**מסמך מסכם לפרויקט הערכת שווי נדל"ן**

**תיאור המאגר**

המאגר בו השתמשנו הוא קובץ, Housing.csv המכיל נתונים על נכסי נדל״ן, הנתונים כוללים מאפיינים כמו שטח, מספר חדרי שינה, מספר חדרי אמבטיה, מספר קומות, חניה, ומאפיינים נוספים כמו מיזוג אוויר, חימום מים חמים, חדר אורחים ומרתף. המטרה הייתה להשתמש במאפיינים אלו כדי להעריך את שווי הנכס.

**שאלות המחקר**

* מהם המאפיינים המשפיעים ביותר על שווי הנכס?
* כיצד ניתן לשפר את דיוק המודלים להערכת שווי הנדל"ן?
* האם ישנם מאפיינים שניתן להוסיף או לשנות כדי לשפר את התחזיות?

**שיטת העבודה על הפרויקט**

הפרויקט כלל מספר שלבים עיקריים:

1. **עיבוד נתונים**: כלל הסרת ערכים חריגים, יצירת פיצ'רים חדשים, וטרנספורמציות לוגריתמיות.
2. **נרמול וקטגוריזציה**: שימוש ב-StandardScaler ו-OneHotEncoder לנרמול וקטגוריזציה של הנתונים.
3. **בחירת תכונות**: שימוש ב-SelectFromModel עם Lasso לבחירת התכונות המשמעותיות ביותר.
4. **הרצת מודלים**: הרצת מספר מודלים כמו רגרסיה ליניארית, עץ החלטה, SVR ו-KNN.
5. **השוואת תוצאות**: ניתוח ביצועי המודלים והשוואת התוצאות.

**סיכום האלגוריתמים**

**רגרסיה ליניארית**

* **תיאור תיאורטי**: רגרסיה ליניארית היא שיטה סטטיסטית לחיזוי ערך של משתנה תלוי על בסיס משתנים בלתי תלויים. היא מניחה קשר ליניארי בין המשתנים.
* **יישום בפרויקט**: השתמשנו ב-LassoCV, RidgeCV, ו-ElasticNetCV  כדי למצוא את המודל הטוב ביותר. כל אחד מהמודלים הללו מוסיף רגולריזציה כדי למנוע התאמת יתר (overfitting).

**עץ החלטה**

* **תיאור תיאורטי**: עץ החלטה הוא מודל חיזוי המשתמש במבנה עץ כדי לקבל החלטות על סמך ערכי התכונות. הוא מחלק את הנתונים לצמתים על בסיס ערכי תכונות.
* **יישום בפרויקט**: השתמשנו ב-GridSearchCV כדי למצוא את הפרמטרים האופטימליים, כמו עומק העץ ומספר הדוגמאות המינימלי לפיצול.

**SVR (תמיכה וקטורית רגרסיה)**

* **תיאור תיאורטי**: SVR הוא מודל חיזוי המשתמש במשטח היפר-מישורי כדי לחזות ערכים רציפים. הוא מנסה למזער את השגיאה בתוך מרווח מסוים.
* **יישום בפרויקט**: השתמשנו במודל בסיסי עם פרמטרים קבועים, כמו kernel='linear'  ו-C=10.0.

**KNN (הקירוב הקרוב ביותר)**

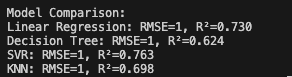
* **תיאור תיאורטי**: KNN הוא מודל חיזוי המשתמש במרחקים בין דוגמאות כדי לחזות ערכים. הוא מבוסס על ההנחה שדוגמאות קרובות יהיו בעלות ערכים דומים.
* **יישום בפרויקט**: השתמשנו ב-GridSearchCV כדי למצוא את הפרמטרים האופטימליים, כמו מספר השכנים והמשקלות.

**אתגרים ופתרונות**

* **נרמול נתונים**: נתקלנו בבעיות עם ערכים חריגים, ולכן השתמשנו ב-RobustScaler כדי להתמודד עם ערכים אלו.
* **בחירת תכונות**: היה צורך לבחור תכונות משמעותיות כדי למנוע התאמת יתר. השתמשנו ב-Lasso לבחירת תכונות.
* **זמן ריצה**: חיפוש פרמטרים אופטימליים היה ארוך, ולכן השתמשנו ב-GridSearchCV עם n\_jobs=-1 כדי להאיץ את התהליך.

**אנליזת התוצאות**

* **רגרסיה ליניארית**: נתנה תוצאות טובות עם RMSE נמוך ו-R² גבוה, כנראה בגלל היכולת שלה להתמודד עם קשרים ליניאריים.
* **עץ החלטה**: היה רגיש לפרמטרים ונתן תוצאות פחות טובות, אך היה קל לפרש את המודל.
* **: SVR**נתן תוצאות טובות, אך זמן הריצה היה ארוך יותר.
* **:KNN**היה רגיש לפרמטרים, אך נתן תוצאות טובות כאשר הפרמטרים היו מותאמים.

****

**מסקנות**

הפרויקט הראה את החשיבות של עיבוד נתונים נכון ובחירת תכונות משמעותיות. רגרסיה ליניארית ו-SVR היו המודלים המוצלחים ביותר, כנראה בגלל היכולת שלהם להתמודד עם נתונים מורכבים. עץ החלטה ו-KNN היו רגישים יותר לפרמטרים ולמבנה הנתונים.

קישור לגיט:  
https://github.com/hagayknoro/Machine\_Learning\_House\_Pricing\_Final\_Project