Лабораторна робота 2
Тема: Задача класифікації

Мета роботи

Мета роботи – ознайомитися із задачею класифікації даних, опанувати процес навчання класифікаторів і оцінювання їхньої якості на реальному наборі даних, а також вивчити можливості бібліотеки scikit-learn для розв'язання такої задачі.

Постановка завдання

I. Ознайомтеся з теоретичним матеріалом, що стосується задачі класифікації, алгоритмів класифікації (логістична регресія, правило k найближчих сусідів тощо), метрик оцінювання якості класифікації, понять навчальної, валідаційної та тестової вибірок, методів ковзного контролю, перенавчання та регуляризації.

II. Ознайомтеся з можливостями бібліотеки scikit-learn для розв'язання задачі класифікації (див. відеозаписи лабораторних занять та подані нижче рекомендації щодо класів і функцій, які можуть знадобитися в роботі).

III. Використовуючи бібліотеку scikit-learn, навчіть два класифікатори,і оцініть їх якість та порівняйте результати. Використовуйте набір даних, заданий в лабораторній роботі 1, або за бажанням свій власний. У процесі виконання роботи рекомендується дотримуватися такого плану:

1. Підготовка даних:
   - Переконайтеся, що у наборі даних немає пропусків.
   - Бінаризуйте або закодуйте якісні ознаки (залежно від того, чи вони номінальні, чи порядкові).
   - Масштабування кількісних ознак є необов'язковим, але його можна виконати, якщо це вважається доцільним для подальшого навчання класифікаторів.
2. Перевірка балансу класів:
   - Визначте, чи збалансовані класи.
   - Виконання балансування класів не є обов'язковим у межах цієї роботи, але якщо класи незбалансовані, опишіть словами можливі наслідки цього, доступні методи балансування та вкажіть, які метрики якості є більш доцільними у такому випадку.
3. Вибір метрики:
   - Оберіть одну або кілька метрик для оцінки якості класифікації (наприклад, accuracy, precision, recall, f1-score), зважаючи на специфіку вашої задачі.
   - Перелік метрик, доступних в sklearn.metrics, можна переглянути за посиланням https://scikit-learn.org/stable/modules/model_evaluation.html#scoring-parameter.
4. Розділення даних:
   - Розділіть набір даних на навчальну та тестову вибірки за допомогою функції train_test_split з модуля sklearn.model_selection.
5. Навчання логістичної регресії без регуляризації:
   - Навчіть модель логістичної регресії без регуляризації на навчальній вибірці.
   - Якщо виникнуть проблеми, спробуйте налаштувати параметр solver або виконати масштабування кількісних ознак.
   - З'ясуйте, Ви будуєте бінарну чи мультиноміальну модель.
   - Виведіть параметри моделі, поясніть їх кількість.
6. Оцінка якості моделі:
   - Розрахуйте обрані материки якості на навчальній та тестовій вибірках.
   - Оцініть ризик перенавчання, порівнюючи значення метрик якості на навчальній та тестовій вибірках.
   - Проаналізуйте матрицю помилок, щоб оцінити, чи добре модель класифікує кожен клас.

7. Оцінка якості моделі на основі перехресної перевірки (на практиці це є альтернативою попередньому пункту):
   - Використайте k-кратну перехресну перевірку для оцінки якості моделі, скориставшись, наприклад, функцією cross_validate або cross_val_score з sklearn.model_selection. Розберіться із параметрами cv та scoring в цих функціях.
   - Розгляньте різну кількість фолдів та проаналізуйте вплив цього параметра на стабільність оцінки.
   - Порівняйте результати перехресної перевірки з результатами на тестовій вибірці.
8. Навчання логістичної регресії з регуляризацією:
   - Використайте регуляризацію (L_2 або L_1 за власним бажанням). L_2 регуляризація допомагає зменшити перенавчання, приглушуючи коефіцієнти моделі, в той час як L_1 не лише сприяє боротьбі із перенавчанням, але й дозволяє виконувати відбір ознак шляхом стиснення деяких коефіцієнтів до нуля.
   - Підберіть оптимальне значення параметра регуляризації C за допомогою таких підходів:
     - «річне» розбиття навчальної вибірки на навчальну і валідаційну та перебір гіперпараметрів;
     - побудова кривої навчання за допомогою функції validation_curve; зверніть увагу на її аргументи scoring та cv;
     - використання GridSearchCV або RandomizedSearchCV для автоматизації підбору гіперпараметрів.
   - Поясніть, чому тестова вибірка не використовується для підбору гіперпараметрів.
9. Оцінка якості моделі з регуляризацією:
   - Розрахуйте метрики для моделі з оптимальним значенням гіперпараметра на навчальній та тестовій вибірках.
   - Поясніть, чи допомогла регуляризація зменшити перенавчання.
10. (Опційно) Відбір ознак:
    - Спробуйте здійснити відбір інформативних ознак, навчити модель на їх основі та оцінити її якість.
    - Цей пункт є необов'язковим. Його можна виконати, якщо є бажання.
11. Навчання додаткового класифікатора:
    - Оберіть інший класифікатор (наприклад, k-NN чи випадковий ліс).
    - Підберіть оптимальні значення гіперпараметрів та оцініть якість моделі:
      - зверніть увагу, що для k-NN слід масштабувати кількісні ознаки;
      - для випадкового лісу врахуйте параметри, які зменшують ризик перенавчання.
12. Порівняння класифікаторів:
    - Оцініть якість моделей на тестовій вибірці.
    - Порівняйте результати, вкажіть сильні та слабкі сторони кожного класифікатора.

За результатами виконання роботи **сформуйте ноутбук *.ipynb**, який має містити код для побудови класифікаторів і оцінки якості їх прогнозів, одержані результати, Ваші пояснення і коментарі.

Під час захисту роботи Ви повинні демонструвати вміння використовувати функції та класи scikit-learn, пояснювати отримані результати та знати теорію, яка лежить в основі алгоритмів та методів оцінки якості класифікації.

Набір даних

Для опрацювання пропонується набір даних https://archive.ics.uci.edu/ml/datasets/Adult (файл adult.data). За бажанням може бути використаний власний набір даних, або будь-який набір з сайту https://archive.ics.uci.edu або https://www.kaggle.com.

Рекомендації щодо використання бібліотеки scikit-learn

Бібліотека scikit-learn дозволяє працювати з наступними класифікаторами, кожен з яких реалізований окремим класом:

- логістична регресія (Logistic Regression) – клас LogisticRegression з модуля sklearn.linear_model;
- класифікатор *k* найближчих сусідів (*k* Nearest Neighbor, *k*NN) – клас KNeighborsClassifier з модуля sklearn.neighbors;
- машина опорних векторів (Support Vector Machine, SVM) – класи LinearSVC та SVC з модуля sklearn.svm;
- дерево рішень (Decision Tree) – клас DecisionTreeClassifier з модуля sklearn.tree;
- ліс випадкових дерев (Random Forest) – клас RandomForestClassifier з модуля sklearn.ensemble;
- наївний байєсовський класифікатор (Naïve Bayes) – клас GaussianNB (якщо усі ознаки кількісні неперервні), MultinomialNB (якщо ознаки кількісні дискретні), BernoulliNB (коли ознаки бінарні) або CategoricalNB (коли ознаки якісні номінальні з довільною кількістю категорій) з модуля naive_bayes;
- лінійний дискримінант (Linear Discriminant Analysis, LDA) – клас LinearDiscriminantAnalysis з модуля discriminant_analysis;
- квадратичний дискримінант (Quadratic Discriminant Analysis, QDA) – клас QuadraticDiscriminantAnalysis з модуля discriminant_analysis.

Кожен з перелічених вище класів бібліотеки scikit-learn має такі важливі методи:

- fit(X, y) – виконує навчання класифікатора на основі навчальної вибірки (X, y);
- predict(X) – повертає прогнозовані класи для об’єктів тренувальної вибірки X;
- predict_proba(X) – повертає імовірності належності об’єктів тренувальної вибірки X до кожного з класів;
- get_params() – повертає значення гіперпараметрів класифікатора;
- set_params(**params) – встановлюються нові значення гіперпараметрів, після цього необхідно заново навчити класифікатор, викликавши метод fit().

Також можна звернути увагу на алгоритм під назвою градієнтний бустінг (Gradient Boosting), який дозволяє будувати ансамбль дерев класифікації. Для роботи з ним знадобиться окрема бібліотека xgboost, і в ній клас XGBClassifier.

Для розрахунку метрик якості та матриці помилок (confusion matrix) можна використовувати відповідні функції з модуля sklearn.metrics.

З метою розбити набір даних на тренувальну та тестову частини можна скористатися функцією train_test_split з модуля sklearn.model_selection.

Для оцінювання якості класифікації на основі ковзного контролю корисними будуть класи групи Splitter Classes з модуля sklearn.model_selection, а також функції cross_validate або cross_val_score того ж модуля.

Під час вибору оптимальних значень гіперпараметрів може бути використана функція validation_curve з модуля sklearn.model_selection. Наприклад, на її основі може бути побудована крива валідації для вибору найкращого значення *k* для класифікатора *k* найближчих сусідів, параметру регуляризції для логістичної регресії, параметру C для машини опорних векторів, кількості дерев у випадковому лісі, тощо.

Для автоматичного вибору значень гіперпараметрів можна використати класи GridSearchCV або RandomizedSearchCV з модуля sklearn.model_selection.

Для побудови кривої навчання (learning curve) доцільно скористатися функцією learning_curve з модуля sklearn.model_selection.

Рекомендована література

Матеріали для вивчення теоретичних засад машинного навчання:

1. Bishop C.M. Pattern Recognition and Machine Learning. Springer, 2006. 738 p.
2. Burkov A. Machine Learning Engineering. 2020. 310 p.
3. Burkov A. The Hundred-Page Machine Learning Book. 2022. 160 p.
4. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Data Mining, Inference, and Prediction. 2009. 745 p.
5. Мацуга О.М., Архангельська Ю.М., Єрещенко Н.М. Навчальний посібник до вивчення курсу «Інформаційні технології розпізнавання образів». Д.: РВВ ДНУ, 2016. 60 с.

Матеріали для вивчення мови Python:

6. Lutz M. Learning Python. O'Reilly, 2013. 1643 p.

Матеріали для знайомства з бібліотеками Python для машинного навчання:

7. Albon C. Machine Learning with Python Cookbook. O'Reilly, 2018. 366 p.
8. Müller A., Guido S. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly, 2016. 398 p.
9. Raschka S., Mirjalili V. Python Machine Learning. Packt Publishing. 2019. 770 p.
10. VanderPlas J. Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly, 2017. 546 p.
11. scikit-learn.org/stable/index.html – документація бібліотеки scikit-learn.