Харківський національний університет радіоелектроніки

(повне найменування вищого навчального закладу)

Кафедра штучного інтелекту

(повна назва кафедри)

**КУРСОВА РОБОТА**

з дисципліни «Машинне навчання»\_\_\_\_\_\_\_\_\_\_\_\_\_

(назва дисципліни)

на тему: практичне дослідження узагальнених лінійних моделей (Generalized Linear Models) з використанням інструментарію Scikit-learn на мові програмування Python

Студента (ки) \_3\_курсу 17-7 групи

напряму підготовки\_\_\_\_\_ІТКН\_\_\_\_\_\_\_\_

\_\_Нефьодова Даниїла Андрійовича \_\_\_\_

(прізвище та ініціали)

Керівник \_доцент каф. ШІ, доц., к.т.н.\_\_\_ \_\_\_\_\_\_\_\_\_Вітько О. В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_ Вітько О. В.\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_ Кулішова Н.Є.\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Філатов В.О.\_\_\_

(підпис) (прізвище та ініціали

Харків - 2019

\_\_\_\_\_\_\_\_\_\_\_\_Харківський національний університет радiоелектронiки\_\_\_\_

# Інститут, факультет, відділення\_\_\_\_\_\_\_КН\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Кафедра, циклова комісія\_\_\_\_\_ШІ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Освітньо кваліфікаційний рівень\_\_\_\_бакалавр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Напрям підготовки\_\_\_6.050101  «Комп’ютерні науки»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_

# 

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 року

## **ЗАВДАННЯ**

## **НА КУРСОВУ РОБОТУ**

## **З ДИСЦИПЛІНИ «МАШИННЕ НАВЧАННЯ»**

**студенту** Нефьодову Даниїлу Андрійовичу

(прізвище, ім’я, по батькові)

**1. Тема роботи:** практичне дослідження узагальнених лінійних моделей (Generalized Linear Models) з використанням інструментарію Scikit-learn на мові програмування Python

**2. Термін здачі студентом закінченої роботи** 26.12.2019

**3. Вихідні дані до проекту:**розробити серверну і клієнтську частини інформаційної системи «Ресторан». Серверна частина являє собою реалізацію бази даних, розроблену для платформи СУБД SQLite. Клієнтська частина має забезпечувати виконання наступних бізнес-функцій. Бізнес-функції для неавторизованих користувачів: перегляд продукції за категоріями, обчислення можливої ціни замовлення, реєстрація на сайті, вхід на сайт. Бізнес-функції авторизованих користувачів користувачів: оформлення замовлення, вихід, перегляд особистого кабінету. Бізнес-функції адміністратора: перегляд замовлень, зміна статусу замовлень(зі статусу «Принято до обробки» до статусів «На виконанні», «Виконано» або «Скасовано»),. Операційна система Windows XP або вище, програмне забезпечення: Python 3.7.2, DataGrip, PyCharm, CASE-засіб All Fusion Data Modeler (ERWin).

**4. Зміст розрахунково‑пояснювальної записки (перелік питань, котрі підлягають розробці):** провести аналіз бізнес-процесів (бізнес-функцій) предметної області та виділити ті з них, які вимагають автоматизації; сформулювати та оформити вимоги до інформаційної системи; створити функціональну модель інформаційної системи («TO-BE»), використовуючи стандарт IDEF0; провести функціональне моделювання, визначити й уточнити вимоги до інформаційної системи; провести логічне моделювання БД з використанням стандарту IDEF1Х; обґрунтувати вибір платформи СУБД для реалізації БД; провести фізичне моделювання БД з використанням стандарту IDEF1Х; провести UML-моделювання проектованої клієнтської частини інформаційної системи, створивши діаграму прецедентів (Use Case Diagram), діаграму послідовності дій (Sequence Diagram), діаграму станів (Statechart Diagram), діаграму активності (Activity Diagram) і діаграму класів (Class Diagram); провести аналіз і виділити основні бізнес-процеси з поділом бізнес-функцій, що виконуються на стороні клієнтської і серверної частин інформаційної системи; реалізувати фізичну модель БД для обраної платформи СУБД, створивши серверну частину інформаційної системи; реалізувати посилальну цілісність даних, а також одну з функцій бізнес-процесу на стороні серверної частини інформаційної системи; реалізувати один або кілька бізнес-процесів на стороні клієнтської частини інформаційної системи, розробивши інтерфейс доступу до БД; розробити відповідно до ГОСТ 34.69890 експлуатаційний документ «Керівництво користувача»; підготувати відповідно до ГОСТ 19.401-78 програмний документ «Текст програми»; підготувати проектний документ «Vision and Scope».

**5. Перелік графічного матеріалу (з точним визначенням обов’язкових креслень):** Концептуальна діаграма бізнес-процесу та її декомпозиції(5 діаграм), логічна та фізична моделі даних, діаграми прецедентів(use case), класів, послідовностей(4 основних), станів(statechart) і активності(activity).

**6. Дата видачі завдання**:

**Керівник роботи** Вітько Олександра Валеріївна

(підпис) (прізвище, ім’я, по батькові)

**Студент** Нефьодов Даниїл Андрійович

(підпис) (прізвище, ім’я, по батькові)

« » 20 р

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва етапів курсового проекту | Термін виконання | Примітка |
| 1 | Отримання завдання на курсову роботу |  |  |
| 2 | Аналіз завдання |  |  |
| 3 | Пошук літератури з теми роботи |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 | Розробка презентації, підготовка доповіді |  |  |
| 10 | Захист курсової роботи |  |  |

**Керівник роботи** Вітько Олександра Валеріївна \_

(підпис) (прізвище, ім’я, по батькові)

**Студент** Нефьодов Даниїл Андрійович

(підпис) (прізвище, ім’я, по батькові)

« » 20 р

**РЕФЕРАТ**

Пояснювальна записка до міждисциплінарного курсового проекту містить 55 сторінок, 18 рисунків, 5 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, WEB-САЙТ , БАЗА ДАНИХ, SQLite, IDEF0, USE CASE, PYTHON 3.7.2, DJANGO 2.2, HTML, JAVASCRIPT.

Об'єктом досліджень міждисциплінарного курсового проекту є процес електронної комерції в інформаційній системі, оформлення, обліку і виконання замовлень, документування всіх облікових даних.

Предметом досліджень міждисциплінарного курсового проекту є інформаційні технології і програмні методи створення клієнтської і серверної частин інформаційної системи, що дозволяє автоматизувати облік замовлень у ресторані.

Мета досліджень: розробка клієнтської і серверної частин інформаційної системи «Ресторан».

В роботі проведено аналіз предметної області, що відноситься до комерційної діяльності. Для визначення і уточнення вимог до розроблюваної інформаційної системи проведено функціональне моделювання (відповідно до стандарту IDEF0), логічне і фізичне моделювання даних (відповідно до стандарту IDEF1Х). Розроблено діаграми UML-моделі. Проведено проектування клієнтської і серверної частин інформаційної системи «Ресторан». За результатами тестування проведено аналіз відповідності розробленого програмного забезпечення інформаційної системи висунутим вимогам.

ЗМІСТ

[Перелік умовних позначень, символів, одиниць, скорочень і термінів](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922110) 6

[ВСТУП 8](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922111)

[1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 9](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922112)

[1.1 Аналіз предметної області, яка визначає діяльність ресторану 9](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922113)

[1.2 Визначення та аналіз основних бізнес-процесів 9](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922114)

[1.3 Постановка завдання на курсове проектування 10](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922115)

[2 РОЗРОБКА ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ 1](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922116)1

[2.1 Визначення функціональних вимог до ІС 11](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922117)

[2.2 Логічне і фізичне модулювання БД ІС 1](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922118)7

[2.3 Розробка вимог до функцій серверної частини ІС 1](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922119)9

[2.4 UML-моделювання клієнтської частини ІС 1](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922120)9

[2.5 Розробка вимог до функцій інтерфейсу клієнтської частини ІС 2](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922121)7

[3. ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ 2](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922122)8

[3.1 Обґрунтування вибору мови програмування 2](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922123)8

[3.2 Обґрунтування вибору СУБД 28](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922124)

[3.3 Створення БД для обраної платформи СУБД 2](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922125)8

[3.4 Розробка інтерфейсу клієнтської частини ІС](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922127) 29

[3.5 Тестування розробленого програмного забезпечення 30](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922128)

[3.6 Аналіз дослідної експлуатації та варіантів використання ІС 3](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922129)0

[ВИСНОВКИ 3](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922130)1

[ПЕРЕЛІК ПОСИЛАНЬ 3](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922131)2

[Додаток А 3](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922132)3

[Додаток Б 42](file:///C:\Users\Danyil\Downloads\Telegram%20Desktop\Пояснувальна_записка_НЕФЬОДОВ_ДАНИЇЛ.docx#_Toc10922133)

Додаток В 51

Перелік умовних позначень, символів, одиниць, скорочень і термінів

GLM – узагальнені лінійні моделі (з англ. Generalized Linear Models);

scikit-learn –

Python –

Модель –

ВСТУП

На сьогодні жодна сфера життя суспільства не може обійтись без прогнозів як засобу передбачення поведінки процесів та об’єктів у майбутньому.

Одним за таких засобів є регресійний аналіз, моделі якого склали йому репутація надійного інструменту аналізу. Головним плюсом цього методу є зведення причини і наслідку.

У даній роботі розглянуто набір моделей, що входять до модулю scikit-learn мови програмування Python.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про узагальнену лінійну модель

Узагальнена лінійна модель – це статистична лінійна модель, що визначається наступним рівнянням:

де – це залежна передбачувана величина, – незалежна передбачувана величина, – вектор коефіцієнтів, – вектор перетину.

Але лінійні моделі підходять тільки для даних, що задовольняють наступним умовам:

1. Лінійний взаємозв’язок між змінними;
2. Нормальний розподіл залишків – різниця між реальними даними та даними на регресійній прямій;
3. Перевірка на мультіколлінеарність;
4. Бажано нормальне розподілення змінних.

До узагальнених лінійних моделей відносять:

1. Метод найменших квадратів (Ordinary Least Squares);
2. Ridge-регресію;
3. Lasso;
4. Elastic Net;
5. Логістичну регресію.

1.2 Постановка задачі на курсову роботу

Головним завданням курсового проектування є дослідження вищевказаних моделей. Розробка програм, що засновані на даних моделях. Дослідження принципів та умов їх застосування.

На базі створених моделей, ознайомитися з їх поведінкою на деяких вибірках даних. Проаналізувати отримані результати.

2 ОПИС АЛГОРИТМІВ УЗАГАЛЬНЕНИХ ЛІНІЙНИХ МОДЕЛЕЙ

2.1 Лінійна регресія

В основі мінімізації функції втрат лінійної регресії є ідея метода найменших квадратів, що визначая та для того, щоб використовувати отриману модель робити подальші прогнози.

Для спрощення використаємо вектор формулу з одним предикатом:

За методом найменшим квадратів ми мінімізуємо квадратичні помилки:

Переназначимо квадратичну суму помилок як:

Тоді зведемо задачу оптимізації наступним чином:

Перетворивши вирази отримаємо:

Отримавши значення отримаємо рівняння моделі, яке буде виглядати як:

2.2 Ridge-регресія

У випадку Ridge-регресії до формули 2.2 додається штраф, еквівалентний квадрату суми коєфіціентів вектора :

де – параметр складності, який контролює коефіцієнти нахилу.

При графік регресії наближається до графіка аналогічної функції вартості лінійної регресії.

Ridge підходить, якщо мається мало предикатів та усі вони мають бути релевантними для передбачення.

2.3 Регресія Lasso

У випадку регресії Lasso до формули 2.2 додається штраф, але вже еквівалентний сумі модулів коефіцієнтів вектора :

де – параметр складності, який контролює коефіцієнти нахилу.

Точно так як і функція вартості Ridge-регресії п ри графік регресії наближається до графіка аналогічної функції вартості лінійної регресії.

Алгоритм Lasso використовується, якщо мається багато предикатів, деякі з яких не такі важливі, ніж інші.

2.4 Elastic Net

Elastic Net – це гібрид Lasso та Ridge, де включені штрафи як по абсолютній величині, так і у квадраті, що регулюються коефіцієнтом :

де – параметр складності, який контролює коефіцієнти нахилу.

2.5 Логістична регресія

Логістична регресія - це статистичний метод аналізу даних, в якому є один або кілька незалежних значень, що визначають результат. Результат вимірюється за допомогою дихотомічної змінної – у котрій є тільки два можливих варіанта або класа 0, або класа 1.

Тож по-перше потрібно розрахувати ймовірність того, що спостереження класу 1:

Коефіцієнти обрані так, щоб максимізувати вірогідність приналежності спостереження до класу 1.

Після чого обирається порогова границя , яка чітко класифікує задане вхідне значення в один з класів, яка породжує та визначає ступінь прийнятності до помилок 1-го та 2-го родів.

2.6 Використані метрики

Для оцінки результатів моделювання лінійної регресії, Ridge, Lasso та Elastic Net викостаєто метрики середньоквадратичної помилки та коефіцієнт детермінації.

Середньоквадратична помилка (позначається як )– величина, що характеризує стандартне відхилення вибіркового середнього, розраховане по вибірці розміром n із генеральної сукупності.

Коефіцієнт детермінації (позначається як ) — статистичний показник, що використовується в статистичних моделях як міра залежності варіації залежної змінної від варіації незалежних змінних. Вказує наскільки отримані спостереження підтверджують модель.

Частка правильно класифікованих об’єктів (позначається як ) — ймовірність того, що клас передбаченої правильно.

де – кількості правильних позитивних, правильних негативних, хибних позитивних та хибних негативних прогнозів відповідно.

F-міра – це поєднання метрик точності та повноти



3 ОПИС ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

3.1 Датасети

По-перше потрібно згенерувати дата сет. Scikit-learn містить велику кількість генераторів датасетів, генерацію яких можна контролювати вхідними параметрами функцій генерування. Ми використаємо make\_regression для генерування датасета для лінійної регресії та методів її регуляризації та make\_classification для логістичної регресії.

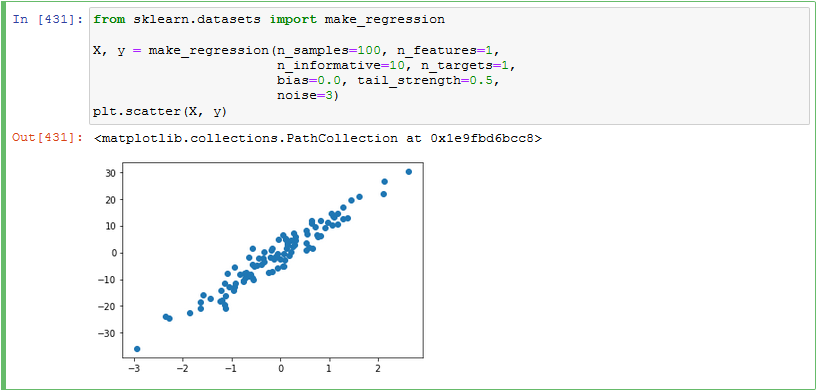


Рисунок 3.1 – Фрагмент коду та змодельований датасет

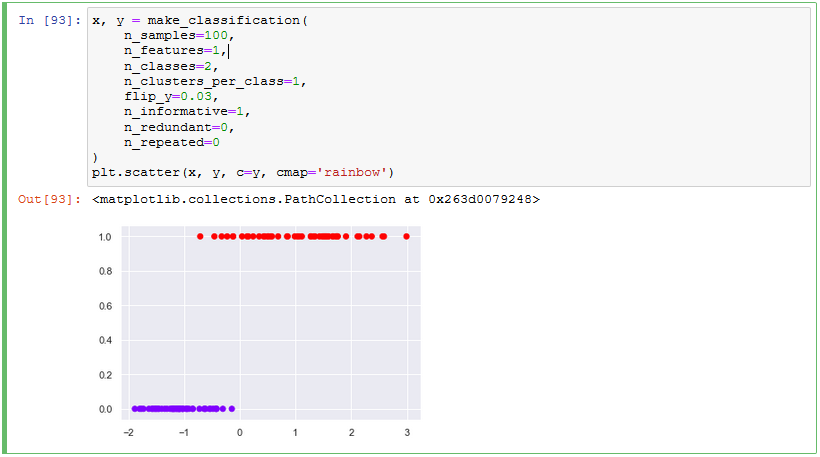


Рисунок 3.2 – Фрагмент коду та змодельований датасет

3.2 Програмування та порівняння лінійної регресії, Ridge, Lasso та Elastic Net

Після генерації датасета, на ньому перевіримо дані методи. Порівнювати методи будемо за метриками RMSE та R2:

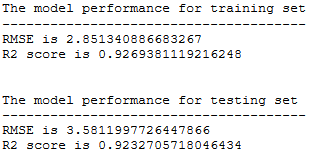
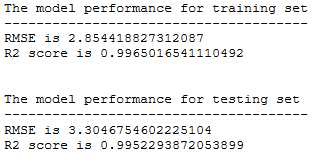
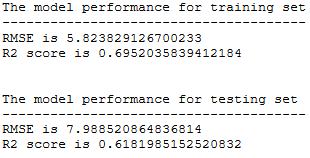


Рисунок 3.3 – Результат лінійної регресії



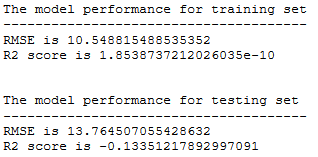
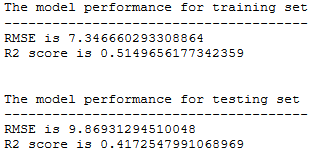


Рисунок 3.4 – Результат методу Ridge з параметром ,

що дорівнює 1, , 2,

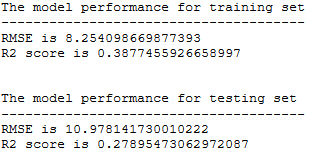
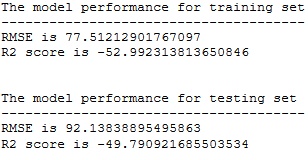
 

Рисунок 3.5 – Результат методу Lasso з параметром ,

що дорівнює 1,

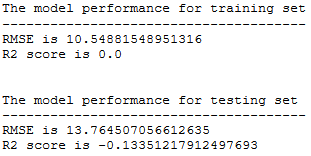
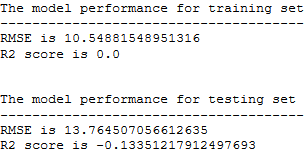
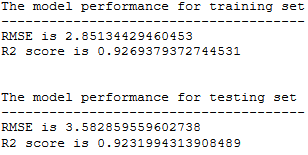
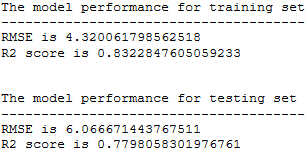
 

Рисунок 3.6 – Результат методу Lasso з параметром ,

що дорівнює 2,



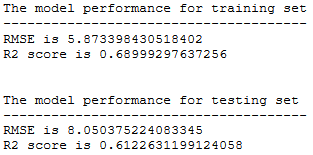
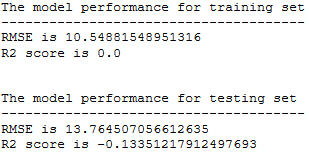
 

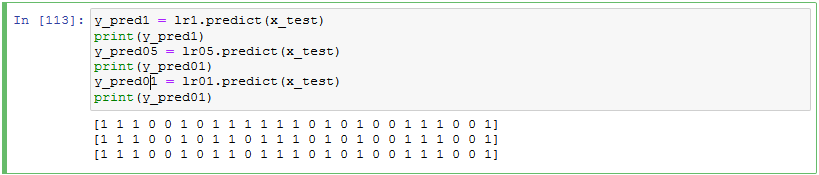
Рисунок 3.7 – Результат методу Elastic Net з параметром ,

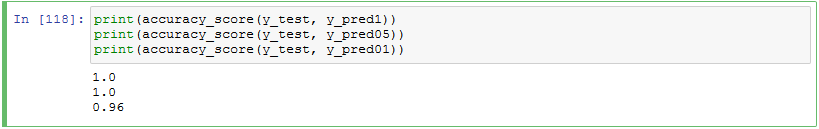
що дорівнює 1, , 2,

Отримані результати свідчать, що на даному дата сеті най адекватніше спрацьовує лінійна регресія, з показником , інші моделі видають гірший результат, що збільшується при зменшенні параметра , тобто регулярізація у даному датасеті зайва, адже не виконується умова перенавчання при використанні мінімізації найменшої суми квадратів у лінійній регресії.

3.2 Програмування та оцінка результатів логістичної регресії

Порівняємо результати навчання логістичної регресії при різних порогових границях . Для цього візьмемо значення , що дорівнюють 1, 0.5 та 0.1. У ході проведення дослідів ми отримали наступні дані для вище перелічених відповідно:

 Рисунок 3.8 – Результати прогнозування моделей



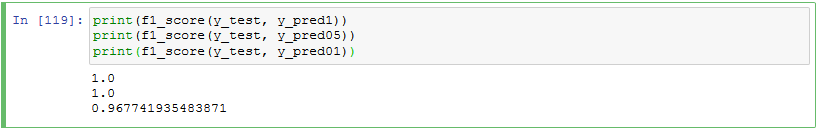


Рисунок 3.9 – Оцінки моделей за метриками частки правильно класифікованих об’єктів та F-мірою

Для

Machine learning is a hot topic in research and industry, with new methodologies developed all the time. The speed and complexity of the field makes keeping up with new techniques difficult even for experts – and potentially overwhelming for beginners.

that represents data in the context of a – problem, often a business problem. The aim is to go from data to insight. For example, if an online retailer wants to anticipate sales for the next quarter, they might use a machine learning algorithm that predicts those sales based on past sales and other relevant data. Similarly, a windmill manufacturer might visually monitor important equipment and feed the video data through algorithms trained to identify dangerous cracks.

Next methods can be the base of machine learning knowledge: regression, classification, clustering, dimensionality reduction etc.

Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels. For example, we can build a classification model to categorize bank loan applications as either safe or risky. Such analysis can help provide us with a better understanding of the data at large. Many classification methods have been proposed by researchers in machine learning, pattern recognition, and statistics. Most algorithms are memory resident, typically assuming a small data size. Recent data mining research has built on such work, developing scalable classification and prediction techniques capable of handling large amounts of disk-resident data. Classification has numerous applications, including fraud detection, target marketing, performance prediction, manufacturing, and medical diagnosis.

Classification can help predict whether an online customer will buy a product. The output can be yes or no: buyer or not buyer. But classification methods aren’t limited to two classes. For example, a classification method could help to assess whether a given image contains a car or a truck. In this case, the output will be 3 different values: 1) the image contains a car, 2) the image contains a truck, or 3) the image contains neither a car nor a truck.

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation.

There are some types of classification algorithms in Machine Learning:

a) Linear Classifiers: Logistic Regression, Naive Bayes Classifier

b) Nearest Neighbor;

c) Support Vector Machines;

1. Decision Trees;
2. Boosted Trees;
3. Random Forest;
4. Neural Networks.

GaussianNB, Logistic Regression and Random Forest will be considered in this coursework.

* 1. Logistic Regression

Logistic regression estimates the probability of an occurrence of an event based on one or more inputs.

For instance, a logistic regression can take as inputs two exam scores for a student in order to estimate the probability that the student will get admitted to a college. Because the estimate is a probability, the output is a number between 0 and 1, where 1 represents complete certainty. For the student, if the estimated probability is greater than 0.5, then we predict that he or she will be admitted. If the estimated probability is less than 0.5, we predict the he or she will be refused.

The figure 1.1 plots the scores of previous students along with whether they were admitted. Logistic regression allows us to draw a line that represents the decision boundary.

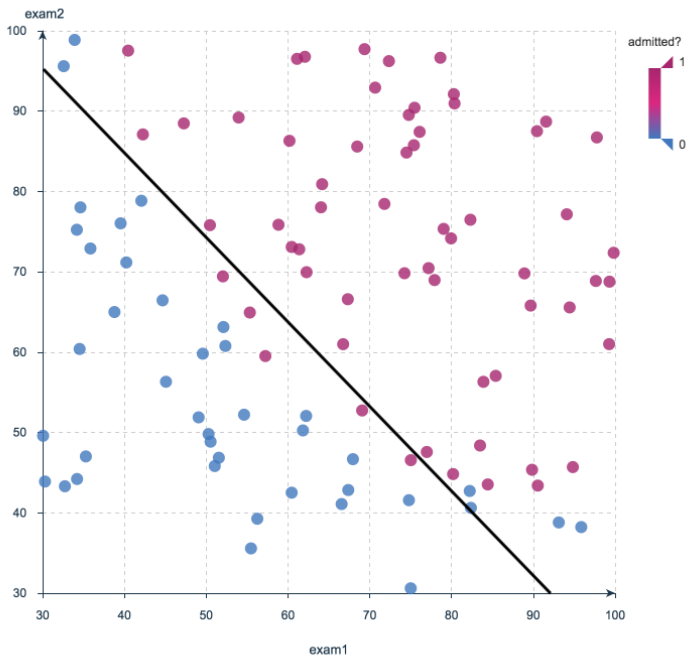


Figure 1.1 – Logistic Regression Decision Boundary

1.3 Gaussian Naïve Bayes

Gaussian Naïve Bayes is an algorithm having a Probabilistic Approach. It involves prior and posterior probability calculation of the classes in the dataset and the test data given a class respectively. In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell-shaped curve which is symmetric about the mean of the feature values as shown in the figure 1.2 below:

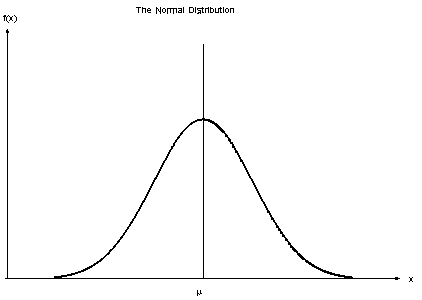
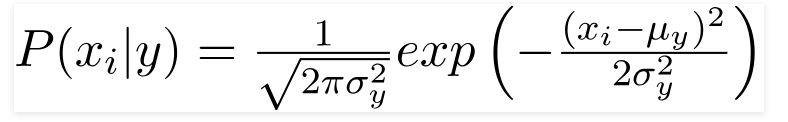


Figure 1.2 – Gaussian Distribution (Normal Distribution)

The likelihood of the features is assumed to be Gaussian. Conditional probability is given by next:

(1.1)

1.4 Random Forest

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction. The figure 1.3 represents the random forest.

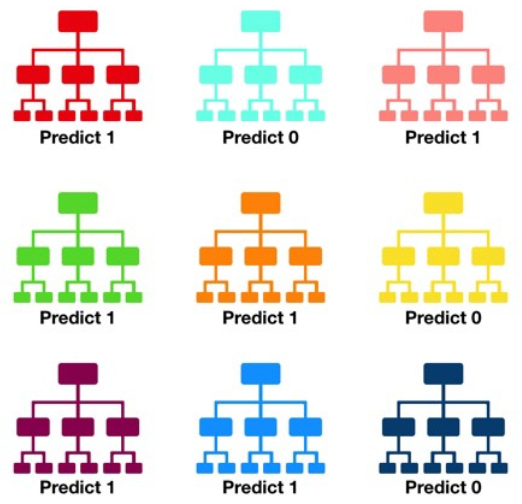


Figure 1.3 – Visualization of a Random Forest Model Making a Prediction

The fundamental concept behind random forest is a simple but powerful one – the wisdom of crowds. In data science speak, the reason that the random forest model works so well is: “Many relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.”.

* 1. Formulation of the problem

The problem of probability calibration in machine learning will be thoroughly discussed in Section 3.1, Section 3.2, Section 3.3. In this section we will define a formulation of the problem and a sequence of steps to achieve it.

A big part of machine learning is classification – we want to know what class (group) an observation belongs to. The ability to precisely classify observations is extremely valuable for various business applications like predicting whether a particular user will buy a product or forecasting whether a given loan will default or not.

Data science provides a of classification algorithms such as logistic regression, support vector machine, naive Bayes classifier, and decision trees. However, not all classifiers return well-calibrated forecasts in a machine learning model for classification problems. For example, GaussianNB worse than LogisticRegression. Calibration can resolve this problem. Thus, the question arises of how to achieve more accurate probabilities and how to implement it in a program.

The stages of the research course project consist of:

a) Study of the work of classifiers, provided that the calibration of

probabilities has not been completed;

1. Consideration of available methods for implementing probability

calibration;

1. Applying the selected method to calibrate probabilities;
2. Graphical presentation of the results;
3. Results analyzing and conclusion.

2 THEORETICAL STUDIES

2.1 Example of calibration

In machine learning, most classification models produce predictions of class probabilities between 0 and 1, then have an option of turning probabilistic outputs to class predictions. Even algorithms that only produce scores like support vector machine, can be retrofitted to produce probability-like predictions.

For a binary classification problem, there are summary metrics — accuracy, precision, recall, F1-score, and so on — that evaluate the quality of binary 0s and 1s outputs. If the outputs are not binary but are floating numbers between 0 and 1, then we can use them as scores for ranking. But we cannot trust floating numbers between 0 and 1 as probabilities.

A model’s output can be viewed as a statement saying how likely something should happen. An example of such a model is the weather service. In particular, how likely is it going to rain. The figure 2.2 represents the weather forecast in Kharkov.

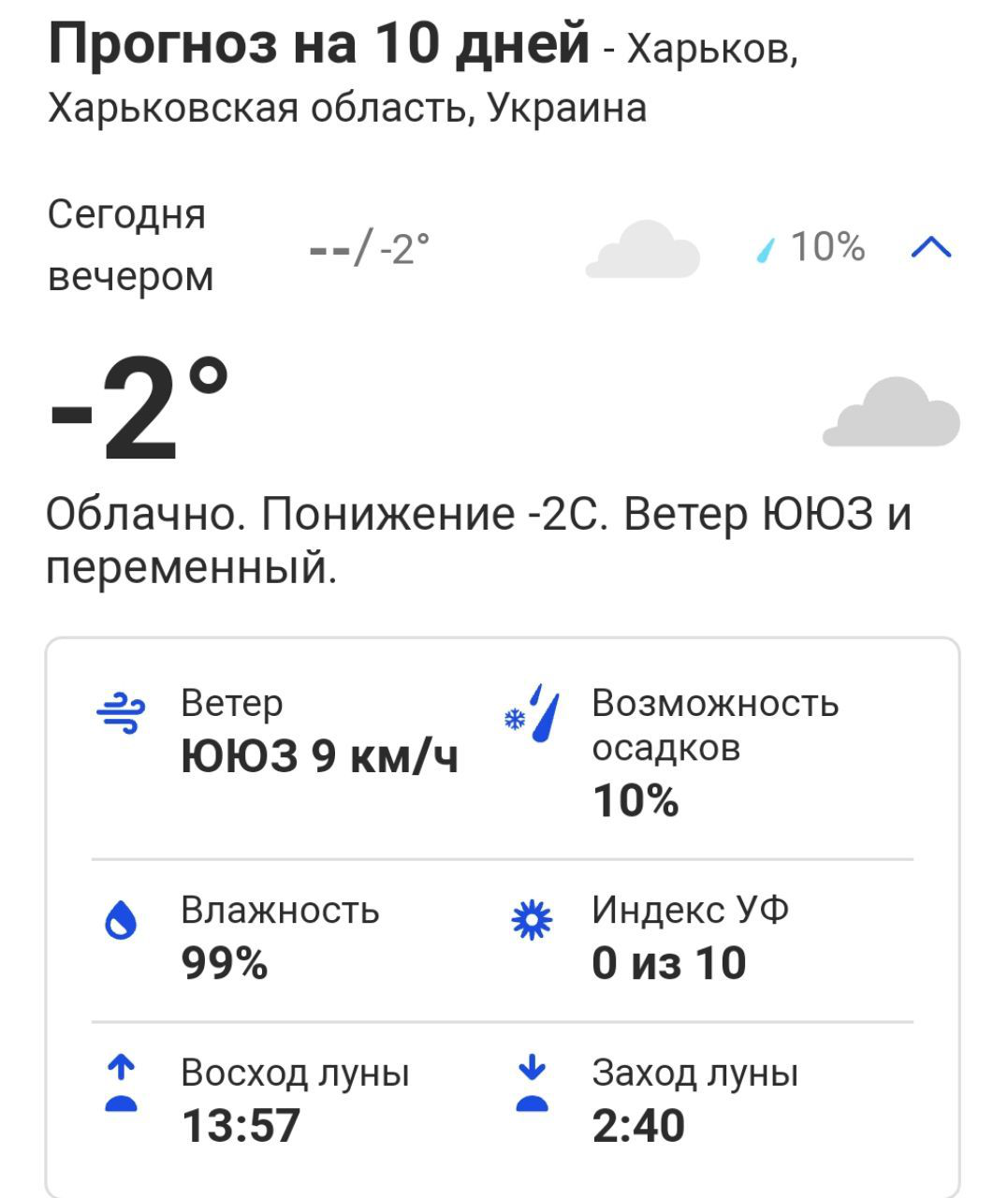


Figure 2.2 - Weather forecast in Kharkov

Weather forecast at the figure 2.2 says that this evening there’s an 10% chance of rain. How trustworthy is this 10%? An accurate weather forecast means that if we looked into 100 days that are predicted with an 10% chance of rain, then there should be around 10 rainy days. It also has to be accurate in other probability ranges. For days that are called to rain 40% of the time, there should be 40 rainy days out of 100 days on average. If this weather forecast service’s predictions all follow this good pattern, then we say their predictions are calibrated.

2.2 Probability calibration

In the Section 3.2 we will analyze in detail how some classifiers cannot return well-calibrated forecasts in a machine learning model for classification problems.

Using three methods as an example, let’s see how probabilities look without calibration. The figure 2.3 represents a comparison of curves made by GaussianNB, Logistic Regression and Random Forest.

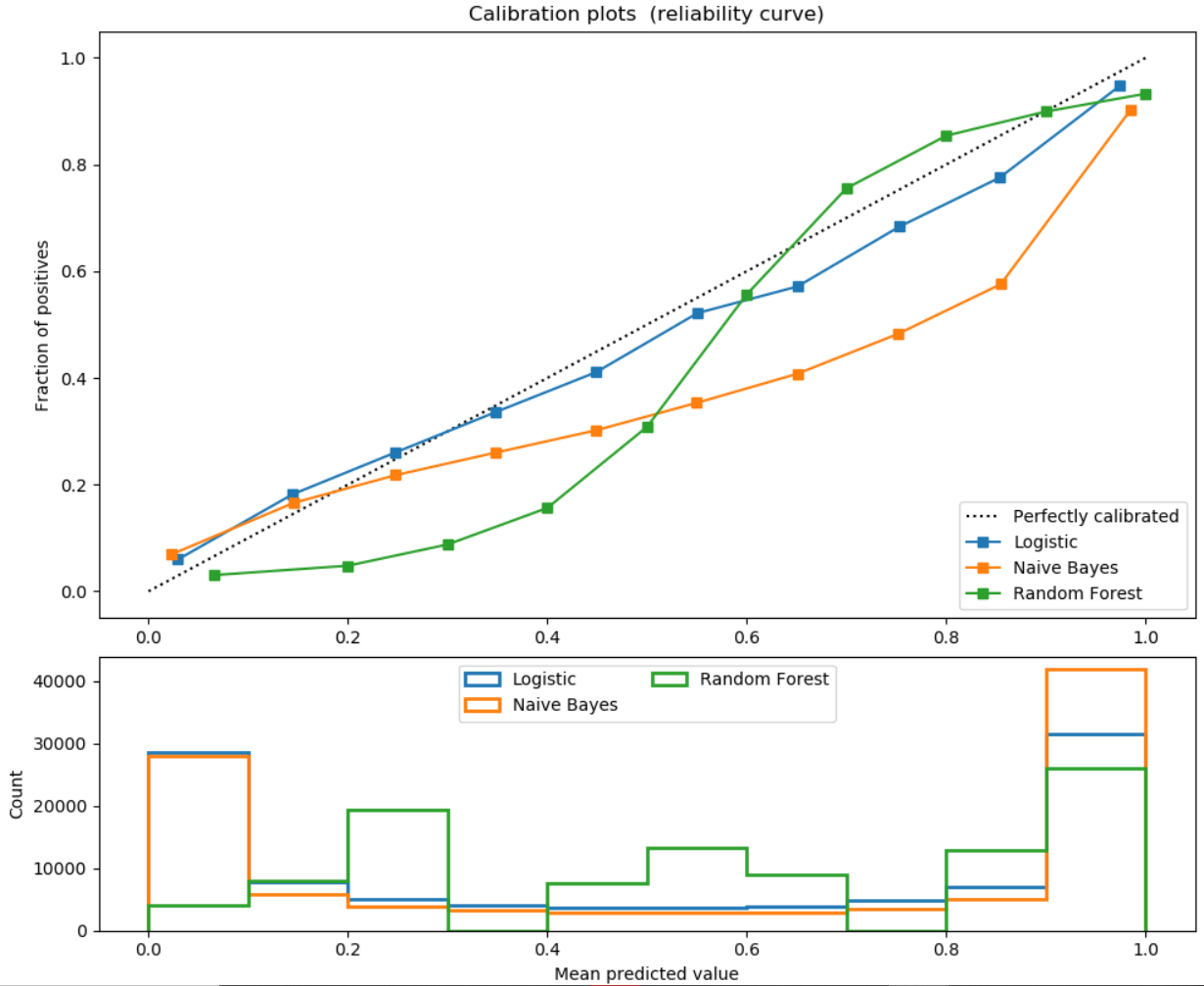


Figure 2.3 – Comparison of curves

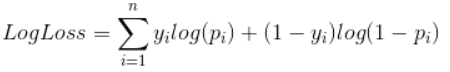
As we can see, GaussianNB and Random Forest show a very big amount of biases in the probabilities. Maximum margin methods such as SVM, boosted trees etc push the real posterior probability away from 0 and 1 while methods such as Naive Bayes tend to push the probabilities towards 0 and 1. And in cases where predicting the accurate probabilities is more important, this poses a serious problem. Boosted trees, random forests and SVMs performs best after calibration.

So, we need to understand how to improve accuracy of predicted probabilities. This problem can be solved in practice in one of two powerful methods.

But first we need to understand what LogLoss Metric is, for the reason that this is exactly the example where the accuracy of the predicted probabilities is very important.

2.3 LogLoss Metric

This Evaluation Metric is employed in the classification tasks where rather than predicting the actual classes, one needs to predict the probabilities corresponding to that classes. So, in a way this metric measures the quality of predictions rather than the accuracy. The formula 2.1 represents Log Loss.

 (2.1)

In the formula 2.1 yi is the actual class value (i.e 0 or 1 in case of binary classification) of a particular observation (row) and pi is the predicted probability of that class.

2.4 Platt scaling

Platt scaling is a way of transforming classification output into probability distribution. For example: If you’ve got the dependent variable as 0 & 1 in train data set, using this method you can convert it into probability.

Let’s now understand how Platt Scaling is applied in real Predictive Modeling problems (in order):

* 1. Split the train data set into training set and Cross Validation set;
  2. Train the model on the training data set;
  3. Score test data set and Cross Validation data set;
  4. Run a logistic model on the Cross-Validation data set using the actual

dependent variable and the predicted values;

* 1. Score the test data set using the model created in step 4 with feature as the

output of scoring on test data set in step 3.

2.5 Isotonic Scaling

Isotonic Regression is similar to Platt Scaling. It’s a non-parametric regression

technique. Non-parametric means that it doesn’t make any assumptions such as of linearity among variables, constant error variance etc.

Isotonic regression fits a piecewise constant, non-decreasing function to a distribution of data. At the figure 2.4 an example of a fitted isotonic regression result is represented.

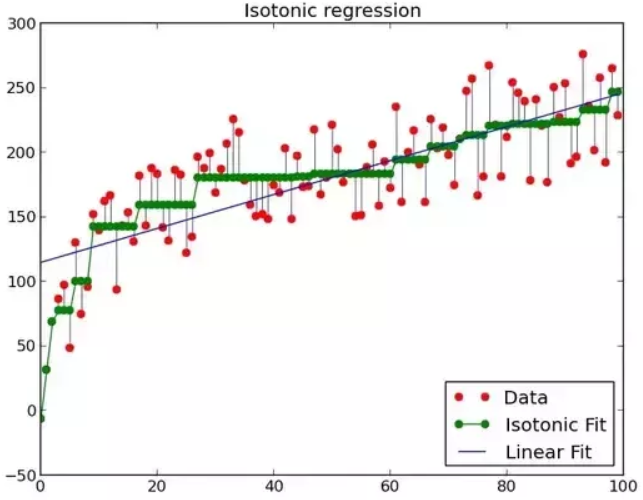


Figure 2.4 – Isotonic regression

Isotonic regression is occasionally useful on its own, but it finds its primary application in solving this scaling problem. Whilst Platt scaling applies a logistic transform to the scores generated by the underlying classifier, isotonic regression applies isotonic regression to them, but the end goal (minimize cumulative neighborhood probability error) is the same.

3 EXPERIMENTAL STUDIES

3.1 Dataset

At first, it is necessary to have a dataset. Scikit-learn includes various random sample generators that can be used to build artificial datasets of controlled size and complexity. For the purposes of this coursework we will use (versions of) the following synthetic, two-class classification dataset, generated by the make\_classification method packaged into sklearn. Make\_classification create multiclass datasets by allocating each class one or more normally-distributed clusters of points. make\_classification specializes in introducing noise by way of: correlated, redundant and uninformative features; multiple Gaussian clusters per class; and linear transformations of the feature space. The piece of code is represented at the figure 3.1.

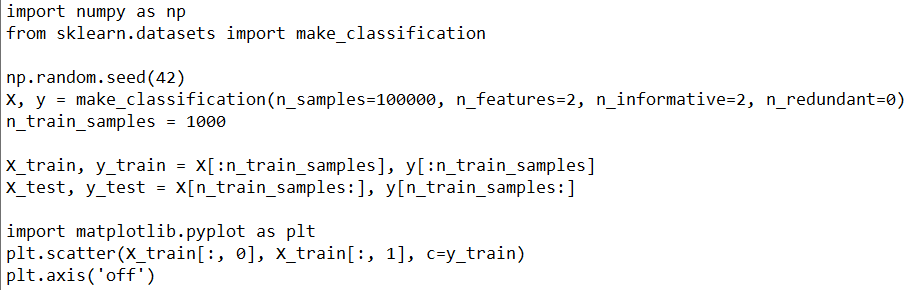


Figure 3.1 – Code fragment

At the figure 3.1 n\_samples is the number of samples; n\_features is the total number of features. These comprise n\_informative informative features and n\_redundant redundant features. The default value for n\_informative is 2.

Figure 3.2 represents the generated dataset with two classes and 100 000 samples:



Figure 3.2 – Generated dataset

3.2 Calibration curves

A great way of checking how a classifier's probability forecasting is

performing on a dataset of interest is using a so-called calibration curve. Building a calibration curve is shown on the figure 3.3 below:

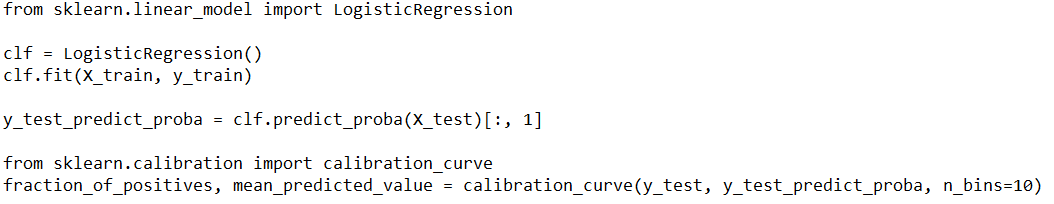


Figure 3.3 – Code fragment

For binary classification tasks predict\_proba returns a matrix containing the first class proba in the first entry, and the second class proba in the second entry. Since there are only two classes one is just 1 - n of the other. The calibration\_curve implementation expects just one of these classes in an array, so we index that.

The calibration curve works by sorting the probabilities assigned to the records being predicted by the probability reported by the classifier. It then bins the values and calculates two things.

The fraction\_of\_positives on the figure 3.4 is the percentage of records in the chosen bin which belong to the dominant class. This is determined by looking at what values these points are actually assigned in y, and it is the empirical truth.

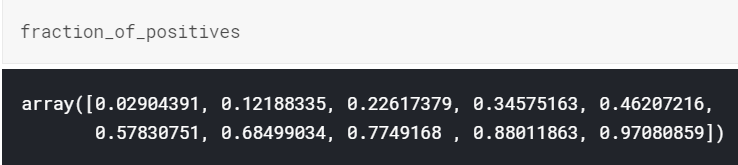


Figure 3.4 – Fraction of positives

The second thing returned, mean\_predicted\_value on the figure 3.5, is the mean probability of these points belonging to the dominant class reported by the algorithm.

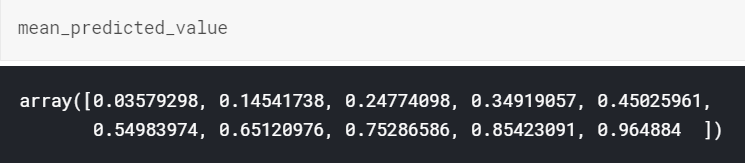


Figure 3.5 – Mean predicted value

If the classifier is forecasting probabilities that are accurate, then we expect the percentage of dominant class classifications and the mean probabilities assigned to the dominant classes in each bin to be close to one another. If it is not doing so accurately, then we will see these two values diverge.

Calibration curve is represented at the figure 3.7.

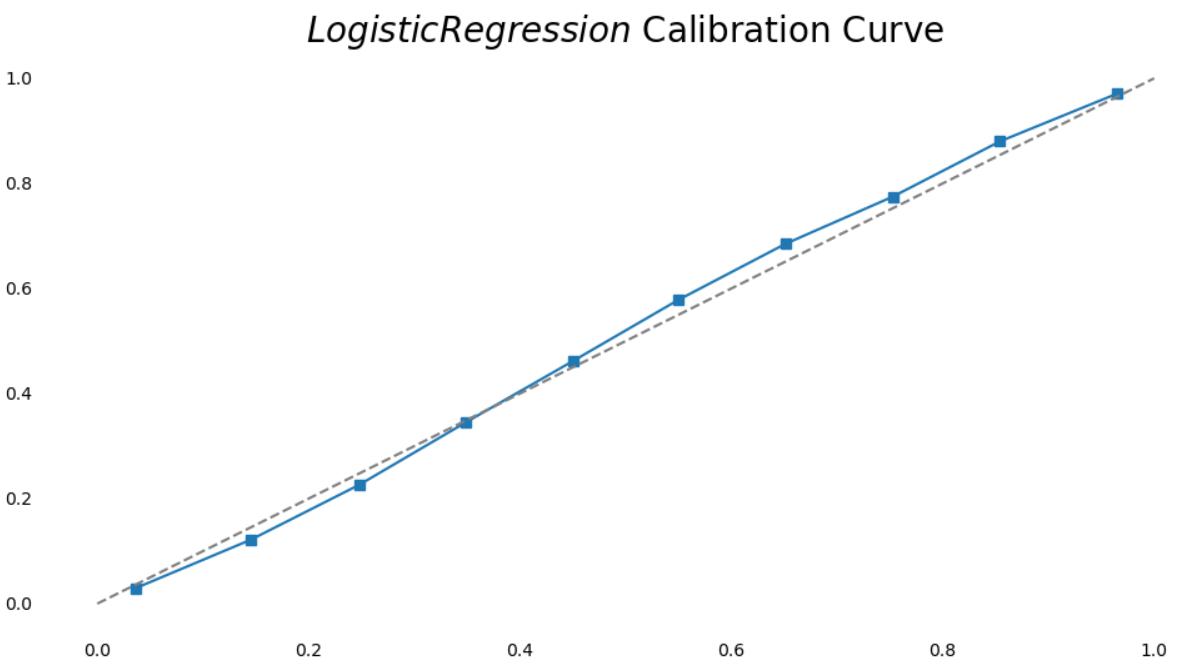


Figure 3.7 – LogisticRegression Calibration Curve

At the figure 3.7 LogisticRegression generates probability predictions that are extremely close to optimal. Some of this is due to the simplicity of the used dataset, but most of this is due to logistic regression itself. Logistic regression tends to have quite accurate probability predictions because it is optimizing log-odds, which is just a convenient restatement of class probability. In other words, probability figures directly in the cost function that LogisticRegression solves for, and hence, unsurprisingly, the algorithm produces unbiased probability estimates.

Unlike the logistic regression, there are also bad examples of an algorithms: GaussianNB and RandomForestClassifier. Figure 3.9 represents a comparing the calibration curves.

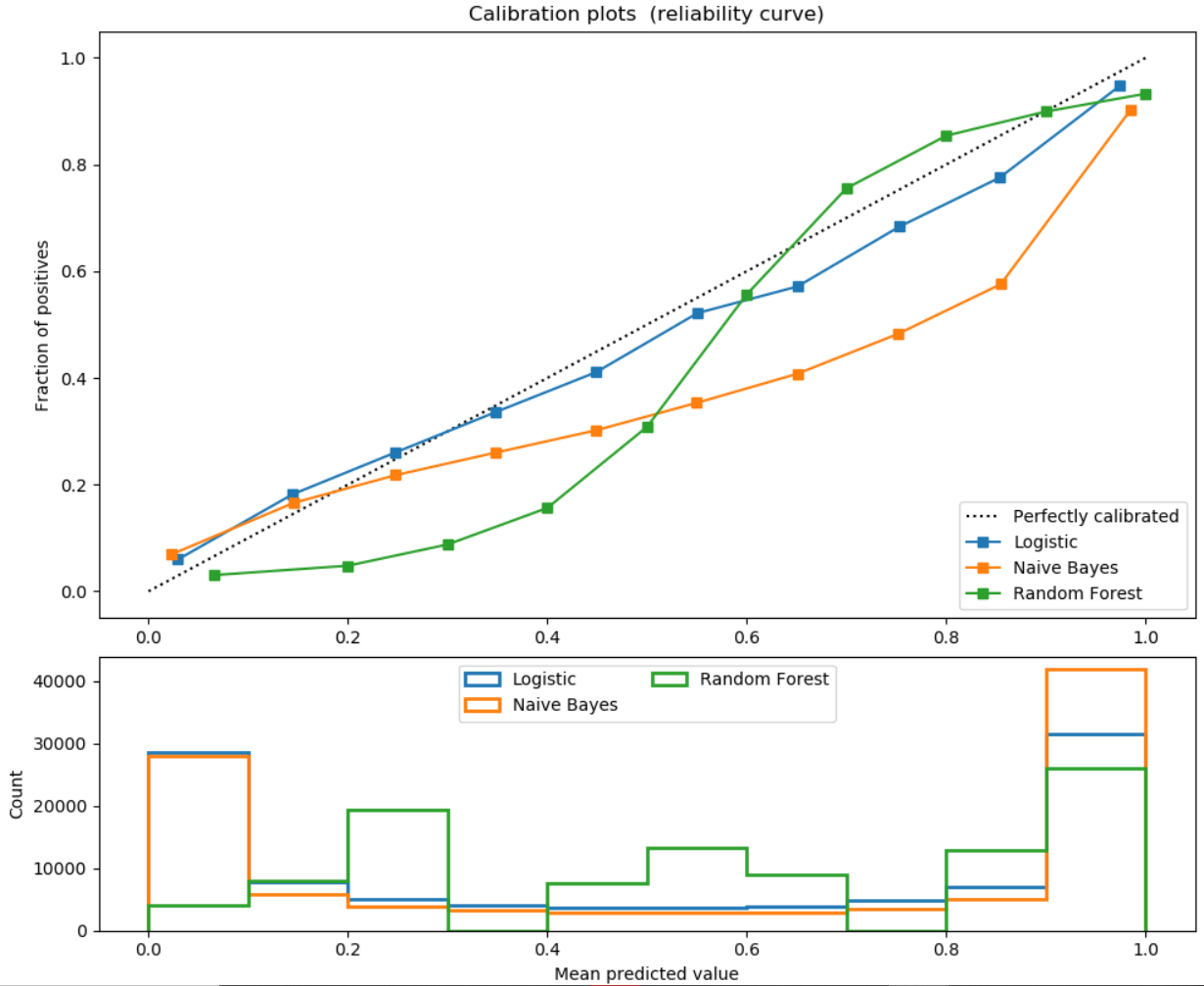


Figure 3.9 – Calibration plots

Naive Bayes algorithms will tend to push probabilities relatively near to, but not all the way at, 0 and 1. This is due to the fact that this algorithm assumes conditional independence of every feature in the dataset, a fairly ridiculous assumption that breaks down very quickly when there is redundancy in the dataset (e.g. columns containing similar information, and hence being non-independent).

In fact, the Naive Bayes algorithm is a great demonstration of how probability is not strongly relevant to pure classification tasks (assigning 0s and 1s). In practice the probabilities generated by the naive Bayes classifier are highly suspect. However, when we use these probabilities to classify, assigning each point to the y class that maximizes its likelihood, we get a result that works remarkably well remarkably often.

RandomForestClassifier shows the opposite behavior: the histograms show peaks at approximately 0.2 and 0.9 probability, while probabilities close to 0 or 1 are very rare.

* 1. Programming probability calibration

Two ways to solve the problem were previously considered: Platt scaling and

isotonic scaling. Both forms of classifier probability calibration are provided by the handy CalibratedClassifierCV transform. The appropriate algorithm to perform the calibration can be specified at declaration time (sigmoid, which is Platt, or isotonic).

The RandomForestClassifier was already returning relatively accurate probability scores, so a correction wasn't strictly necessary. However, the applying the Platt transformation did perhaps further reduce the bias. The plot is represented at the figure 3.10.

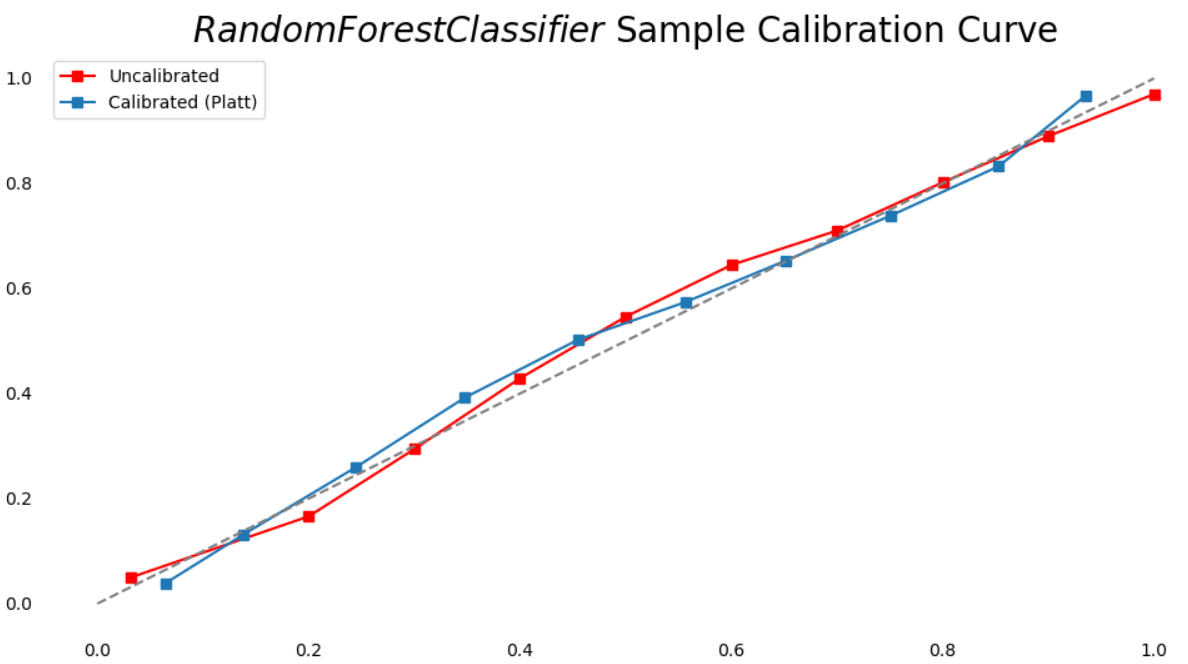


Figure 3.10 – RandomForestClassifire Sample CalibrationCurve

The plot at the figure 3.11 is more interesting.

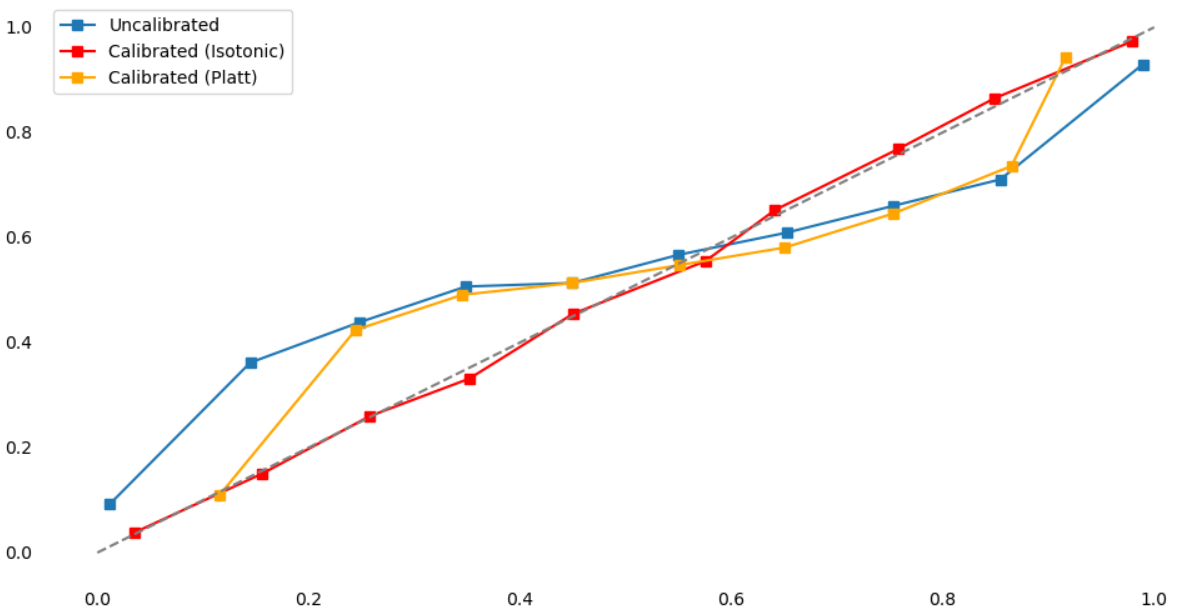


Figure 3.11 – GaussianNB Calibration Curve

In the blue there is a calibration curve for the original GaussianNB classifier. In the yellow is the recalibrated curve is generated with a Platt correction. It is seen that the Platt correction did more harm than good. At the original calibration curves for the various algorithms, it is seen that for the data used in that example, GaussianNB did not truly have the same sigmoidal error structure than RandomForest have. For this data, it overshoots the low probabilities by too little, and undershoots the high-probability data by too much. Platt calibration works when the amount of bias on either side is close to the same, and it fails here.

Isotonic calibration, on the other hand, really gets the job done. It is seen here that the isotonic adjusted probabilities, in red, are a very close match to the true mean probabilities.

СONCLUSIONS

In this coursework we have examined in what cases it is necessary to use a calibration of probabilities. Probability calibration is only useful when we are interested in probabilities. One such case could be if the metric of consideration is Log Loss for model selection where rather than predicting the actual classes, one needs to predict the probabilities corresponding to that classes. This technique Kaggle often employs to improve their scores.

There are three classifiers which are considered at this coursework: GaussianNB, LogisticRegression and RandomForest. Some of them show a very big amount of biases in the probabilities, namely GaussianNB and RandomForest.

We need to understand how to improve accuracy of predicted probabilities. Two ways to realize a calibration was considered: Platt scaling and isotonic scaling.

Using theoretical knowledge and capabilities of the programming language Python we have represented a calibration curve for 3 classifiers. LogisticRegression returns well calibrated predictions by default as it directly optimizes log-loss. In contrast, the other methods return biased probabilities; with different biases per method.

Isotonic scaling was used to calibrate model with RandomForest classifier. Platt and isotonic scaling were used to calibrate model with GaussianNB classifier. The results showed that the Platt correction did more harm than good. On the other hand, Isotonic regression seems to have better fitting performance, and in the general case it does. However, this is only true for sufficiently large datasets (ones with over 10,000 samples). As non-parametric method, isotonic calibration has a strong tendency to overfit. So, this method should to be only used for such problems.

All the results were represented graphically in order to better understand.

REFERENCES

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і

техніки. Структура та правила оформлювання. Чинний від 2017-07-01. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.