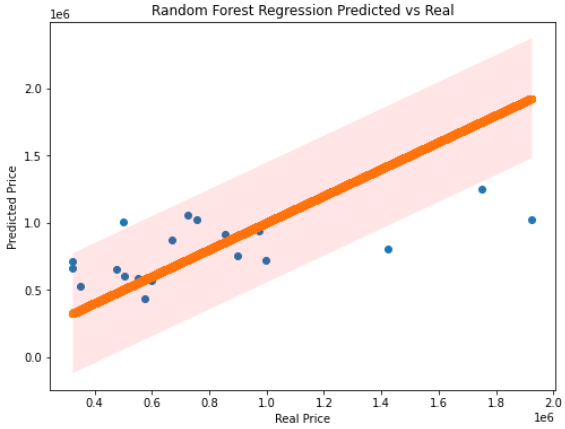
$$Mean\backslash Median\ Reg < Linear\ Reg < KNN(with\ lat\backslash lng) < MLP\ Reg < RF\ Reg$$

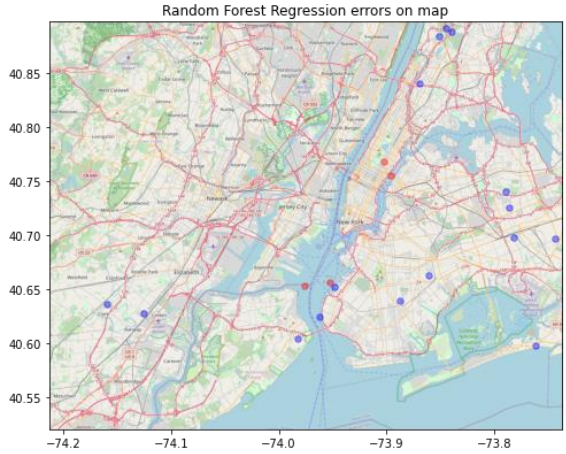
<u>סיכום:</u>

אם כך, המודל הנבחר הינו Random Forest Regression, שמשתמש אך ורק בקווי האורך\גובה ולא במאפיינים שנוספו בעזרת Google Places (בתי כנסת\קניונים וכו'). בחרנו במודל זה מכיוון שהציג את התוצאות הטובות והיציבות ביותר עבור 2 סוגי פונקציות השגיאה איתם בחרנו לעבוד, גם בשלב הוולידציה ($K - Fold\ Cross\ Validation$) וגם בשלב המבחן הסופי.

תזכורת לביצועי המודל:

```
Random Forest Regression Negative Mean Squarred Log Error is:-0.07687994249503438
Random Forest Regression R2 Score is:0.4119335485354832
Result is calculated using 8-cross-validation
```





5. עבודה עתידית

רעיונות לעבודות המשך שעלו במהלך הפרויקט:

1. **הרחבת הפרויקט בעזרת אמצעיים תקציביים**

הפרויקט שלנו מניח את עבודת השטח למחקר מעמיק בנושא חיזוי מחירי הדירות בניו-יורק. סקאלת הפרויקט הייתה מוגבלת מאוד בגלל אמצעיים תקציביים שעמדו לרשותנו: היינו מוגבלים לתקציב החינמי שגוגל מספקת- לא יכולנו לג'נרט פיצ'רים עבור כל הדאטא (מה שגרם לכך שבפועל השתמשנו רק בחלק מזערי מהנתונים שעיריית ניו-יורק מאפשרת להפיק). בפרויקט ניתן למצוא את **רוב אם לא כל הקוד הדרוש, בצורה נגישה ומתועדת**, ע"מ להרחיב את המחקר בנושא, הן מבחינת היקף הדאטא והן מבחינת המאפיינים (פי'צרים) הנוספים שניתן להוסיף לה.

הצעתנו המתבקשת היא לקחת את הפרויקט שלנו ולהרחיבו כך שיכיל את כל הנתונים, החל משנת 2003 ולבדוק את השינוי המתקבל בתוצאות. כמו-כן ניתן לחקור פי'צרים נוספים על בסיס הפיצ'רים אותם בנינו כדי לשפר את יכולות החיזוי של המודלים.

2. **יישום הפרויקט על ערים נוספות:**

סביר להניח כי למאגר אליו נחשפנו עבור העיר ניו-יורק קיימים מקבילים עבור ערים נוספות. פרויקט המשך מעניין ייקח את הקוד שלנו וימצא מאגרי נתונים נוספים שניתן להתבסס עליהם עבור ערים נוספות ויערוך השוואה בין תוצאות המודלים עבור ערים שונות. היישום הינו מיידי, נוכל להריץ את הקוד שלנו גם על מאגרי מידע שונים.

3. **שילוב כוחות עם סוכני נדלן:**

העבודה שעשינו לקחה בחשבון ידע קודם שיש לנו על שוק הנדלן. פרויקט המשך מעניין ישלב סוכני נדלן מהעיר הנחקרת. שימוש באנשי מקצוע שמכירים את הווי העיר והשכונות יכול להביא למחקר ממוקד יותר ולהתמקדות בפיצ'רים חדשים אותם אנשי המקצוע תופסים כרלוונטיים ובכך למנף את תוצאות הפרויקט באמצעות ידע מקדים.

6. ביבליוגרפיה

Database - https://www1.nyc.gov/site/finance/taxes/property-annualized-sales-update.page

Google Geocoding API - https://developers.google.com/maps/documentation/geocoding/overview

Google Places API - https://developers.google.com/places/web-service/overview

Maps Database - https://www.openstreetmap.org

Random Forest Regressor paper - Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." *R news* 2.3 (2002): 18-22.