Projekt – Naiwny Klasyfikator Bayesowski

Ogólne informacje

Akceptowane języki programowania

Projekt może być realizowany w Pythonie lub R.

Zespoły

Projekt powinien być realizowany w zespołach dwuosobowych. Każdy zespół powinien mieć repozytorium na GitHubie, wymagane jest przesłanie linku do repozytorium i pliku zip zawierającego wszystkie pliki.

Struktura rozwiązania

Repozytorium powinno zawierać: plik z implementacją naiwnego klasyfikatora bayesowskiego, skrypt lub notatnik z wstępną analizą danych i skrypt/notatnik z oceną jakości modelu. Opis rozwiązania należy umieścić w przejrzysty sposób w pliku README (w języku polskim lub angielskim).

Kryteria oceny

Projekt będzie oceniany na podstawie jakości kodu, poprawności implementacji algorytmu i jakości analizy danych (EDA, ocena modelu).

Zbiór Danych

Dla potrzeb projektu zostaną użyte zbiory danych *Mushroom Classification* i *Iris*. Pierwszy zbiór zawiera informacje o różnych cechach grzybów, które mogą posłużyć do ich klasyfikacji jako jadalne lub trujące. Drugi zawiera informacje o 150 kwiatach z trzech gatunków irysów (setosa, versicolor, virginica), opisanych przez cztery cechy ilościowe: długość i szerokość kielicha oraz długość i szerokość płatka, używane do klasyfikacji gatunku.

Zbiór *Iris* w Pythonie jest dostępny w bibliotece scikit-learn. Zbiór *Mushroom Classification* jest dostępny na platformie Kaggle.

Cel

Celem projektu jest implementacja naiwnego klasyfikatora bayesowskiego.

Materiały

- Wykład (4. prezentacja: *zmienne-nd* strony 81-85).
- StatQuest: multinomial, gaussian.
- Dużo dostępnych materiałów na Kaggle'u na przykład notatnik.
- Naiwny klasyfikator bayesowski w sckit-learn.
- Przykład wykorzystania klasyfikatora z scikit-learn.

Kroki Realizacji Projektu

1. **Zrozumienie Naive Bayes Classifier**: Przejrzyj dostępne materiały dotyczące naiwnego klasyfikatora Bayesowskiego.
2. **Wstępna analiza danych**: Przykładowe pytania dla zbioru *Mushroom*:
   - Czy występują brakujące wartości?
   - Jaka jest proporcja grzybów jadalnych do trujących?
   - Jakie wartości przyjmują poszczególne cechy? Które wartości cech występują najczęściej?
   - Jakie cechy wyróżniają grzyby trujące od jadalnych? Które cechy są szczególnie ważne dla klasyfikacji?
   - Czy istnieją wyraźne wzorce lub grupy w przestrzeni cech, które mogą sugerować podział na klasy *jadalny* i *trujący*?
3. **Implementacja klasy NaiveBayesClassifier**: Zaimplementuj klasę `NaiveBayesClassifier` z metodami (w przypadku R powinna to być klasa S4):
   - `fit` - do trenowania modelu na zbiorze treningowym.
   - `predict` - do przewidywania klasy dla nowych danych.
   - `predict_proba` - do zwracania prawdopodobieństw przynależności do każdej klasy.
4. **Podział Zbioru Danych**: Podziel zbiór danych na część treningową i testową (70:30). Większa część będzie użyta do trenowania modelu, a mniejsza do oceny jego skuteczności. Do podziału na zbiór treningowy i testowy można użyć funkcji `train_test_split` (dostępnej w Pythonie w bibliotece scikit-learn).
5. **Trenowanie Modelu**: Użyj części treningowej zbioru danych do trenowania modelu.
6. **Ewaluacja Modelu**: Przetestuj działanie modelu na zbiorze testowym. Oblicz *accuracy*, czyli procent poprawnych odpowiedzi.

7. **Wsparcie dla Różnych Rodzajów Cech**: Wszystkie cechy w zbiorze danych *Mushroom* to zmienne kategoryczne, natomiast wszystkie cechy w zbiorze *Iris* to zmienne ilościowe. Implementacja powinna wspierać oba typy zmiennych (najprościej zaimplementować dwie osobne klasy MulitnomialNaiveBayesClassifier i GaussianNaiveBayesClassifier).

Opis Algorytmu

Naiwny klasyfikator bayesowski to algorytm uczenia maszynowego oparty na twierdzeniu Bayesa. Jego celem jest przypisanie zestawowi cech $\{x_1, x_2, \ldots, x_n\}$ najbardziej prawdopodobnej klasy c_j. Klasyfikator ten stosuje regułę decyzyjną *maximum a posteriori* (MAP), co oznacza, że wybiera klasę, dla której wartość prawdopodobieństwa (a posteriori) $p(c_j \mid x_1, x_2, \ldots, x_n)$ jest największa.

Twierdzenie Bayesa

Twierdzenie Bayesa można zapisać jako:

$$p(c_j \mid x_1, x_2, \ldots, x_n) = \frac{p(c_j) \cdot p(x_1, x_2, \ldots, x_n \mid c_j)}{p(x_1, x_2, \ldots, x_n)}$$

gdzie:

- $p(c_j \mid x_1, x_2, \ldots, x_n)$ - prawdopodobieństwo a posteriori klasy c_j przy założeniu cech $x_1, x_2, \ldots, x_n$,
- $p(c_j)$ - prawdopodobieństwo aprioryczne klasy c_j,
- $p(x_1, x_2, \ldots, x_n \mid c_j)$ - prawdopodobieństwo warunkowe zestawu cech przy założeniu klasy c_j,
- $p(x_1, x_2, \ldots, x_n)$ - prawdopodobieństwo wystąpienia zestawu cech $x_1, x_2, \ldots, x_n$ (jest stałe dla wszystkich klas).

Założenie niezależności cech

W klasyfikatorze naiwnym zakładamy, że każda cecha x_i jest warunkowo niezależna od pozostałych cech przy danej klasie c_j. Dzięki temu uproszczeniu możemy wyrazić $p(x_1, x_2, \ldots, x_n \mid c_j)$ jako iloczyn indywidualnych prawdopodobieństw warunkowych:

$$p(x_1, x_2, \ldots, x_n \mid c_j) = \prod_{i=1}^{n} p(x_i \mid c_j)$$

Reguła Decyzyjna

Zamiast obliczać dokładne prawdopodobieństwo a posteriori, wystarczy porównać wartości $p(c_j) \cdot \prod_{i=1}^{n} p(x_i \mid c_j)$ dla każdej klasy c_j i wybrać klasę, dla której ta wartość jest największa:

$$\hat{c} = \underset{c_j}{\operatorname{argmax}}\, p(c_j) \cdot \prod_{i=1}^{n} p(x_i \mid c_j)$$

gdzie $\hat{c}$ - przewidywana klasa dla zestawu cech $x_1, x_2, \ldots, x_n$.

Cechy Ilościowe i Kategoryczne

Naiwny klasyfikator bayesowski różni się w podejściu do obliczania prawdopodobieństw, w zależności od tego, czy zmienne są kategoryczne, czy ilościowe.

1. Naiwny Klasyfikator Bayesowski dla Zmiennych Kategorycznych

Dla zmiennych kategorycznych (takich jak kolor, typ, forma, itp.), naiwny klasyfikator bayesowski oblicza prawdopodobieństwa wystąpienia każdej wartości cechy dla danej klasy poprzez zliczenie liczby wystąpień danej wartości cechy w danej klasie.

Przykład: Załóżmy, że mamy zmienną kategoryczną x_i „kolor" z możliwymi wartościami „czerwony", „zielony" i „niebieski" oraz klasy c_j takie jak „jadalny" i „trujący". $p(x_i = \text{czerwony} \mid c_j = \text{jadalny})$ obliczamy jako stosunek liczby grzybów jadalnych o czerwonym kolorze do liczby wszystkich jadalnych grzybów. Ponieważ wartości zmiennej kategorycznej są skończone i przyjmują konkretne wartości, rozkład zmiennej jest modelowany poprzez zliczenie częstości wystąpień, co jest prostą metodą i dobrze sprawdza się w przypadku tego typu zmiennych.

2. Naiwny Klasyfikator Bayesowski dla Zmiennych Ilościowych (Ciągłych)

Dla zmiennych ilościowych (ciągłych), takich jak długość, szerokość, masa, itp., metoda zliczania wystąpień nie jest praktyczna. W takim przypadku wykorzystujemy rozkład prawdopodobieństwa dla zmiennej ciągłej, najczęściej **rozkład normalny**.

Rozkład Normalny Załóżmy, że dla danej klasy c_j, zmienna ilościowa x_i przyjmuje wartość zbliżoną do średniej μ_{c_j,x_i} z pewnym odchyleniem standardowym σ_{c_j,x_i}.

Prawdopodobieństwo $p(x_i \mid c_j)$ dla wartości ciągłej x_i można wyrazić za pomocą funkcji gęstości rozkładu normalnego:

$$p(x_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{c_j,x_i}^2}} \exp\left(-\frac{(x_i - \mu_{c_j,x_i})^2}{2\sigma_{c_j,x_i}^2}\right)$$

gdzie:

- μ_{c_j,x_i} to średnia wartość zmiennej x_i w klasie c_j,
- σ_{c_j,x_i} to odchylenie standardowe zmiennej x_i w klasie c_j.

Przykład: Załóżmy, że mamy zmienną ilościową „średnica kapelusza" dla grzybów w klasach „jadalny" i „trujący". Wówczas:

- Dla klasy „jadalny" obliczamy średnią $\mu_{\text{jadalny},x_{\text{średnica}}}$ oraz odchylenie standardowe $\sigma_{\text{jadalny},x_{\text{średnica}}}$ średnicy kapelusza.
- Następnie, dla nowego grzyba z daną średnicą kapelusza $x_{\text{średnica}}$, obliczamy $p(x_{\text{średnica}} \mid \text{jadalny})$ używając wzoru rozkładu normalnego.
- Podobnie obliczamy prawdopodobieństwo dla klasy „trujący" z odpowiednimi parametrami średnicy kapelusza.

Dzięki temu uzyskujemy prawdopodobieństwa dla różnych wartości zmiennych ciągłych w różnych klasach, co pozwala efektywnie klasyfikować dane przy użyciu naiwnego klasyfikatora bayesowskiego.

Podsumowanie

- **Zmienna kategoryczna**: Prawdopodobieństwa $p(x_i \mid c_j)$ są obliczane przez zliczanie wystąpień dla każdej kategorii.
- **Zmienna ilościowa**: Prawdopodobieństwa $p(x_i \mid c_j)$ są modelowane za pomocą funkcji gęstości prawdopodobieństwa rozkładu normalnego, który uwzględnia średnią i odchylenie standardowe dla każdej zmiennej w każdej klasie.

Wykorzystanie rozkładu normalnego dla zmiennych ilościowych umożliwia obliczenie prawdopodobieństwa dla dowolnej wartości ciągłej, co czyni klasyfikator bardziej elastycznym i zdolnym do pracy z różnymi typami danych.

Zasady Wykorzystania Dużych Modeli Językowych (LLM)

W ramach realizacji projektu dopuszcza się ograniczone wykorzystanie dużych modeli językowych (LLM) do wsparcia procesu nauki i implementacji, jednak należy stosować się do określonych zasad, aby zapewnić uczciwość akademicką oraz samodzielność pracy.

Dopuszczalne Zastosowania LLM

Studenci mogą korzystać z modeli językowych w następujących przypadkach:

- **Poprawa i recenzja kodu**: LLM mogą być używane do analizy oraz recenzji napisanego kodu, aby zidentyfikować potencjalne błędy i sugerować optymalizacje.
- **Debuggowanie i poszukiwanie błędów w implementacji**: Można korzystać z modeli językowych do pomocy w debuggowaniu, diagnozowaniu błędów oraz w zrozumieniu, dlaczego kod nie działa zgodnie z oczekiwaniami.
- **Zrozumienie algorytmu**: Modele językowe mogą być używane do wyjaśniania koncepcji związanych z klasyfikacją bayesowską lub innymi zagadnieniami teoretycznymi, aby ułatwić studentom zrozumienie algorytmu i jego działania.
- **Recenzja tekstu**: Możliwe jest korzystanie z LLM w celu recenzji tekstu, np. sprawdzenia spójności i jasności opisu, korekty stylistycznej oraz poprawy gramatycznej.

Niedopuszczalne Zastosowania LLM

Zabrania się wykorzystywania dużych modeli językowych do generowania całej implementacji algorytmu lub jego kluczowych elementów. Oczekuje się, że studenci samodzielnie zrealizują implementację algorytmu oraz wszystkie kluczowe kroki projektu.

Etyka Akademicka

Korzystanie z dużych modeli językowych jest dopuszczalne jedynie w ramach wsparcia i nauki. Każde użycie LLM powinno być transparentne, a wygenerowane treści powinny zostać zrozumiane, sprawdzone i dostosowane przez studentów przed ich finalnym włączeniem do projektu. Niedopuszczalne jest całkowite poleganie na LLM w celu realizacji projektu, gdyż może to prowadzić do naruszenia zasad etyki akademickiej.

Uwaga: Każde naruszenie powyższych zasad może skutkować obniżeniem oceny lub innymi konsekwencjami akademickimi.