שינוי אלגוריתם Gradient Boosting Regressor

1. על פי התיעוד הרשמי של מימוש ה- Gradient Boosting Regressor קיימים כמה הייפר-פרמטרים שלדעתינו אפשר לשנות:

a. ***n_estimators*:** מספר שלבי ה-Boosting שהאלגוריתם מבצע, כלומר מספר העצים באנסמבל. אפשר לנסות להגדיל פרמטר זה ולראות האם קיימים שיפור בדיוק החיזוי. **לדעתנו, יש להגדיל את מספר העצים כדי להגדיל את ה- Robustness לתופעת ה- Overfitting.**

b. ***max_depth*:** עומק המקסימלי של העצים באנסמבל, שהוא חסם על מספר הקודקודים בעץ (יכול להגיע גם למצב של Overfitting). **לדעתנו, יש ליצור עצים קטנים יותר שבקודקדים שלהם יותר רשומות לבניית מודל ה-SVR, ולכן נקטין פרמטר זה.**

c. ***min_samples_leaf*:** מספר הרשומות המינימלי הנדרש בעלי העץ. קובע מתי מסיימים את גידול העץ ויכול להשפיע על דיוק החיזוי. **לדעתנו, יש להשאיר מספיר רשומות בשביל שמודל ה-SVR יהיה קביל ולכן נגדיל פרמטר זה.**

d. ***loss*:** פונקציית חישוב הטעות (בין ערך האמיתי של מתשנה המטרה לבין הערך החזוי). קיימים מספר ערכים אפשריים loss : {'ls', 'lad', 'huber', 'quantile'}. **נבדוק באמצעות כמה ניסויים את השפעת פרמטר זה על הביצועים.**

2. השינוי שביצענו הוא להשתמש במחלקה GradientBoostingRegressor של Scikit-learn ולרשת ממנה בכדי ליצור את המחלקה החדשה SVRGradientBoostingRegressor, כלומר:

```
Class SVRGradientBoostingRegressor(GradientBoostingRegressor):
```

המחלקה SVRGradientBoostingRegressor יוצרת בבנאי שלה instance של GradientBoostingRegressor ומאתחלת משנה בשם model ששם נישמור את מודל ה-SVR המאומן.

מימשנו את שתי המתודות:

א. `fit (self, X, y, sample_weight=None, monitor=None)`: יוצרת instance של מודל SVR ומשתמשת בו כדי לאמן GradientBoostingRegressor, כאשר בעלים יתבצע מיצוע לפי SVR.

ב. `predict (self, X)`**:** מבצעת חיזוי של ערך המטרה של רשומות חדשות, אחרי שקיים מודל רגרסיה מאומן.

<u>הערות:</u>

- הקובץ `Program.py` עוטף את המחלקה SVRGradientBoostingRegressor החדשה, וטוען את קבצי האימון המצויים בתיקייה `Data`.

- בכדי להריץ את הקוד, יש לקרוא לקובץ Program.py ולדאוג שקבצי האימון יהיו בתיקייה Data, בנתיב שבו מריצים את הקובץ.
- הקוד יבצע ריצות עבור כל הפרמטרים שמפורטים בעמודים הבאים בצורה סדרתית.
- <u>פלט לדוגמא</u>:

```
------------------------
Scikit regressor on Data\airfoil_self_noise.csv: 0.696618159546 sec
Modified regressor on Data\airfoil_self_noise.csv: 2.68984368857 sec
mse     -3.072
mse    -46.522
------------------------
```

3. ביצענו כמה ניסויים על 5 קבצי אימון המותאמים לבעיית Reggression:

Dataset	**Number of Instances**	**Number of attributes (last is prediction attribute)**
Airfoil self-noise	1503	6
Appliances energy prediction	10355	27
Cycle Power Plant	9568	5
Glass	214	10
Concrete	1030	9

- בחלק מקבצי האימון היו ערכים חסרים, אותם השלמנו ע״י מתן ערך הממוצע של אותה תכונה.
- תוך שימוש ב-**<u>10-fold Cross Validation</u>**, השווינו את מדד ה-MSE על מימוש ה-GBReggressor המקורי ועל שינוי המימוש SVR-GBReggressor, עבור הייפר-הפרמטרים השונים. התוצאות מוצגות בטבלאות להלן.

ליעד נחמיאס Gradient Boosting Regressor – 2 מטלה

מיכאל חבקין יישום אלגוריתמים לומדים 2017-2018

Tuning parameter: **loss**

loss	n_estimators	min_samples_leaf	max_depth
LS	**500**	**3**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
Data Set	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	2.753	0.293	46.88	1.48
Appliances_energy_prediction	5766.435	10.013	12942.001	66.865
Cycle_Power_Plant	10.23	1.911	279.133	51.603
Glass	1.146	0.139	5.212	0.195
Concrete	20.583	0.375	273.038	0.936

loss	n_estimators	min_samples_leaf	max_depth
LAD	**500**	**3**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
Data Set	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	3.682	0.945	46.129	2.3
Appliances_energy_prediction	8797.726	10.742	12919.791	69.166
Cycle_Power_Plant	12.289	2.759	252.97	54.091
Glass	1.14	0.487	5.094	0.597
Concrete	23.116	0.87	271.506	1.497

loss	n_estimators	min_samples_leaf	max_depth
HUBER	**500**	**3**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
Data Set	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	3.288	0.1422	46.331	2.825
Appliances_energy_prediction	6635.752	12.977	12931.212	72.986
Cycle_Power_Plant	9.918	3.817	279.836	53.56
Glass	1.151	0.68	5.064	0.736
Concrete	21.264	1.225	271.828	2.081

loss	n_estimators	min_samples_leaf	max_depth

ליעד נחמיאס Gradient Boosting Regressor – 2 מטלה

מיכאל חבקין יישום אלגוריתמים לומדים 2017-2018

QUANTILE	**500**	**3**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
Data Set	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	10.517	0.842	45.136	2.122
Appliances_energy_prediction	10906.94	10.842	12854.754	68.395
Cycle_Power_Plant	33.154	2.7334	249.475	58.49
Glass	21.795	0.101	5.516	0.127
Concrete	45.532	0.963	271.269	1.587

ליעד נחמיאס Gradient Boosting Regressor – 2 מטלה

מיכאל חבקין יישום אלגוריתמים לומדים 2017-2018

Tuning parameter: **n_estimators**

loss	n_estimators	min_samples_leaf	max_depth
LS	**100**	**3**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
Data Set	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	5.06	0.069	41.424	0.383
Appliances_energy_prediction	7416.804	2.14	12778.119	15.945
Cycle_Power_Plant	13.41	0.45	205.101	12.92
Glass	1.175	0.029	3.582	0.037
Concrete	26.813	0.089	265.747	0.23

loss	n_estimators	min_samples_leaf	max_depth
LS	**200**	**3**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
Data Set	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	3.805	0.123	45.089	0.644
Appliances_energy_prediction	6614.12	4.235	12895.664	28.852
Cycle_Power_Plant	12.138	0.789	263.552	21.928
Glass	1.71	0.557	4.645	0.069
Concrete	23.265	0.149	270.768	0.39

loss	n_estimators	min_samples_leaf	max_depth
LS	**500**	**3**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
Data Set	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	2.755	0.294	46.881	1.599
Appliances_energy_prediction	5780.989	9.629	12942.529	67.471
Cycle_Power_Plant	10.227	1.779	279.135	50.684
Glass	1.192	0.137	5.227	0.167
Concrete	20.553	0.352	273.024	0.911

loss	n_estimators	min_samples_leaf	max_depth

ליעד נחמיאס Gradient Boosting Regressor – 2 מטלה

מיכאל חבקין יישום אלגוריתמים לומדים 2017-2018

LS	**1000**	**3**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
Data Set	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	2.231	0.598	47.206	2.928
Appliances_energy_prediction	5334.021	19.494	12955.387	128.695
Cycle_Power_Plant	9.186	6.376	290.435	118.165
Glass	1.145	0.597	5.201	0.711
Concrete	1.455	19.864	273.496	3.003

ליעד נחמיאס Gradient Boosting Regressor – 2 מטלה

מיכאל חבקין יישום אלגוריתמים לומדים 2017-2018

Tuning parameter: **min_samples_leaf**

loss	n_estimators	min_samples_leaf	max_depth
LS	**500**	**3**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
DataSet Name	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	2.753	0.293	46.88	1.48
Appliances_energy_prediction	5766.435	10.013	12942.001	66.865
Cycle_Power_Plant	10.23	1.911	279.133	51.603
Glass	1.146	0.139	5.212	0.195
Concrete	20.583	0.375	273.0.38	0.936

loss	n_estimators	min_samples_leaf	max_depth
LS	**500**	**5**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
DataSet Name	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	2.937	0.32	46.622	1.642
Appliances_energy_prediction	5758.194	13.122	12941.647	91.352
Cycle_Power_Plant	10.43	3.188	273.526	63.885
Glass	1.111	0.208	4.77	0.249
Concrete	20.018	0.544	272.747	1.267

loss	n_estimators	min_samples_leaf	max_depth
LS	**500**	**10**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
DataSet Name	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	3.438	0.407	44.926	1.884
Appliances_energy_prediction	6020.301	19.26	12934.989	95.29
Cycle_Power_Plant	10.842	2.936	262.612	60.947
Glass	1.08	0.166	4.04	0.209
Concrete	19.925	0.515	42.573	1.284

loss	n_estimators	min_samples_leaf	max_depth

ליעד נחמיאס Gradient Boosting Regressor – 2 מטלה

מיכאל חבקין יישום אלגוריתמים לומדים 2017-2018

LS	**500**	**20**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
DataSet Name	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	3.466	0.444	44.598	2.028
Appliances_energy_prediction	6223.665	20.986	12933.368	93.752
Cycle_Power_Plant	10.766	3.128	273.823	61.345
Glass	0.992	0.115	3.388	0.189
Concrete	21.16	0.343	270.15	0.992

ליעד נחמיאס Gradient Boosting Regressor – 2 מטלה

מיכאל חבקין יישום אלגוריתמים לומדים 2017-2018

Tuning parameter: **max_depth**

loss	n_estimators	min_samples_leaf	max_depth
LS	**500**	**10**	**3**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
DataSet Name	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	4.866	0.339	18.15	1.883
Appliances_energy_prediction	7252.307	10.466	12447.037	97.455
Cycle_Power_Plant	13.147	1.775	43	59.913
Glass	1.182	0.149	1.708	0.19
Concrete	23.527	0.366	230.91	1.074

loss	n_estimators	min_samples_leaf	max_depth
LS	**500**	**10**	**4**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
DataSet Name	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	3.968	0.351	40.407	1.758
Appliances_energy_prediction	971.281	16.084	1807.235	102.379
Cycle_Power_Plant	11.901	2.3511	191.371	60.964
Glass	1.091	0.146	2.999	0.195
Concrete	20.67	0.368	267.748	1.026

loss	n_estimators	min_samples_leaf	max_depth
LS	**500**	**10**	**5**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
DataSet Name	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	3.437	0.463	44.928	2.089
Appliances_energy_prediction	6018.894	22.453	12934.993	108.857
Cycle_Power_Plant	10.842	2.974	262.614	61.481
Glass	1.077	0.141	4.045	0.172
Concrete	19.913	0.379	272.099	0.994

ליעד נחמיאס Gradient Boosting Regressor – 2 מטלה

מיכאל חבקין יישום אלגוריתמים לומדים 2017-2018

loss	n_estimators	min_samples_leaf	max_depth
LS	**500**	**10**	**6**

	Scikit Gradient Boosting Regressor		SVR Gradient Boosting Regressor	
DataSet Name	MSE	Execution Time (sec)	MSE	Execution Time (sec)
airfoil_self_noise	3.071	0.685	46.519	2.648
Appliances_energy_prediction	5600.272	26.514	12953.976	112.828
Cycle_Power_Plant	9.942	3.67	286.131	62.615
Glass	1.082	0.129	4.449	0.159
Concrete	20.933	0.395	273.466	0.952

4. מסקנות:

- <u>זמן ריצה</u>: מודל ה- Gradient Boosting Regressor המקורי מחשב את הממוצע/החציון של ערכי העלים, ואילו הוספת מודל ה-SVR מאלצת את האלגוריתם לבנות מודל רגרסיה נוסף עבור כל עלה ולכן זמן הריצה עבור גרסת האלגוריתם החדשה גבוה יותר. יש לזכור כי חסם עליון לכמות העלים בעץ הוא מספר הרשומות בקובץ האימון ולכן הוספת מודל SVR לכל עלה עלולה לא להיות scalable.
- <u>משתנה n_estimators</u>: הגדלת כמות העצים במודל תרמה לשיפור הביצועים בין הרצות עוקבות עם ערכי המשתנה n_estimators={100, 200, 500, 1000}. אף על פי כן, עבור האלגוריתם החדש, הערך של 100 מסווגים היה מעט טוב יותר, ואילו עבור האלגוריתם המקורי, 1000 מסווגים הצליחו להוריד אף יותר את הטעות. מספר רב יותר של מסווגים יכול להוריד את הסבירות ל-overfitting, אך מספר רב מדי יכול לבטל את ההשפעה החיובית של האנסמבל (כפי שרואים עבור האלגוריתם החדש). החלטנו להגדיר את הערך <u>האופטימלי</u> של <u>500 estimators</u> בשאר הניסויים.
- <u>משתנה ה-Loss</u>: נבדקו ארבעה ערכי הפרמטר loss={ls, lad, huber, quantile}. עבור הרצת האלגוריתם המקורי, התקבל כי שימוש בפונקציית Least Squares הניב את השגיאה הקטנה ביותר, אך עבור האלגוריתם החדש שמימשנו, התקבל כי פונקציית ה-quantile (פונקציית ההתפלגות המצטברת של משתנה המטרה) מזערה את השגיאה בצורה הטובה ביותר.
- <u>משתנה min_samples_leaf</u>: המשתנה מציין את המספר המינימלי של רשומות הנדרש בעלים כדי לקבל החלטת סיווג. מהניסויים שביצענו ראינו כי עבור האלגוריתם המקורי (המשתמש בממצוע/חציון ערכי העלים) מספיק 3 רשומות כדי לקבל החלטה. לעומת זאת, עבור האלגוריתם החדש (המפתח מודל רגרסיה SVR) יש צורך ביותר רשומות שמהם ניתן לבנות מודל רגרסיה. הערך האופטימלי עבור רוב קבצי האימון היה <u>10 רשומות</u> ובו השתמשנו בעת הניסויים על המשתנים האחרים.
- <u>משתנה max_depth</u>: המשתנה מציין את העומק המקסימלי של העצים האינדיוידואליים במודל, ומהווה חסם עליון על מספר הקודקודים במודל. ניתן לראות בוודאות כי הגדלה של עומק העץ מעבר ל-5 רמות מגדילה את הטעות, ולכן עבור קבצי האימון שבחרנו מומלץ לקבוע אותו ל-3 או 5 רמות. עבור האלגוריתם החדש, עץ קטן יותר ישאיר יותר רשומות בעלים, ולכן מודל רגרסיה יהיה מדויק יותר.
- <u>השפעה על MSE</u>: המודל החדש מניב טעות גבוהה יותר בכל המקרים, וזאת עקב בניית רגרסיה על בסיס תכונות הפיצול ברמה מעל העלים, עבורם לא ניתן להסיק בוודאות כי אלה התכונות הכי טובות לבניית המודל (במיוחד עבור קבצי אימון מרובי תכונות).

יש לציין כי במקרים בהם מבוצעת השוואה בין תוצאות אלגוריתמים שונים, היה נכון לבצע מבחן סטטיסטי ולבדוק את המובהקות הסטטיסטית של ההבדלים, אך במסגרת התרגיל הזה לא נדרש.