Uczenie Maszynowe
Laboratorium 3: Metody Bayesowskie

1 Cele laboratorium

- Praktyczne zapoznanie się z Naiwnym Klasyfikatorem Bayesa - własna implementacja i testy na standardowych zbiorach danych
- Implementacja oraz testowanie Liniowej Regresji Bayesowskiej w trybie online

2 Literatura

- *Pattern Recognition and Machine Learning*, Christopher M. Bishop, Springer 2006.
- Bayesian linear regression for practitioners (M. Halford): `https://maxhalford.github.io/blog/bayesian-linear-regression/`

3 Przykładowe dane

- Iris dataset (Scikit Learn)
- Breast cancer wisconsin dataset (Scikit Learn)
- 20 news groups dataset
- Kaggle Adult Income dataset (`https://www.kaggle.com/datasets/wenruliu/adult-income-dataset`) - dwie klasy dla rocznego przychodu: $\leqslant 50k$ USD, $> 50k$ USD

4 Przydatne biblioteki i funkcje

1. SciKit Learn:
   - `load_iris()`
   - `load_wine()`
   - `train_test_split()`
   - `KFold`

- `fetch_openml`
- `cross_val_score()`, `confusion_matrix()`, `f1_score()`

2. Seaborn `https://seaborn.pydata.org`

5 Gaussowski Naiwny Klasyfikator Bayesa

$$p(\mathcal{C}_k|x_1,\ldots,x_n) = \frac{p(x_1|\mathcal{C}_k)p(x_2|\mathcal{C}_k)\ldots p(x_n|\mathcal{C}_k)p(\mathcal{C}_k)}{p(x_1)p(x_2)\ldots p(x_n)} \tag{1}$$

$$p(\mathcal{C}_k|x_1,\ldots,x_n) = \frac{p(\mathcal{C}_k)\prod_{i=1}^{n} p(x_i|\mathcal{C}_k)}{\prod_{i=1}^{n} p(x_i)} \tag{2}$$

$$p(\mathcal{C}_k|x_1,\ldots,x_n) \propto p(\mathcal{C}_k)\prod_{i=1}^{n} p(x_i|\mathcal{C}_k) \tag{3}$$

$$\mathcal{C} = \underset{\mathcal{C}_k}{\operatorname{argmax}} \left\{ p(\mathcal{C}_k)\prod_{i=1}^{n} p(x_i|\mathcal{C}_k) \right\} \tag{4}$$

po zastosowaniu logarytmu:

$$\mathcal{C} = \underset{\mathcal{C}_k}{\operatorname{argmax}} \left\{ \log p(\mathcal{C}_k) + \sum_{i=1}^{n} p(x_i|\mathcal{C}_k) \right\} \tag{5}$$

1. Zaimplementuj Naiwny Klasyfikator Bayesa dla danych ciągłych zakładając normalny rozkład prawdopodobieństwa dla każdej z cech z osobna. W implementacji możesz użyć standardowego interfejsu `scikit-learn` dla modeli predykcyjnych - metody `fit()` oraz `predict()`.

$$p(x_i|\mathcal{C}_k) = \frac{1}{\sqrt{2\pi\sigma_{i,\mathcal{C}_k}^2}} \exp\left\{ -\frac{(x_i - \mu_{i,\mathcal{C}_k})^2}{2\sigma_{i,\mathcal{C}_k}^2} \right\} \tag{6}$$

2. Wyznacz $\mu_{i,\mathcal{C}_k}$, $\sigma_{i,\mathcal{C}_k}^2$ - średnią i odchylenie standardowe ciągłej cechy x_i dla danej klasy $\mathcal{C}_k$, a następnie oblicz prawdopodobieństwa posterior korzystając z Twierdzenia Bayesa oraz wiarygodności danej wzorem Eq. 6.

3. Przetestuj działanie własnej implementacji klasyfikatora dla zbioru danych *Iris* (4 cechy). Zastosuj losowy podział zbioru danych na część trenignową i testową według proporcji 0.6, 0.4. Powtórz eksperyment 20-krotnie i zmierz średnie wartości:

   - *accuracy*
   - *f1-score*
   - *precision*

   Porównaj wyniki uzyskane z zastosowanie własnej implementacji oraz wyniki uzyskane za pomocą `scikit-learn GaussianNB`.

4. Przetestuj działanie klasyfikatora dla zbioru danych *Breast Cancer* (30 cech). Zastosuj trzy podane wyżej miary jakości klasyfikacji. Jako zbiór testowy wykorzystaj 0.3 dostępnego zbioru danych. Zbadaj wpływ:

   - skalowania (`StandardScaler`)
   - redukcji wymiaru (PCA i Kernel PCA, dobierz najlepszą wartość `n_components` oraz funkcję jądra na Kernel PCA)
   - transformacji Box-Cox (dobierz najlepszą wartość parametru λ)

   na średnie wyniki klasyfikacji. Czy tego typu wstępne przetwarzanie zbioru danych wpływa na ich poprawę? Porównaj uzyskaną dokładność klasyfikacji (Naiwny Klasyfikator Bayesa) z dokładnością klasyfikacji uzykaną dla lasów losowych (domyślne wartości hiperparametrów) w tej samej konfiguracji eksperymentu. Omów uzyskane wyniki.

6 Naiwny Klasyfikator Bayesa z rozkładem Bernoulliego

Zaimplementuj Naiwny Klasyfikator Bayesa zakładając rozkład prawdopodobieństwa Bernoulliego dla każdej z binarnych cech. Wykorzystaj wygładzanie Laplace'a (*Laplace smoothing*) w celu uniknięcia problemów związanych zerowaniem się prawdopodobieństwa przy wystąpieniu danych, które nie pojawiły się w danej klasie w zbiorze treningowym. Przedstaw test działania klasyfikatora korzystając z wybranego zbioru danych złożonego z krótkich tekstów (np. *20 news groups*).

$$p(x_i|\mathcal{C}_k) = \theta_{i,\mathcal{C}_k}^{x_i}(1-\theta_{i,\mathcal{C}_k}^{x_i})^{1-x_i} \tag{7}$$

7 Naiwny Klasyfikator Bayesa dla zbioru danych Adult Income

Wykorzystując *hold-out* w proporcji 70%, 30% zbadaj bazową dokładność klasyfikatora NKB dla zbioru Adult Income. Wypisz średnią dokładność klasyfikacji oraz odchylenie standardowe dla 50 podziałów (baseline). Następnie zaproponuj kilka strategii wstępnego przetwarzania zbioru danych (obsługa brakujących wartości, NA, ...), kilka strategii inżynierii cech (kodowanie cech nominalnych, skalowanie, wielokrotne transformacje Box-Cox, ...) oraz selekcji cech (Sequential Feature Selection), tak aby poprawić dokładność klasyfikacji (zbliżyć się do 84%). W implementacji możesz wykorzystać `sklearn.pipeline.Pipeline` oraz zarówno biblioteczne, jak i własne implementacje klasyfikatorów NKB.

8 Liniowa Regresja Bayesowska online ⋆

1. Zaimplementuj Liniową Regresję Bayesowską w wersji online korzystając z poniższej klasy pomocniczej

```
class BayesianLinearRegression:

    def __init__(self, n_features, alpha, beta):
        self.n_features = n_features
        self.alpha = alpha
        self.beta = beta
        self.mean = np.zeros(n_features)
        self.cov = np.identity(n_features) * alpha

    def learn(self, x, y):
        # Update the inverse covariance matrix
        # Update the mean vector
        return self

    def predict(self, x):
        # Obtain the predictive mean
        # Obtain the predictive variance
        return stats.norm(loc=y_pred_mean, scale=y_pred_var ** .5)

    @property
    def weights_dist(self):
        return stats.multivariate_normal(mean=self.mean, cov=self.cov)
```

2. Wygeneruj sztuczny zbiór danych w 2D (10 punktów z przedziału $[-1, 1]$) zgodnie ze wzorem $t = -0.2 + 0.6x + \epsilon$, gdzie ϵ jest szumem Gaussowskim o średniej 0 i odchyleniu standardowym 0.2.

3. Zwizualizuj kolejne kroki Liniowej Regresji Bayesowskiej online ($\beta = 25$, $\alpha = 2$) dla pierwszych 7 punktów ze zbioru danych. Dla każdego kroku i:

   a) Narysuj (`contourf()`) dwuwymiarowy rozkład prior dla wag w_1 i w_2 modelu $y = w_1 + w_2 x$

   b) Narysuj (`contourf()`) dwuwymiarowy rozkład posterior dla wag w_1 i w_2

   c) Narysuj rozkład predykcyjny (predictive mean, predictive interval), prostą $t = -0.2 + 0.6x$ oraz punkty wykorzystane do budowy rozkładu predykcyjnego.

4. Pokaż jak zmienia się kształt rozkładu posterior wraz z dodawaniem kolejnych punktów?