МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Аналіз даних
Лабораторний практикум

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за освітньою програмою «Математичні методи моделювання, розпізнавання
образів та комп'ютерного зору»
спеціальності 113 «Прикладна математика»*

Київ
КПІ ім. Ігоря Сікорського
2022

Аналіз даних. Лабораторний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 113 «Прикладна математика» / Н. М. Куссуль, А.Ю. Шелестов, С. А. Тарасенко, Г.О. Яйлимова; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 539 Кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 28 с.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 2 від 30.09.2022 р.) за поданням Вченої ради Фізико-технічного інституту Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (протокол № 11 від 01.09.2022 р.)

Електронне мережне навчальне видання

Аналіз даних

Лабораторний практикум

Автори: *Куссуль Наталія Миколаївна,* д. техн. наук, проф.
Шелестов Андрій Юрійович, д. техн. наук, проф.
Тарасенко Степан Анатолійович, асистент.
Яйлимова Ганна Олексіївна, д. філософії, асистентка

Відповідальний редактор *Смирнов С.А.,* к.ф.-м.н., доц.

Рецензент *Лавренюк А.М.*., канд. техн. наук, доц. кафедри математичного моделювання та аналізу даних НН ФТІ Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського"

Навчальний посібник «Аналіз даних. Лабораторний практикум» присвячено вивченню аналізу даних за допомогою мови Python студентами за спеціальністю 113 «Прикладна математика». Метою є отримання навичок аналізу даних, очистки даних та побудови моделей на основі наявних даних для їх подальшого використання. Посібник містить необхідний теоретичний матеріал, приклади програм, а також завдання для виконання лабораторного практикуму.

ЗМІСТ

Лабораторна робота №1
Data Wrangling

Мета роботи: отримати навички очищення, нормалізації та стандартизації даних (підготовки даних).

1.1. Теоретичні відомості

Data Wrangling – це процес очищення та об'єднання «сирих» і складних наборів даних для легкого доступу та аналізу.

Оскільки кількість даних і джерел даних швидко зростають і розширюються, стає все більш важливою організацією великих обсягів доступних даних для аналізу. Цей процес зазвичай включає в себе ручне перетворення та відображення даних з однієї форми в інший формат, щоб забезпечити більш зручне використання та організацію даних.

На практиці є три загальні завдання, пов'язані з Data Wrangling процесом:

- Data cleaning
- Data transformation
- Data enrichment

Книга Hands-On Data Analysis with Pandas від Packt Publishing, яка написана Стефані Молін охоплює основне розуміння того, як аналітики та вчені збирають та аналізують дані, алгоритми машинного навчання (ML) для виявлення закономірностей та багато іншого. У ній можна ознайомитися з основами Data Wrangling та його процесом.

Для подальшої роботи в межах даного курсу необхідно встановити pandas: https://pandas.pydata.org/docs/getting_started/install.html та matplotlib: https://matplotlib.org/stable/users/installing/index.html.

Також нам знадобиться «Автомобільний датасет» за наступним посиланням: https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data. Ми будемо використовувати цей набір даних протягом усього курсу.

Читання набору даних із URL-адреси та додавання відповідних заголовків

Спочатку ми призначаємо URL-адресу набору даних "ім'я файлу".

```
filename = "https://cf-courses-data.s3.us.cloud-object-
```

```
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DA0101EN-SkillsNetwork/labs/Data%20files/auto.csv"
```

Створюємо список заголовків Python, що містять назви колонок датасету:

```
headers = ["symboling","normalized-losses","make","fuel-type","aspiration", "num-of-doors","body-style",
         "drive-wheels","engine-location","wheel-base", "length","width","height","curb-weight","engine-type",
         "num-of-cylinders", "engine-size","fuel-system","bore","stroke","compression-ratio","horsepower",
         "peak-rpm","city-mpg","highway-mpg","price"]
```

Використовуйте метод Pandas read_csv(), щоб завантажити дані з веб-адреси. Встановіть параметр "names" рівним списку "headers" Python:

```
df = pd.read_csv(filename, names = headers)
```

Використовуйте метод head(), щоб відобразити перші п’ять рядків фрейму даних. Після того як ми викликали функцію датафрейму df.head(), бачимо, у фреймі даних з'явилося кілька знаків питання; це відсутні значення, які можуть перешкодити нашому подальшому аналізу. Отже, як нам визначити всі ці втрачені занчення та впоратися з ними?

Визначення та обробка відсутніх значень

Задача перетворити "?" до NaN (not a number).

У наборі даних автомобілей є відсутні дані. позначені знаком питання "?". Замінюємо "?" на NaN (не число), маркером відсутнього значення Python за замовчуванням з міркувань швидкості та зручності обчислень. Тут ми використовуємо функцію:

.replace(A, B, inplace = True) замінити A на B.

```
df.replace("?", np.nan, inplace = True)
```

Оцінка на предмет відсутніх даних

Відсутні значення перетворюються за замовчуванням. Існує два способи виявлення відсутніх даних:

```
.isnull()
.notnull()
```

```
missing_data = df.isnull()
missing_data.head(5)
```

Вихідним є логічне значення, яке вказує, чи фактично в значенні, яке передається в аргумент, відсутні дані чи наявні. Порахуйте пропущені значення в кожному стовпці. Використовуючи цикл for у Python, ми можемо швидко визначити кількість відсутніх значень у кожному стовпці. Як зазначалося вище, "True" представляє відсутнє значення, а "False" означає, що значення присутнє в наборі даних. У тілі циклу for метод ".value_counts()" підраховує кількість значень "True".

Можливе рішення задачі поставленої вище:

```
for column in missing_data.columns.values.tolist():
    print(column)
    print (missing_data[column].value_counts())
    print("")
```

Як боротися з відсутніми даними?

Видаліть дані

- а. Видаліть весь рядок
- б. Видаліть всю колонку

Замініть дані

- а. Заміна відсутніх даних на середнє значення
- б. Заміна відсутніх даних за частотою
- с. Заміна відсутніх даних за допомогою інших функцій

Цілі стовпці слід вилучати, лише якщо більшість записів у стовпці порожні. У нашому наборі даних жоден із стовпців не є достатньо порожнім, щоб

повністю його видалити. Ми маємо певну свободу у виборі методу заміни даних; однак деякі методи можуть здатися більш розумними, ніж інші.

Обчисліть середнє значення для стовпця «normalized-losses»

```
avg_norm_loss = df["normalized-losses"].astype("float").mean(axis=0)
print("Average of normalized-losses:", avg_norm_loss)
```

Замініть "NaN" середнім значенням у стовпці "normalized-losses"

```
df["normalized-losses"].replace(np.nan, avg_norm_loss, inplace=True)
```

Обчисліть середнє значення для стовпця 'bore'

```
avg_bore=df['bore'].astype('float').mean(axis=0)
print("Average of bore:", avg_bore)
```

Замініть "NaN" середнім значенням у стовпці 'bore'

```
df["bore"].replace(np.nan, avg_bore, inplace=True)
```

Коректування формату даних

Останнім кроком очищення даних є перевірка та переконання, що всі дані мають правильний формат (int, float, текстовий або інший).

У Pandas ми використовуємо:

.dtype(), щоб перевірити тип даних

.astype(), щоб змінити тип даних

Перетворення типу даних у правильний формат:

```
df[["bore", "stroke"]] = df[["bore",
```

```
"stroke"]].astype("float")
df[["normalized-losses"]] = df[["normalized-
losses"]].astype("int")
df[["price"]] = df[["price"]].astype("float")
df[["peak-rpm"]] = df[["peak-rpm"]].astype("float")
```

Стандартизація даних

Дані зазвичай збираються від різних агентств у різних форматах. (Стандартизація даних також є терміном для певного типу нормалізації даних, де ми віднімаємо середнє і ділимо на стандартне відхилення.)

Що таке стандартизація?

Стандартизація — це процес перетворення даних у загальний формат, що дозволяє дослідникам проводити змістовне порівняння.

Нормалізація даних

Нормалізація — це процес перетворення значень кількох змінних у подібний діапазон. Типові нормалізації включають масштабування змінної таким чином, щоб середнє значення змінної становило 0, масштабування змінної, щоб дисперсія становила 1, або масштабування змінної, щоб значення змінної були в діапазоні від 0 до 1.

Бінінг

Бінінг — це процес перетворення неперервних числових змінних у дискретні категоріальні «біни» для згрупованого аналізу.

```
df["horsepower"]=df["horsepower"].astype(int, copy=True)
```

Індикаторна змінна

Індикаторна змінна (або фіктивна змінна) — це числова змінна, яка використовується для позначення категорій. Їх називають dummies, тому що самі числа не мають внутрішнього значення. Ми використовуємо індикаторні змінні, щоб ми могли використовувати категоріальні змінні для регресійного аналізу в наступних модулях.

1.3. Порядок виконання роботи

1. Проаналізувати умову задачі.
2. Написати мовою Python програму для вирішення поставленої задачі.
3. Результати роботи оформити протоколом.

1.4. Завдання

1. Замініть NaN у стовпці ‘stroke’ середнім значенням.
2. Нормалізуйте стовпець ‘height’.
3.1. Створіть змінну індикатора для стовпця ‘aspiration’
3.2. Об’єднайте новий фрейм даних із вихідним фреймом даних, а потім видаліть стовпець ‘aspiration’.
4. Замініть усі NaN у всіх стовбцях середнім значенням.

1.5. Контрольні запитання

1. Що таке стандартизація?
2. Що таке нормалізація?
3. Якими способами можна позбутись відсутності даних?
4. Що таке бінінг?
5. Як нормалізувати дані?

Лабораторна робота №2
Аналіз даних за допомогою Python

Мета роботи: отримати навички аналізу даних використовуючи мову програмування Python.

2.1. Теоретичні відомості

Для подальшої роботи необхідно встановити бібліотеку seaborn https://seaborn.pydata.org/installing.html.

Аналіз індивідуальних моделей функцій за допомогою візуалізації

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Під час візуалізації окремих змінних важливо спочатку зрозуміти, з яким типом змінної ви маєте справу. Це допоможе нам знайти правильний метод візуалізації для цієї змінної.

Неперервні числові змінні — це змінні, які можуть містити будь-яке значення в межах певного діапазону. Вони можуть мати тип "int64" або "float64". Відмінним способом візуалізації цих змінних є використання діаграм розсіювання з підігнаними лініями.

Щоб розпочати розуміння (лінійного) зв'язку між окремою змінною та ціною, ми можемо використовувати "regplot", який відображає діаграму розсіювання плюс підібрану лінію регресії для даних.

Змінні Категорій

Це змінні, які описують «характеристику» одиниці даних і вибираються з невеликої групи категорій. Категоричні змінні можуть мати тип "object" або "int64". Хорошим способом візуалізації категоріальних змінних є використання діаграм (boxplots).

```
sns.boxplot(x="body-style", y="price", data=df)
```

Описовий статистичний аналіз

Давайте спочатку подивимося на змінні, використовуючи функцію опису.

Функція опису автоматично обчислює основну статистику для всіх безперервних змінних. Будь-які значення NaN автоматично пропускаються в цій статистиці. Функція виведе:

- the count of that variable
- the mean
- the standard deviation (std)
- the minimum value
- the IQR (Interquartile Range: 25%, 50% and 75%)
- the maximum value

```
df.describe()
```

Описова статистика – це опис та інтегральні параметри наборів даних. Вона використовує два основні підходи:

Кількісний підхід, який визначає загальні чисельні показники даних. **Візуальний підхід**, що ілюструє дані за допомогою діаграм, графіків, гістограм та інших графічних образів. Описову статистику можна застосовувати до одного або кількох наборів даних або змінних. Коли ви описуєте та обчислюєте характеристики однієї змінної, то виконуєте одномірний аналіз. Коли ви шукаєте статистичні зв'язки між парою змінних, ви робите двомірний аналіз. Аналогічним чином, багатовимірний аналіз пов'язаний із декількома змінними одночасно. Спостереження та вибірки

Спостереження чи генеральна сукупність в статистиці — це набір всіх елементів, щодо яких передбачається робити висновки щодо конкретного завдання. Найчастіше генеральна сукупність дуже велика, що робить її непридатною для збирання та аналізу. Ось чому у статистиці зазвичай намагаються зробити деякі висновки про популяцію, обираючи та досліджуючи репрезентативну підгрупу цієї сукупності.

Така підмножина називається вибіркою. В ідеалі вибірка має задовільно зберігати суттєві статистичні характеристики генеральної сукупності. Таким чином, можна використовувати вибірку для отримання висновків про спостереження.

Викид – це така точка, яка суттєво відрізняється від більшості значень, взятих із вибірки чи сукупності. Є безліч можливих причин появи викидів і ось для початку лише кілька:

Найчастішою причиною появи викидів є помилки збору даних. Наприклад, обмеження вимірювальних приладів або самих процедур збору інформації і це означає, що правильні дані не можуть бути отримані. Інші помилки можуть бути викликані прорахунками, зашумленням даних, людською помилкою та багато іншого.

Точного математичного визначення викидів немає. І тут необхідно покладатися на власний досвід, знання про предмет інтересу та здоровий глузд, щоб зрозуміти, чи справді підозріла точка є аномалією в даних і як слід з нею поводитися.

Value Counts

Підрахунок значень – це хороший спосіб зрозуміти, скільки одиниць кожної характеристики/змінної ми маємо. Ми можемо застосувати метод "value_counts" до стовпця "drive-wheels". Не забувайте, що метод "value_counts" працює лише на series pandas, а не на фреймах даних pandas.

У результаті ми включаємо лише одну дужку df['drive-wheels'], а не дві дужки df[['drive-wheels']].

Basics of Grouping

Метод «groupby» групує дані за різними категоріями. Дані групуються на основі однієї або кількох змінних, а аналіз проводиться для окремих груп.

```
df_group_one = df_group_one.groupby(['drive-wheels'],as_index=False).mean()
```

Давайте використаємо heat map, щоб уявити зв’язок між "body-style" та "price".

```
plt.pcolor(grouped_pivot, cmap='RdBu')
plt.colorbar()
plt.show()
```

Heat map зображує цільову змінну (price) пропорційно кольору щодо змінних 'drive-wheel' і 'body-style' на вертикальній та горизонтальній осях відповідно. Це дозволяє нам уявити, як ціна пов’язана з 'drive-wheel' і 'body-style'. Візуалізація дуже важлива в науці про дані, а пакети візуалізації Python надають велику свободу. Основне питання, на яке ми хочемо відповісти в цьому модулі: «Які основні характеристики найбільше впливають на ціну автомобіля?». Щоб краще виміряти важливі характеристики, ми розглянемо кореляцію цих змінних з ціною автомобіля. Іншими словами: як ціна автомобіля залежить від цієї змінної?

Correlation and Causation

Кореляція: міра ступеня взаємозалежності між змінними.

Причинно-наслідковий зв'язок: зв'язок між причиною і наслідком між двома змінними.

Важливо знати різницю між цими двома. Кореляція не означає причинно-наслідкового зв’язку. Визначення кореляції набагато простіше, ніж визначення причинно-наслідкового зв’язку, оскільки причинно-наслідковий зв’язок може вимагати незалежного експерименту. Кореляція Пірсона вимірює лінійну залежність між двома змінними X і Y, де отриманий коефіцієнт є значенням від -1 до 1 включно, де:

1: Ідеальна позитивна лінійна кореляція.
0: Відсутня лінійна кореляція, дві змінні, швидше за все, не впливають одна на одну.
-1: Ідеальна негативна лінійна кореляція.

Кореляція Пірсона — це стандартний метод функції «corr». Як і раніше, ми можемо обчислити кореляцію Пірсона для змінних 'int64' або 'float64'.

Найбільш загальновідомою мірою залежності між двома величинами є коефіцієнт кореляції Пірсона (англ. Pearson product-moment correlation coefficient, PPMCC, або англ. Pearson's correlation coefficient), який зазвичай називають просто «коефіцієнт кореляції» (англ. the correlation coefficient). Його отримують взяттям відношення коваріації двох розгляданих змінних нашого чисельного набору даних, унормованої квадратним коренем їхніх дисперсій. Математично, коваріацію цих двох змінних просто ділять на добуток їхніх стандартних відхилень. Карл Пірсон розробив цей коефіцієнт на основі подібної, але дещо відмінної ідеї Френсіса Гальтона. Для обчисленя коефіцієнту кореляції Пірсона ми використовуємо формулу

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$, де

- r = коефіцієнт кореляції,
- x_i = значення x-змінної у вибірці
- $\bar{x}$ = середнє значення x-змінної
- y_i = значення y-змінної у вибірці
- $\bar{y}$ = середнє значення x-змінної

Коефіцієнт кореляції Пірсона намагається встановити лінію, яка найкраще допасовується до набору даних із двох змінних, по суті викладаючи очікувані значення, а отриманий коефіцієнт кореляції Пірсона вказує, наскільки далеким від очікуваних значень є фактичний набір даних. Залежно від знаку нашого коефіцієнта кореляції Пірсона ми можемо отримати як від'ємну, так і додатну кореляцію, якщо якийсь зв'язок між змінними нашого набору даних існує.

Коефіцієнти рангової кореляції, такі як коефіцієнт рангової кореляції Спірмена та коефіцієнт рангової кореляції Кендалла (τ), вимірюють, до якої міри в разі збільшення однієї змінної інша змінна схильна збільшуватися, не вимагаючи, щоби це збільшення було подано лінійною залежністю. Якщо за збільшення однієї змінної інша зменшується, то коефіцієнти рангової кореляції будуть від'ємними. Ці коефіцієнти рангової кореляції часто розглядають як альтернативи коефіцієнту Пірсона, які використовують або для зменшення кількості обчислень, або для того, щоби зробити коефіцієнт менш чутливим до не нормальності в розподілах. Проте ця точка зору має мало математичних підстав, оскільки коефіцієнти рангової кореляції вимірюють інший тип зв'язку, ніж коефіцієнт кореляції Пірсона, і їх найкраще розглядати як показники іншого типу зв'язку, а не як альтернативну міру генерального коефіцієнту кореляції.

P-value

Що таке P-value? P-value — це значення ймовірності того, що кореляція між цими двома змінними є статистично значущою. Зазвичай ми вибираємо рівень значущості 0,05, що означає, що ми на 95% впевнені, що кореляція між змінними є значущою. Ми можемо отримати цю інформацію за допомогою модуля "stats" у бібліотеці "scipy".

В статистиці, кожна гіпотеза щодо невідомого розподілу {\displaystyle F}F випадкової величини {\displaystyle X}X називається статистичною гіпотезою. Якщо ми стверджуємо про одну гіпотезу і нашою метою є статистична перевірка чи є ця гіпотеза не хибною, але не маємо наміру, одночасно з тим, досліджувати іншу гіпотезу, тоді така перевірка називається

перевіркою значимості. Статистична гіпотеза, яка стосується лише числових значень невідомих параметрів певного розподілу називається параметричною гіпотезою. Методи перевірки статистичних гіпотез називаються статистичними тестами. Тести, що перевіряють параметричні гіпотези називаються параметричними тестами.

P-value застосовується у контексті перевірки нульової гіпотези для надання кількісної оцінки поняттю статистичної значущості доведення. Зауважте що статистична значущість результату не означає те, що результат так само має наукову значимість. Доведення нульової гіпотези це метод доведення до абсурду — аргументування, що прийняте у статистиці. По суті, твердження вважається правильним, якщо його протилежне твердження є неймовірним.

Таким чином, єдиною гіпотезою яку необхідно визначити при такій перевірці є протилежна гіпотеза, що називається нульовою гіпотезою (тобто гіпотеза, яка вважається не правдивою). Результат вважатиметься статистично значимим якщо нульову гіпотезу можна перевірено спростувати. Тобто іншими словами, при методі доведення до абсурду, для статистично значимого результату нульова гіпотеза матиме дуже малу ймовірність того, що вона є правдивою. Спростування нульової гіпотези означає, що правильна гіпотеза полягає в логічному доповнені до нульової гіпотези. Однак, якщо існує хоча б одна альтернатива нульовій гіпотезі, її спростування не може точно означати яка з інших альтернативних гіпотез є правдивою.

Якщо X є випадковою величиною, що представляє собою дані спостереження і H — статистична гіпотеза, що розглядається, тоді нотація статистичної значимості можна інтуїтивно визначити за допомогою умовної імовірності P(X|H), яка задає ймовірність спостереження за умови, що гіпотеза припускається правдивою. Однак, якщо X є неперервною випадковою величиною і спостерігається її реалізація x, P(X=x|H)=0. У такому застосуванні, інтуїтивне визначення є не адекватним і його необхідно змінити так, щоб воно відповідало неперервним випадковим величинам.

Дисперсійний аналіз

Дисперсійний аналіз — це статистичний метод, який використовується для перевірки того, чи існують суттєві відмінності між середніми показниками двох або більше груп. Повертає два параметри:

Оцінка F-test: передбачає, що середні значення для всіх груп однакові, обчислює, наскільки фактичні середні відхиляються від припущення, і

повідомляє це як оцінку F-test. Більший бал означає, що між середніми є більша різниця.

P-value: P-value вказує, наскільки статистично значуще наше обчислене значення.

Якщо наша змінна ціни сильно корелює зі змінною, яку ми аналізуємо, ми очікуємо, що дисперсійний аналіз поверне значну оцінку F-тесту та невелике значення p.

В будь-якому експерименті середні значення досліджуваних величин змінюються у зв'язку зі зміною основних факторів (кількісних та якісних), що визначають умови досліду, а також і випадкових факторів. Дослідження впливу тих чи інших факторів на мінливість середніх є задачею дисперсійного аналізу.

Дисперсійний аналіз використовує властивість адитивності дисперсії випадкової величини, що обумовлено дією незалежних факторів. В залежності від числа джерел дисперсії розрізняють однофакторний та багатофакторний дисперсійний аналіз.

Дисперсійний аналіз особливо ефективний при вивченні кількох факторів. При класичному методі вивчення змінюють тільки один фактор, а решту залишають постійними. При цьому для кожного фактору проводиться своя серія спостережень, що не використовується при вивченні інших факторів. Крім того, при такому методі досліджень не вдається визначити взаємодію факторів при одночасній їх зміні. При дисперсійному аналізі кожне спостереження служить для одночасної оцінки всіх факторів та їх взаємодії.

Дисперсійний аналіз полягає у виділенні й оцінюванні окремих факторів, що викликають зміну досліджуваної випадкової величини. При цьому проводиться розклад сумарної вибіркової дисперсії на складові, обумовлені незалежними факторами. Кожна з цих складових є оцінкою дисперсії генеральної сукупності. Щоб дати оцінку дієвості впливу даного фактору, необхідно оцінити значимість відповідної вибіркової дисперсії у порівнянні з дисперсією відтворення, обумовленою випадковими факторами. Перевірка значимості оцінок дисперсії проводять з допомогою критерію Фішера.

Коли розрахункове значення критерію Фішера виявиться меншим табличного, то вплив досліджуваного фактору немає підстав вважати значимим. Коли ж розрахункове значення критерію Фішера виявиться більшим табличного, то цей фактор впливає на зміни середніх. В подальшому ми вважаємо, що виконуються наступні припущення:

1. Випадкові помилки спостережень мають нормальний розподіл.
2. Фактори впливають тільки на зміну середніх значень, а дисперсія спостережень залишається постійною.

Фактори, що розглядаються в дисперсійному аналізі, бувають трьох родів:

- з випадковими рівнями, коли вибір рівнів проходить з безмежної сукупності можливих рівнів та супроводжується рандомізацією і рівні вибираються випадковим чином;
- з фіксованими рівнями;
- змішаного типу — частина факторів розглядається на фіксованих рівнях, але рівні решти вибираються випадковим чином.

Дисперсійний аналіз застосовується в різних формах в залежності від структури об'єкту, що досліджується; вибір відповідної форми є однією з головних трудностей в практичному застосуванні аналізу.

2.2. Порядок виконання роботи

1. Проаналізувати умову задачі.
2. Написати мовою Python програму для вирішення поставленої задачі.
3. Результати роботи оформити протоколом.

2.3. Завдання

1. Знайдіть кореляцію між такими стовпцями: bore, stroke, compression-ratio, and horsepower. (.corr())
2. Знайдіть кореляцію між x="stroke" and y="price".
3. Враховуючи результати кореляції між "stroke" та "price", чи очікуєте ви лінійної залежності?
4. Використовуйте функцію «groupby», щоб знайти середню "price" кожного автомобіля на основі "body-style".
5. Обчисліть коефіцієнт кореляції Пірсона та P-value для ‘wheel-base’ та ‘price’.
6. Обчисліть коефіцієнт кореляції Пірсона та P-value для ‘horsepower’ та ‘price’.
7. Зробіть дисперсійний аналіз декількох пар параметрів та напишіть висновки.

2.4. Контрольні запитання і завдання

1. Що таке кореляція?
2. Що таке дисперсійний аналіз?
3. Що таке P-value?
4. Що таке F-test?
5. Для чого використовується groupby?

Лабораторна робота №3
Model Development

Мета роботи: отримати навички розробки моделей для передбачення.

3.1. Теоретичні відомості

Завантажте пакет scikit-learn: https://scikit-learn.org/stable/install.html

Одним із прикладів моделі даних, яку ми будемо використовувати, є:

Проста лінійна регресія – це метод, який допомагає нам зрозуміти зв'язок між двома змінними:

Незалежна змінна (X)
Відповідь/залежна змінна (яку ми хочемо передбачити) (Y)
Результатом лінійної регресії є лінійна функція, яка прогнозує змінну відповіді (залежну) як функцію провісної (незалежної) змінної.

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
X = df[['highway-mpg']]
Y = df['price']
lm.fit(X,Y)
```

Що робити, якщо ми хочемо передбачити ціну автомобіля, використовуючи більше однієї змінної? Якщо ми хочемо використовувати більше змінних у нашій моделі для прогнозування ціни автомобіля, ми можемо використовувати множинну лінійну регресію. Множина лінійна регресія дуже схожа на просту лінійну регресію, але цей метод використовується для пояснення зв'язку між однією залежною і двома або більше предикторними (незалежними) змінними. Більшість реальних регресійних моделей включають кілька предикторів.

Графік регресії

Коли справа доходить до простої лінійної регресії, відмінним способом візуалізації відповідності нашої моделі є використання графіків регресії. Цей графік покаже комбінацію розсіяних точок даних (діаграма розсіювання), а також підігнану лінію лінійної регресії, що проходить через дані. Це дасть нам розумну оцінку зв'язку між двома змінними, силу кореляції, а також напрям (позитивна чи негативна кореляція).

Хорошим способом візуалізації дисперсії даних є використання графіка залишку. Різниця між спостережуваним значенням (y) і прогнозованим значенням (Y) називається залишком (e). Коли ми дивимося на графік регресії, залишок — це відстань від точки даних до встановленої лінії регресії.

Графік залишків – це графік, який показує залишки на вертикальній осі Y і незалежну змінну на горизонтальній осі X.

```
width = 12
height = 10
plt.figure(figsize=(width, height))
sns.residplot(df['highway-mpg'], df['price'])
plt.show()
```

Як зобразити модель для множинної лінійної регресії? Це стає дещо складнішим, тому що ви не можете візуалізувати це за допомогою регресії або залишкового графіка. Один із способів поглянути на підгонку моделі — це поглянути на графік розподілу. Ми можемо подивитися на розподіл підібраних значень, які є результатом моделі, і порівняти його з розподілом фактичних значень.

У статистиці лінійна регресія — це метод моделювання залежності між скалярною змінною y та векторною (у загальному випадку) змінною X. У разі, якщо змінна X також є скаляром, регресію називають простою.

При використанні лінійної регресії взаємозв'язок між даними моделюється за допомогою лінійних функцій, а невідомі параметри моделі оцінюються за вхідними даними. Подібно до інших методів регресійного аналізу лінійна регресія повертає розподіл умовної імовірності y в залежності від X, а не розподіл спільної імовірності y та X, що стосується області мультиваріативного аналізу.

При розрахунках параметрів моделі лінійної регресії зазвичай застосовується метод найменших квадратів (МНК), але також можуть бути використані інші методи. Але метод найменших квадратів може бути використаний і для нелінійних моделей, тому МНК та лінійна регресія, хоч і є тісно пов'язаними, але не є синонімами.

Поліноміальна регресія та конвеєри

Поліноміальна регресія є окремим випадком моделі загальної лінійної регресії або моделей множинної лінійної регресії. Ми отримуємо нелінійні зв'язки, зводячи в квадрат або встановлюючи члени вищого порядку змінних-провісників.

У статистиці, поліноміальна регресія є однією з форм регресійного аналізу, в якому залежність між незалежною змінною x і залежною змінною y моделюється як поліном від x ступеню n. Поліноміальна регресія відповідає нелінійній залежності між значенням x та відповідним умовним математичним сподіванням y, що позначається E(y|x). Хоча поліноміальна регресія налаштовує нелінійній моделі даних, з боку теорії оцінювання ця задача є лінійною, в тому сенсі, що функція регресії E(y| x) є лінійною за невідомих параметрів які оцінюються за даними. З цього приводу поліноміальна регресія вважається приватним випадком множинної лінійної регресії.

Пояснювальні (незалежні) змінні, що є результатом поліноміального розширення «базових» змінних, відомі як терміни вищого ступеня. Такі змінні також використовуються в налаштуваннях класифікації.

Хоча поліноміальна регресія технічно є частковим випадком багаторазової лінійної регресії, інтерпретація побудованої моделі поліноміальної регресії вимагає дещо іншої перспективи. Часто буває важко інтерпретувати окремі коефіцієнти в поліноміальній регресії, оскільки основні одночлени можуть бути високо корельованими. Наприклад, x та x2 мають кореляцію близько 0.97 коли x рівномірно розподіляється на інтервалі (0, 1). Хоча кореляцію можна зменшити за допомогою ортогональних поліномів, загалом більш інформативно розглядати побудовану функцію регресії в цілому. Поточкові або одночасні довірчі смуги потім можуть бути використані для забезпечення відчуття невизначеності в оцінці функції регресії.

Pipeline

Конвеєри даних спрощують етапи обробки даних. Для створення конвеєра ми використовуємо модуль Pipeline. Ми також використовуємо StandardScaler як крок у нашому конвеєрі.

Оцінюючи наші моделі, ми хочемо не тільки візуалізувати результати, а й кількісний показник, щоб визначити, наскільки точна модель.

Два дуже важливі показники, які часто використовуються в статистиці для визначення точності моделі:

Середня квадратична помилка (MSE)
R^2, також відомий як коефіцієнт детермінації, є мірою, яка вказує, наскільки близькі дані до підігнаної лінії регресії.

Значення R^2 – це відсоток варіації змінної відповіді (y), що пояснюється лінійною моделлю.

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

y_i - представляє фактичні значення,

$\hat{y}_i$ - представляє передбачення,

$\bar{y}_i$ - середнє значення всіх значень

Середня квадратична помилка вимірює середнє значення квадратів помилок. Тобто різниця між фактичним значенням (y) і розрахунковим (передбаченим) значенням (ŷ).

Прийняття рішень: визначення хорошої відповідності моделі

Якщо порівнювати моделі, то для даних краще підходить модель з більшим значенням R-квадрата.

При порівнянні моделей, модель з найменшим значенням MSE краще підходить для даних.

Давайте подивимося на значення для різних моделей.
Проста лінійна регресія: використання Highway-mpg як прогнозна змінна ціни.

R-квадрат: 0,49659118843391759
MSE: 3,16 x 10^7
Множина лінійна регресія: використання кінських сил, спорядженої маси, об’єму двигуна та пробігу по шосе як змінні ціни.

R-квадрат: 0,80896354913783497
MSE: 1,2 x10^7
Поліноміальна підгонка: використання Highway-mpg як прогнозна змінна ціни.

R-квадрат: 0,6741946663906514
MSE: 2,05 x 10^7

Проста модель лінійної регресії (SLR) проти моделі множинної лінійної регресії (MLR)
Зазвичай, чим більше у вас змінних, тим краще ваша модель прогнозує, але це не завжди вірно. Іноді у вас може бути недостатньо даних, ви можете зіткнутися з чисельними проблемами, або багато змінних можуть бути некорисними і навіть діяти як шум. У результаті ви завжди повинні перевіряти MSE та R^2.

Щоб порівняти результати моделей MLR і SLR, ми розглянемо комбінацію як R-квадрата, так і MSE, щоб зробити найкращий висновок щодо відповідності моделі.

MSE: MSE SLR становить 3,16x10^7, тоді як MLR має MSE 1,2 x10^7. MSE MLR набагато менший.
R-квадрат: у цьому випадку ми також бачимо, що існує велика різниця між R-квадратом SLR та R-квадратом MLR. R-квадрат для SLR (~0,497) дуже малий порівняно з R-квадрат для MLR (~0,809).
Цей R-квадрат у поєднанні з MSE показує, що MLR здається кращою моделлю в цьому випадку в порівнянні з SLR.

Проста лінійна модель (SLR) проти поліноміальної відповідності
MSE: Ми бачимо, що Polynomial Fit збив MSE, оскільки ця MSE менша, ніж у SLR.
R-квадрат: R-квадрат для Polynomial Fit більше, ніж R-квадрат для SLR, тому Polynomial Fit також значно збільшив R-квадрат.
Оскільки поліноміальна підгонка призвела до нижчого MSE і більшого R-квадрата, ми можемо зробити висновок, що це була краща модель, ніж проста лінійна регресія для прогнозування «ціни» з «highway-mpg» як провісною змінною.

Множина лінійна регресія (MLR) проти поліноміальної відповідності
MSE: MSE для MLR менший, ніж MSE для поліноміального підгонки.
R-квадрат: R-квадрат для MLR також набагато більший, ніж для Polynomial Fit.

У статистиці середньоквадрати́чна по́хибка, сере́дня квадрати́чна по́хибка (СКП, англ. mean squared error, MSE) або середньоквадрати́чне відхи́лення, сере́днє квадрати́чне відхи́лення (СКВ, англ. mean squared deviation, MSD) оцінювача (процедури оцінювання неспостережуваної величини) вимірює усереднення квадратів похибок — тобто, середнє квадратичної різниці між оцінками значень та справжнім значенням. СКП є функцією ризику, яка відповідає математичному сподіванню квадрату похибкових втрат. Той факт, що СКП є майже завжди строго додатною (а не нульовою), випливає з випадковості, або з того, що оцінювач не враховує інформації, яка могла би давати точнішу оцінку.

СКП є мірою якості оцінювача. Оскільки вона походить від квадрата евклідової відстані, її значення є завжди додатним, і зменшується, коли похибка наближається до нуля.

СКП є другим моментом похибки (відносно оригіналу) і таким чином, охоплює як дисперсію оцінювача (наскільки широким є розкид оцінок від одного зразка даних до іншого), так і його зміщення (наскільки віддаленим є усереднене оцінене значення від істинного). Для незміщеного оцінювача СКП є його дисперсією. Як і дисперсія, СКП має ті ж одиниці вимірювання, що й квадрат оцінюваної величини. За аналогією зі стандартним відхиленням, взяття квадратного кореня СКП дає кореневу середньоквадратичну похибку, або кореневе середньоквадратичне відхилення[en] (КСКП або КСКВ, англ. RMSE, RMSD), що має ті ж одиниці вимірювання, що й оцінювана величина. Для незміщеного оцінювача КСКП є квадратним коренем дисперсії, відомим як стандартна похибка.

3.2. Порядок виконання роботи

1. Проаналізувати умову задачі.
2. Написати мовою Python програму для вирішення поставленої задачі.
3. Результати роботи оформити протоколом.

3.3. Завдання

1. Навчіть лінійно-регресійну модель, використовуючи engine-size як незалежну змінну та price як залежну змінну? Чи вдала ця модель для даного випадку?
2. Створіть і навчіть модель множинної лінійної регресії, де змінною відповіді є price, а змінною-прогнозувальником є "normalized-losses" та "highway-mpg".
3. Створіть поліноміальну модель 11 порядку зі змінними x і y (залежну і незалежну змінні оберіть на ваш розсуд).
4. Побудуйте для деяких змінних на ваш розсуд усі моделі які ми розгянули, запишіть висновки.

3.4. Контрольні запитання

1. Що таке лінійна регресія?
2. Що таке множинна регресія?
3. Що таке R^2? MSE?
4. Як зрозуміти яка модель краще робить передбачення?
5. Як використати побудовану модель? І для потрібно її використовувати?

Лабораторна робота №4
Model Evaluation and Refinement

Мета роботи: отримати навички оцінки якості моделі.

4.1. Теоретичні відомості
Cross-Validation Score

```
from sklearn.model_selection import cross_val_score
Rcross = cross_val_score(lre, x_data[['horsepower']], y_data, cv=4)
```

Перехресне затверджування (англ. cross-validation), іноді зване ротаційним оцінюванням (англ. rotation estimation) або позавибірковим випробуванням (англ. out-of-sample testing), — це будь-яка з подібних методик затверджування моделі для оцінювання того, наскільки результати статистичного аналізу узагальнюватимуться на незалежний набір даних. Його переважно використовують в постановках, де метою є передбачування, й потрібно оцінювати те, наскільки точно передбачувальна модель працюватиме на практиці. В задачі передбачування, моделі зазвичай дають набір відомих даних, на яких виконують тренування (тренувальний набір даних), та набір невідомих даних (або вперше бачених даних), на яких модель випробовують (званий затверджувальним або випробувальним набором даних). Метою перехресного затверджування є випробувати здатність моделі передбачувати нові дані, які не використовувалися при її визначенні, щоби просигналізувати про такі проблеми як перенавчання та ви́біркове упередження, і щоби дати уявлення про те, як ця модель узагальнюватиметься на незалежний набір даних (тобто, невідомий набір даних, наприклад, з реальної задачі).

Один раунд перехресного затверджування включає розбивання вибірки даних на взаємодоповнювальні піднабори, виконання аналізу на одному з піднаборів (званому тренувальним набором) та затверджування результатів на іншому піднаборі (званому затверджувальним або випробувальним набором). З метою зниження мінливості, в більшості методів виконують декілька раундів перехресного затверджування з використанням різних розбиттів, і, щоби дати оцінку передбачувальної продуктивності моделі, результати затверджування поєднують (наприклад, усереднюють) над раундами.

Коротко, перехресне затверджування поєднує (усереднює) міри допасованості в передбачуванні, щоби вивести точнішу оцінку передбачувальної продуктивності моделі.

Overfitting, Underfitting and Model Selection

Виявилося, що тестові дані, які іноді називають «даними поза вибіркою», є набагато кращим показником того, наскільки добре ваша модель працює в реальному світі. Однією з причин цього є overfitting. Давайте розглянемо кілька прикладів. Виявляється, ці відмінності більш очевидні в множинній лінійній регресії та поліноміальній регресії, тому ми розглянемо overfitting в цьому контексті. Створіть об’єкти множинної лінійної регресії та потренуйте модель, використовуючи **'horsepower'**, **'curb-weight'**, **'engine-size'** and **'highway-mpg'** як фічі.

Перенавчання (overfitting) відбувається, коли модель відповідає шуму, але не відповідає основному процесу. Таким чином, під час тестування вашої моделі за допомогою тестового набору ваша модель працює не так добре, оскільки вона моделює шум, а не основний процес, який створив зв’язок.

Розглянемо регресію Ridge та побачимо, як параметр alpha змінює модель. Зауважте, тут наші тестові дані будуть використовуватися як дані перевірки. Виконаємо поліноміальне перетворення другого ступеня над нашими даними.

У статистиці та машинному навчанні одним із найпоширеніших завдань є допасовування «моделі» до набору тренувальних даних таким чином, щоби уможливити здійснення надійних передбачень на загальних даних, на яких не здійснювалося тренування. При перенавчанні (англ. overfitting) статистична модель описує випадкову похибку або шум, замість взаємозв'язку, що лежить в основі даних. Перенавчання виникає тоді, коли модель є занадто складною, такою, що має занадто багато параметрів відносно числа спостережень. Перенавчена модель має погану передбачувальну[en] продуктивність, оскільки вона занадто сильно реагує на другорядні відхилення в тренувальних даних.

Можливість перенавчання існує тому, що критерій, який застосовується для тренування моделі, відрізняється від критерію, який застосовується для оцінки її ефективності. Зокрема, модель зазвичай тренують шляхом максимізації її продуктивності на якомусь наборі тренувальних даних. Проте її ефективність визначається не її продуктивністю на тренувальних даних, а її здатністю працювати добре на даних небачених. Перенавчання стається тоді, коли модель починає «запам'ятовувати» тренувальні дані, замість того, щоби «вчитися» узагальненню з тенденції. Як крайній приклад, якщо число параметрів є таким же, або більшим, як число спостережень, то проста модель або процес навчання може відмінно передбачувати тренувальні дані, просто запам'ятовуючи їх повністю, але така модель зазвичай зазнаватиме рішучої

невдачі при здійсненні передбачень про нові або небачені дані, оскільки ця проста модель взагалі не навчилася узагальнювати.

Потенціал перенавчання залежить не лише від кількостей параметрів та даних, але й від відповідності структури моделі формі даних, та величини похибки моделі в порівнянні з очікуваним рівнем шуму або похибки в даних.

Навіть коли допасована модель не має надмірного числа параметрів, слід очікувати, що допасований взаємозв'язок працюватиме на новому наборі даних не так добре, як на наборі, використаному для допасовування. Зокрема, значення коефіцієнту детермінації відносно первинних тренувальних даних скорочуватиметься.

Щоби уникати перенавчання, необхідно використовувати додаткові методики (наприклад, перехресне затверджування, регуляризацію, ранню зупинку, обрізку, баєсові апріорні параметрів або порівняння моделей), які можуть вказувати, коли подальше тренування не даватиме кращого узагальнення. Основою деяких методик є або явно штрафувати занадто складні моделі, або перевіряти здатність моделі до узагальнення шляхом оцінки її продуктивності на наборі даних, не використаному для тренування, який вважається наближенням типових небачених даних, з якими стикатиметься модель.

Гарною аналогією перенавчання задачі є уявити дитину, яка намагається вивчити, що є вікном, а що не є вікном, ми починаємо показувати їй вікна, і вона виявляє на початковому етапі, що всі вікна мають скло та раму, і через них можна дивитися назовні, деякі з них може бути відчинено. Якщо ми продовжимо показувати ті самі вікна, то дитина може також зробити помилковий висновок, що всі вікна є зеленими, і що всі зелені рами є вікнами. Перенавчаючись таким чином цієї задачі.

```
from sklearn.linear_model import Ridge
pr=PolynomialFeatures(degree=2)
x_train_pr=pr.fit_transform(x_train[['horsepower', 'curb-
weight', 'engine-size', 'highway-mpg','normalized-
losses','symboling']])
x_test_pr=pr.fit_transform(x_test[['horsepower', 'curb-
weight', 'engine-size', 'highway-mpg','normalized-
losses','symboling']])
RigeModel=Ridge(alpha=1)
RigeModel.fit(x_train_pr, y_train)
```

Grid Search

Термін альфа є гіперпараметром. Sklearn має клас GridSearchCV, щоб спростити процес пошуку найкращого гіперпараметра.

```
from sklearn.model_selection import GridSearchCV
Grid1 = GridSearchCV(RR, parameters1, cv=4)
```

4.2. Порядок виконання роботи

1. Проаналізувати умову задачі.
2. Написати мовою Python програму для вирішення поставленої задачі.
3. Результати роботи оформити протоколом.

4.3. Завдання

1. Власноруч потренуйте модель таким чином, щоб для трейні даних вона працювала ідеально, а для тестових погано, побудуйте графік щоб візуально показати Overfitting.
2. Те саме що в пункті 1 тільки Underfitting.
3. Використовуючи GridSearch знайдіть для Ridge Regression гіперпараметр, для довільних фіч датасету.

4.4. Контрольні запитання

1. Що таке Overfitting, Underfitting?
2. Що таке GridSearch?
3. Що таке Cross-Validation Score? Для чого він потрібен?

Список літератури
References

1. https://www.coursera.org/learn/introduction-to-data-analytics?specialization=ibm-data-analyst
2. A. Shelestov, B. Yailymov, H.Yailymova, L. Shumilo, M. Lavreniuk Advanced Method of Land Cover Classification Based on High Spatial Resolution Data and Convolutional Neural Network. Proceedings of International Conference on Applied Innovation in IT. Volume 10, Issue 1. pp. 125-132. doi:10.25673/76943.
3. Nataliia Kussul, Hanna Yailymova, Sophia Drozd, Andrii Shelestov Validation of the global human settlement layer and NASA population data for Ukraine. 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. 2021. Cracow (virtual format). P. 288 – 291. DOI: 10.1109/IDAACS53288.2021.9661062.
4. Leonid Shumilo, Mykola Lavreniuk, Sergii Skakun, Nataliia Kussul Is Soil Bonitet an Adequate Indicator for Agricultural Land Appraisal in Ukraine? Міжнародний науковий журнал «Sustainability» видавництва MDPI. 2021. N0. 13, 12096. doi: 10.3390/su132112096.
5. Andrii Shelestov, Hanna Yailymova, Bohdan Yailymov, Oleg Samoilenko, Leonid Shumilo Ground Based Validation of Copernicus Atmosphere Monitoring Service Data for Kyiv. 2021 IEEE 19th International Conference on Smart Technologies (EUROCON). 2021. Lviv (virtual format). pp. 88-91, doi: 10.1109/EUROCON52738.2021.9535629.
6. Mikhail Emelyanov, Hanna Yailymova, Andrii Shelestov, Bohdan Yailymov Intellectual Analysis of Major Crops Area due to Climate Changes in Ukraine. 2021 IEEE 19th International Conference on Smart Technologies (EUROCON). 2021. Lviv (virtual format). pp. 192-196, doi: 10.1109/EUROCON52738.2021.9535607.
7. M. Hosseini et. All. A Comparison between Support Vector Machine and Water Cloud Model for Estimating Crop Leaf Area Index. Remote Sensing. 2021. Vol. 13, No. 7. P. 1-20. DOI: 10.3390/rs13071348
8. N. Kussul, K. Deininger, L. Shumilo, M. Lavreniuk, D. Ayalew Ali., O. Nivievskyi Biophysical Impact of Sunflower Crop Rotation on Agricultural Fields. Sustainability. 2022. No. 14(7):3965. pp. 125-132. https://doi.org/10.3390/su14073965.
9. Molin, S. (2019). Hands-On Data Analysis with Pandas: Efficiently perform data collection, wrangling, analysis, and visualization using Python. Packt Publishing Ltd.