



Технически Университет – София
Факултет Приложна Математика и Информатика

КУРСОВ ПРОЕКТ

Тема: Задание по дисциплината „Програмиране на Екосистеми за Интернет на Нещата“

Имена на студента: Мариян Василев Апостолов

Факултетен номер: 791322015

Съдържание

1. Теоретична част – Описание на работата на избрания алгоритъм и какво управлява всеки един от параметрите му.....	1
1.1. Описание на Bayesian Ridge.....	1
1.1.1. Параметри.....	2
1.1.2. Атрибути	3
1.1.3. Методи.....	4
2. Алгоритъм за изпълнение на практическата част	4
3. Разработен софтуер – описва разработените променливи и методи.....	6
4. Анализ на резултати и заключение	9
4.1.Резултати.....	9
4.2. Предимства на Bayesian Regression.....	15
4.3. Заключение.....	15
5. Източници.....	15

1. Теоретична част – Описание на работата на избрания алгоритъм и какво управлява всеки един от параметрите му

Избран алгоритъм (модел) – Bayesian Ridge + MultiOutputRegressor

1.1. Описание на Bayesian Ridge

Bayesian Regression е мощен статистически модел, който интегрира предишни знания с наблюдавани данни, за да прави прогнози и да прави изводи за връзки между променливи.

Преди да разгледаме подробно Bayesian Linear Regression, е важно да опишем линейната регресия. Тя е основен статистически метод, използван за моделиране на връзка между една зависима променлива и една или повече независими променливи. Моделът предполага линейна зависимост между независимите променливи. Класическият подход към линейната регресия включва оценка на параметрите на модела с помощта на създадени методи.

Bayesian Linear Regression разширява класическия модел на линейната регресия, като чрез принципите на Bayes. При този подход предишните разпределения са посочени за коефициентите на регресия и тези предишни стойности се актуализират въз основа на наблюдаваните данни, за да се получат последващи предположения. Несигурността в оценките на параметрите се улавят и позволяват вероятностни изводи. Предсказуемостта на зависимата променлива също може да бъде получено, като се вземат предвид както несигурността в оценките на параметрите, така и променливостта в наблюдаваните данни.

Bayesian Regression позволява на естествен механизъм да оцелее при недостатъчни данни или лошо разпределени данни, чрез формулиране на линейна регресия, използвайки разпределители на вероятности, а не точкови оценки. Резултатът или отговорът „у“ се приема, че се извлича от разпределение на вероятностите, а не се оценява като единична стойност.

Математически, за да се получи напълно вероятностен модел, се приема, че отговорът y е базиран на Гаус, разпределен около $X_{\omega}x$, както следва:

$$\rho(y|X, \omega, \alpha) = N(y|X_{\omega}, \alpha)$$

Един от най-полезните видове е Bayesian Regression, която оценява вероятностен модел на регресионния проблем. Тук априорът за коефициента w се дава чрез сферичен Гаус, както следва:

$$\rho(\omega|\lambda) = N(\omega|0, \lambda^{-1}I_p)$$

Този получен модел се нарича Bayesian Ridge Regression и в `scikit-learn` `sklearn.linear_model.BayesianRidge` се използва метода Bayesian Ridge Regression.

1.1.1. Параметри

Следващата таблица съдържа параметрите, използвани в `BayesianRidge`

Параметър	Тип	Стойност по подразбиране	Описание
max_iter	int	None	Максимален брой итерации върху целия набор от данни преди спиране, независимо от всеки критерий за по-ранно спиране. Ако стойността е <code>None</code> , това съответства на <code>max_iter=300</code> .
tol	float	1e-3	Спиране на алгоритъма, ако w е сближено
alpha_1	float	1e-6	Хипер параметър: параметър на формата за гама-разпределението, в сравнение с алфа-параметъра.
alpha_2	float	1e-6	Хипер параметър: параметър на обратната скала (параметър на скоростта) за гама разпределението, в сравнение с алфа параметъра.
lambda_1	float	1e-6	Хипер параметър: параметър на формата за гама разпределението, в сравнение с лямбда параметъра.
lambda_2	float	1e-6	Хипер параметър: обрънат параметър на мащаба (параметър на скоростта) за гама разпределението, в сравнение с лямбда параметъра.
alpha_init	float	None	Начална стойност за алфа (прецизност на шума). Ако не е зададено, <code>alpha_init</code> е $1/\text{Var}(y)$.
lambda_init	float	None	Начална стойност за лямбда (прецизност на дълбочината). Ако не е зададено, <code>lambda_init</code> е 1.
compute_score	bool	False	Ако е <code>True</code> , изчислете логаритмичната пределна вероятност на всяка итерация при оптимизацията.

fit_intercept	bool	True	Отбелязва дали да се изчисли пресечната точка за избрания модел. Прехващането не се третира като вероятностен параметър, следователно няма свързана дисперсия. Ако е зададено на False, при изчисленията няма да се използва отсечка (т.е. данните се очаква да бъдат центрирани).
copy_X	bool	True	Ако е True, X ще бъде копиран; в противен случай може да бъде презаписан.
verbose	bool	False	Подробен режим при добавянето на модела.
n_iter	int		Максимален брой повторения. Трябва да е по-голямо или равно на 1. n_iter е деприкейтнат във 1.3 и ще бъде премахнат във 1.5. Вместо това използвайте max_iter .

1.1.2. Атрибути

Следващата таблица съдържа атрибутите, използвани в BayesianRidge

Атрибут	Тип	Описание
coef_	array-like of shape (n_features,)	Коефициенти на регресионния модел (средно на разпределение)
intercept_	float	Независим атрибут във функцията за решаване. Стойността е зададена на 0.0, ако fit_intercept = False.
alpha_	float	Оценка на точността на шума.
lambda_	float	Оценка на точността на дълбочината.
sigma_	array-like of shape (n_features, n_features)	Оценка на дисперсионната ковариационна матрица на дълбочината.
scores_	array-like of shape (n_iter+1,)	Ако computed_score е True, стойността на пределната вероятност се увеличава при всяка итерация на оптимизацията. Масивът започва със стойността на логаритмичната пределна вероятност, получена за първоначалните стойности на алфа и ламбда, и завършва със стойността, получена за изчисленията алфа и ламбда.
n_iter_	int	Действителният брой повторения за достигане на критерия за спиране.

X_offset_	ndarray of shape (n_features,)	Ако fit_intercept=True, отместването се изважда за закръгление на данните до нулева средна стойност. В противен случай се задава np.zeros(n_features).
X_scale_	ndarray of shape (n_features,)	Задаване на np.ones(n_features).
n_features_in_	int	Брой функции, наблюдавани по време на добавяне (фитване).
feature_names_in_	ndarray of shape (n_features_in_,)	Имена на функции, наблюдавани по време на добавяне (фитване). Дефинира се само когато X има имена на функции, от които всеки един от тях е текст (string).

1.1.3. Методи

Следващата таблица съдържа методите, използвани в BayesianRidge

Метод	Описание
fit(X, y[, sample_weight])	Поставяне (фитване) на модела.
get_metadata_routing()	Вземане на метаданни на обект.
get_params([deep])	Вземане на параметри от оценителя.
predict(X[, return_std])	Прогнозиране с помощта на линеен модел.
score(X, y[, sample_weight])	Връща коефициента за детерминация на изпълненото прогнозиране.
set_fit_request(*[, sample_weight])	Метаданните на заявката се предават на метода за добавяне (фитване).
set_params(**params)	Задаване на параметрите на оценителя.
set_predict_request(*[, return_std])	Метаданните на заявката се предават на метода за прогнозиране.
set_score_request(*[, sample_weight])	Метаданни за заявката се предават на метода за резултата.

2. Алгоритъм за изпълнение на практическата част

2.1. Първа стъпка

В имплементираното приложение е дадена възможност на потребителите да зададат входящи стойности, които впоследствие да представят предвижданите стойности. Създаден е входен файл, с въвеждане на необходимите първоначални стойности.

2.2.Втора стъпка

Предстои проверка, за съществуването на записани обединяващи файлове от входящите екселски файлове (day.xls и hour.xls).

При тяхната липса се преминава към стъпка 2.2.1.

Ако са налични, преминаваме към стъпка 2.3.

2.2.1. Преминаваме през процес на прочит, обединяване на колонките, премахване на липсващите данни между основните два входящи файла и тяхното записване в избрана папка. Преминаваме към стъпка 2.3.

2.3.Трета стъпка

Данните от всяка метеорологична станция се извличат от записаните обединяващи файлове. Преминава се през процес на тяхното трансформиране преди обучение. Използваме трансформатора RobustScaler, за да тестваме предсказанията след тяхното моделиране.

След трансформацията се създава transformer joblib модел на всяка станция. Подготвените данни се предават към следващата стъпка 2.4.

2.4.Четвърта стъпка

Трансформираните данни се подават (фитват) към модела BayesianRidge, към който от входящия файл за настройки се задават стойности за max_iter и n_jobs. Регресионния модел преминава и през MultiOutputRegressor, както е предвидено по задание.

Като резултат след моделирането се създава joblib модел за всяка метеорологична станция.

Резултатните данни се записват в променлива, след изпълнението на имплементирания метод.

Преминаваме към стъпка 2.5.

2.5.Пета стъпка

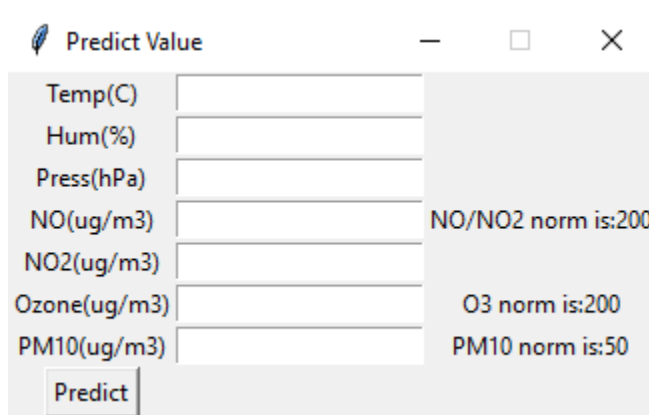
Записаните данни от стъпка 2.4. и 2.5. се подават в отделен модул за тестване и анализ. Извеждаме следната информация:

- Принтира се стойността на изчисленият `model.score()`
- Показва се изчисленият `model.predict()`
- Показва се и се записва графика с преглед на регресията по време на предвиждането на стойностите (NO, NO2, O3, PM10) за всяка метеорологична станция.

След завършване на тестовете се преминава към стъпка 2.6.

2.6.Шеста стъпка

Създава се графичен интерфейс на модела през Tkinter. Имаме създаден входящ файл с настройките, чрез който може да се избере създадения модел на метеорологичната станция.

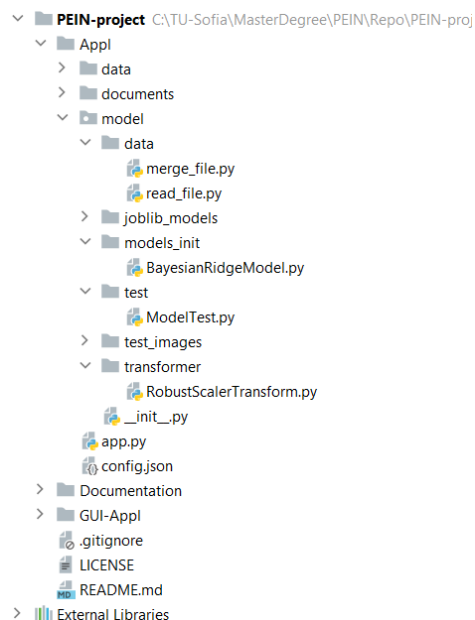


The screenshot shows a Tkinter window titled "Predict Value". It features a list of input fields on the left: Temp(C), Hum(%), Press(hPa), NO(ug/m3), NO2(ug/m3), Ozone(ug/m3), and PM10(ug/m3). To the right of these fields, there are three lines of text indicating norms: "NO/NO2 norm is:200", "O3 norm is:200", and "PM10 norm is:50". At the bottom left, there is a button labeled "Predict". The window has standard Tkinter window controls (minimize, maximize, close) in the top right corner.

3. Разработен софтуер – описва разработените променливи и методи

3.1.Софтуерен продукт

На фиг.3.1.1. е показана архитектурната подредба на основните файлове на имплементирания софтуерен продукт.



Фигура 3.1.1. Архитектурна подредба на файловете

В таблицата е представено кратко описание на всяка папка:

Папка	Под-папка	Описание
data		Съдържа входящите екселски файлове в софтуерния продукт
documents		Съдържа новосъздадените екселски файлове с обединена информация от входящите
model	data	Съдържа Python файлове за прочит на входящите файлове и обединяване на информацията от избраните колонки
model	joblib_models	Съдържа създадените модели
model	test	Съдържа Python файлове за тестване и анализ на избрания модел
model	test_images	Съдържа създадените графики в режим на тестване и анализ на избрания модел
model	transformer	Съдържа Python файлове за трансформиране на данните

Основния файл за стартиране на софтуерното приложение е **app.py**. На фиг.3.1.2. е представено неговото съдържание.


```

1 import warnings
2 from Appl.model import App as app
3
4 warnings.filterwarnings('ignore')
5
6 if __name__ == '__main__':
7     app().run()

```

Фигура 3.1.2 Основен файл на софтуерния продукт

На фиг. 3.1.3. е показан входящия файл с настройки в приложението, в който се задават входящите параметри **config.json**:

- **i_files**: Списък с входящите файлове, които ще се проверяват и ще оформят получения модел.
- **i_sheets**: Списък с екселските sheet-ове, които са част от всеки файл
- **i_usecols**: Избор на колонките от екселските файлове, които ще бъдат обединени в записания файл
- **test_size**: Параметър, който се подава в метода `train_test_split` на стъпката за трансформиране на данните
- **train_size**: Параметър, който се подава в метода `train_test_split` на стъпката за трансформиране на данните
- **random_state**: Параметър, който се подава в метода `train_test_split` на стъпката за трансформиране на данните
- **max_iter**: Параметър, който се подава при добавянето (фитването) на трансформираните данни към `BayesianRidge` на стъпката за създаване на модела
- **n_jobs**: Параметър, който се подава при добавянето (фитването) на трансформираните данни към `BayesianRidge` на стъпката за създаване на модела

```

1 {
2     "i_files": ["day.xls", "hour.xls"],
3     "i_sheets": ["orlov", "mladost", "druzba", "nadejda", "pavlovo", "hipodruma"],
4     "i_usecols": ["A:C", [0, 2, 3, 4, 6, 7]],
5     "test_size": 0.3,
6     "train_size": 0.7,
7     "random_state": 0,
8     "max_iter": 1000,
9     "n_jobs": -1
10 }

```

Фигура 3.1.3. Основен файл с настройки на софтуерния продукт

3.2.Графичен интерфейс на модела

На фиг.3.2.1. е показана архитектурната подредба на основните файлове на имплементирания графичен интерфейс.



Фигура 3.2.1. Графичен интерфейс на модела

На фиг. 3.2.2. е показан входящия файл с настройки в приложението, в който се задават входящите параметри **config.json**:

- **model**: Параметър за избрания модел на графичния интерфейс



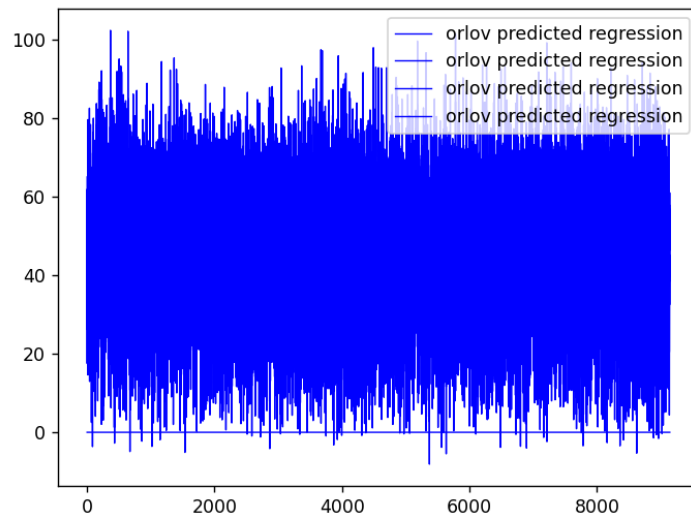
Фигура 3.2.2. Файл с конфигурация на избрания модел на графичния интерфейс

4. Анализ на резултати и заключение

4.1.Резултати

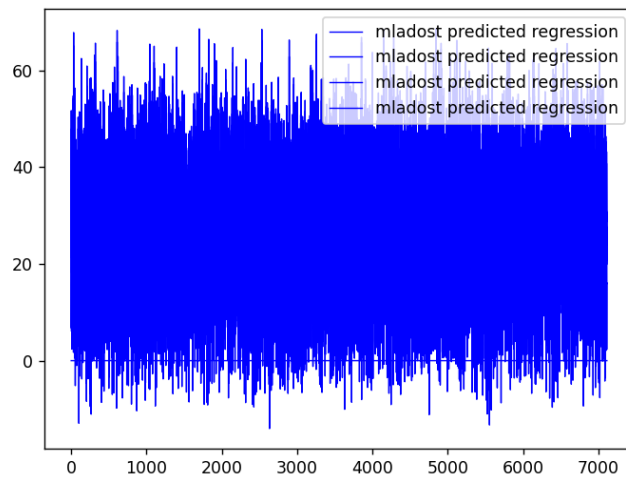
4.1.1. Тест 1

- Входящи данни
 - `"test_size": 0.3,`
`"train_size": 0.7,`
`"random_state": 0,`
`"max_iter": 1000,`
`"n_jobs": -1`
- Резултати
 - Orlov
 - Средна стойност на предвидените резултати
0.15477191467493423
 - Регресивна графика на предвижданията



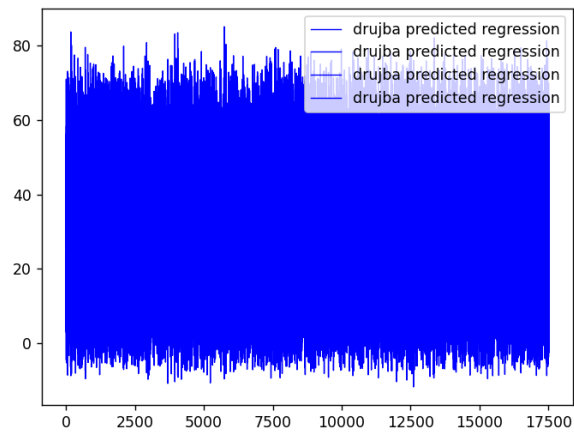
○ Младост

- Средна стойност на предвидените резултати
-0.7727370025467165
- Регресивна графика на предвижданията



○ Дружба

- Средна стойност на предвидените резултати
-32.55427128875114
- Регресивна графика на предвижданията

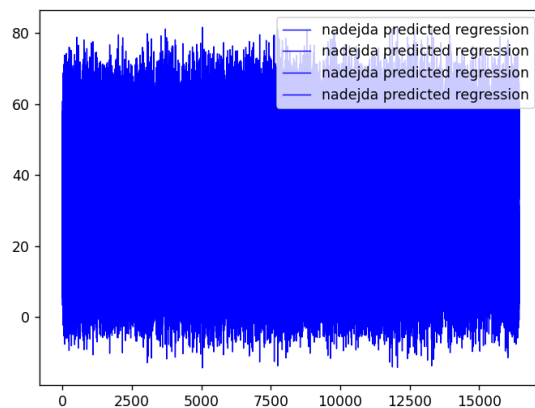


○ Nadejda

- Средна стойност на предвидените резултати

-0.338858247052394

- Регресивна графика на предвижданията

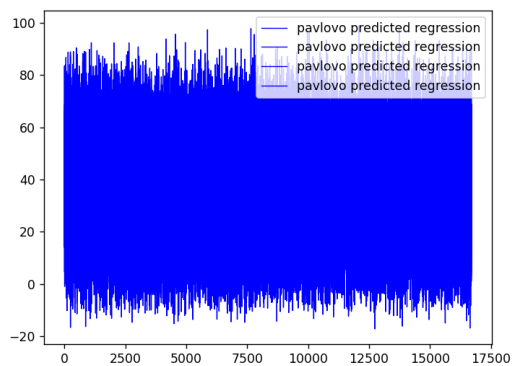


○ Pavlovo

- Средна стойност на предвидените резултати

-326.9677174449787

- Регресивна графика на предвижданията

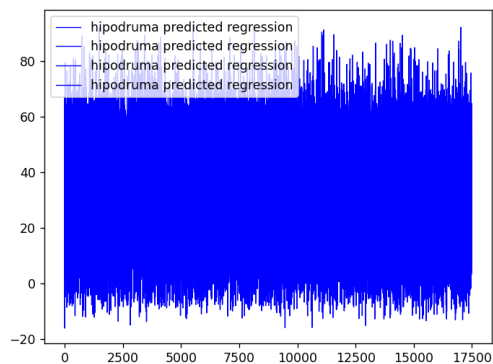


- Hipodruma

- Средна стойност на предвидените резултати

-35.71522856697344

- Регресивна графика на предвижданията



4.1.2. Тест 2

- Входящи данни

- `"test_size": 0.5,`
`"train_size": 0.5,`
`"random_state": 1,`
`"max_iter": 500,`
`"n_jobs": -1`

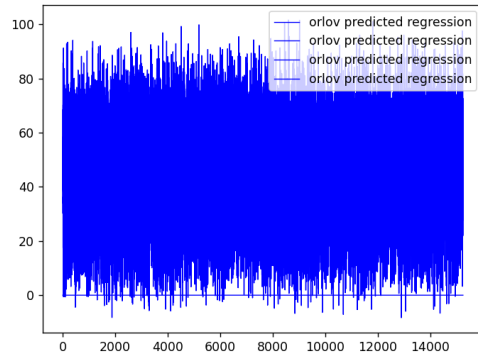
- Резултати

- Orlov

- Средна стойност на предвидените резултати

0.174815392072124

- Регресивна графика на предвижданията

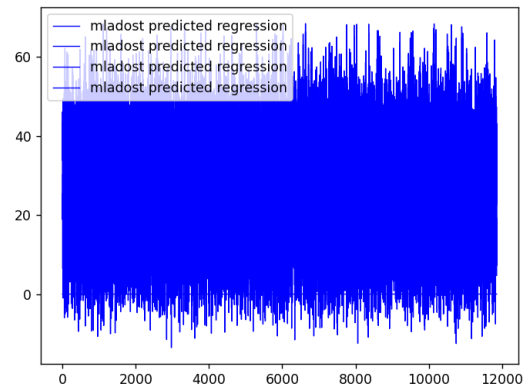


○ Mladost

- Средна стойност на предвидените резултати

-0.6624945575959204

- Регресивна графика на предвижданията

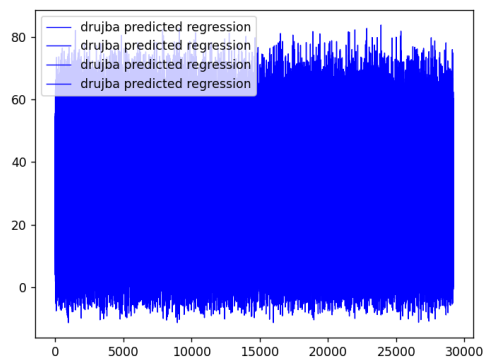


○ Drujba

- Средна стойност на предвидените резултати

-34.86234347398408

- Регресивна графика на предвижданията

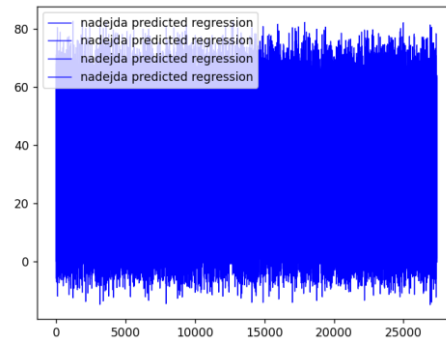


- Nadejda

- Средна стойност на предвидените резултати

-0.3612073376711312

- Регресивна графика на предвижданията

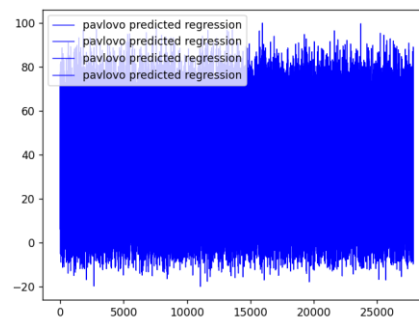


- Pavlovo

- Средна стойност на предвидените резултати

-351.6002811052098

- Регресивна графика на предвижданията

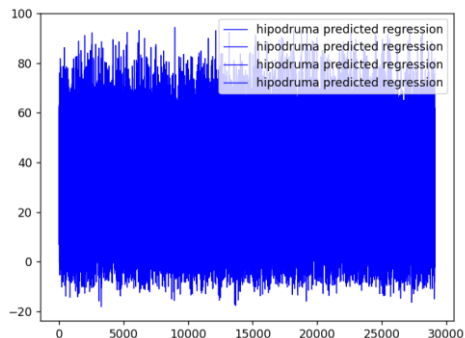


- Hipodruma

- Средна стойност на предвидените резултати

-36.15863344258178

- Регресивна графика на предвижданията



4.2. Предимства на Bayesian Regression

- Отчитане на несигурността: Bayesian Regression осигурява последователна рамка за количествено определяне и включване на несигурността в процеса на моделиране. Това е особено ценно, когато се работи с малки или шумни набори от данни.
- Гъвкавост в предварителната спецификация: Bayesian Regression позволява включването на предишни знания или вярвания относно параметрите, което може да бъде особено полезно при работа с информация, специфична за домейна.
- Боравене с мултиколинеарност: Bayesian-ските методи могат ефективно да се справят със ситуации, при които независимите променливи са силно корелирани, което може да бъде проблематично за класическите регресионни методи.
- Регуляризация: Bayesian Regression естествено включва регуляризация чрез избора на предишни разпределения, осигурявайки начин за предотвратяване на пренастройването и подобряване на ефективността на генерализацията.

4.3. Заключение

Bayesian Regression предлага оформена рамка за моделиране на връзката между променливите, включваща предишни знания и количествено определяне на несигурността. Приложението му в области на приложна регресия демонстрира своята ефективност при справяне със сложни предизвикателства при моделиране и подобряване на точността на прогнозиране.

5. Източници

Заглавие	Линк
sklearn.linear_model.BayesianRidge	https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.BayesianRidge.html
Scikit Learn - Bayesian Ridge Regression	https://www.tutorialspoint.com/scikit_learn/scikit_learn_bayesian_ridge_regression.htm
sklearn.multioutput.MultiOutputRegressor	https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html