FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Lineární diskriminační analýza na proudu příznaků
Student:	Ruslana Severa
Vedoucí:	Ing. Jan Motl
Studijní program:	Informatika
Studijní obor:	Znalostní inženýrství
Katedra:	Katedra aplikované matematiky
Platnost zadání:	Do konce letního semestru 2020/21

Pokyny pro vypracování

Implementujte lineární diskriminační analýzu (LDA) pracující na proudu příznaků.

Implementace musí:
1) být "inkrementální", tzn. schopna se přiučovat na nových příznacích bez nutnosti učit se celý model od začátku,
2) být schopna se vypořádat s kolineárními příznaky,
3) podporovat diskrétní (multi-class) klasifikaci,
4) vracet výsledky porovnatelné s "dávkovou" implementací LDA, kde všechny příznaky jsou od začátku k dispozici.

Proveďte:
1) rešerši,
2) popište implementaci,
3) porovnejte inkrementální a dávkovou implementaci LDA na syntetických datech a alespoň 15 reálných datových sadách,
4) vyhodnoťte výsledky.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Karel Klouda, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 11. února 2020

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Lineární diskriminační analýza na proudu příznaků

Ruslana Severa

Katedra aplikované matematiky
Vedoucí práce: Ing. Jan Motl

29. července 2020

Poděkování

Ráda bych poděkovala panu Ing. Janovi Motlovi za trpělivé vedení bakalářské práce a za vzácné rady, které mi pomohly tuto práci dokončit.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 29. července 2020 . . . . . .

Odkaz na tuto práci

Severa, Ruslana. *Lineární diskriminační analýza na proudu příznaků.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Tato práce se věnuje lineární diskriminační analýze rozšířené o vertikální inkrementaci s konstantní regularizací. Pod pojmem vertikální inkrementace se rozumí přidání vektorů příznaků objektů místo tradičního doplnění množiny vzorků. Regularizace slouží k řešení problému multikolinearity (závislých příznaků). Je podporována diskrétní klasifikace, která probíhá na základě Bayesova rozhodovacího pravidla. Pro zrychlení výpočtu vnitřní struktury modelu je využita Choleského dekompozice, dopředná a zpětná substituce. Implementace je napsána v jazyce Python a je testována na reálných datových sadách a syntetických datech. Výsledky testování ukazují, že klasifikační model s vertikální inkrementací může nabídnout 10× rychlejší učení modelu oproti jeho dávkovému analogu při stejné klasifikační přesnosti. Finální časové zrychlení vertikální inkrementace závisí na množství příznaků a vzorků.

Klíčová slova lineární diskriminační analýza, inkrementální učení, online učení, regularizace, Choleského dekompozice, LDA, RDA

Abstract

The thesis deals with linear discriminant analysis extended by vertical increment with constant regularization. The term vertical increment means adding descriptor/feature vectors of objects instead of traditional complement of set of samples. Regularization serves to solve the problem of multicollinearity (dependent descriptors). Multiclass classification is supported and takes place on the basis of the Bayes decision rule. The Cholesky decomposition, forward and backward substitution are used to accelerate the computation of the inner structure of the model. Implementation is written in Python and is tested on real datasets and synthetic data. The results of testing show that the classification model with vertical increment could offer 10× faster model training compared to its batch analog while having the same classification precision. The final time acceleration of vertical increment depends on the amount of features and samples.

Keywords linear discriminant analysis, incremental learning, online learning, regularization, Cholesky decomposition, LDA, RDA

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Lineární diskriminační analýza[1] je klasická metoda statistiky a strojového učení, která spočívá v nalezení nejlepších lineárních kombinací příznaků pro zařazení objektů do dvou nebo více disjunktních tříd. Nalezená kombinace se používá pro redukci dimenze prostoru před následující klasifikací nebo pro samotnou klasifikaci. V dané práci se zabývám zkoumáním klasifikačního potenciálu rozšířené metody LDA. Za několik minulých desetiletí vědeckého zkoumání metoda LDA získala hodně různých forem [1], [2]. Typickým příkladem je regularizovaná diskriminační analýza[2]. Metoda RDA rozšiřuje klasickou metodu LDA o regularizaci, která řeší problém navzájem závislých příznaků a pomáhá předejít přílišnému přizpůsobení klasifikačního modelu datům (přeučení). Další formou metody LDA je inkrementální/online diskriminační analýza, která se dokáže přiučit řádkům nových vzorků. V některých případech je však získání nových vzorků příliš náročné, a proto má smysl vytvořit model, který by se dokázal přiučit sloupcům nových vlastností již získaných objektů. Z tohoto důvodu jsem se rozhodla implementovat metodu LDA, rozšířenou o výše zmíněnou regularizaci a inkrementaci nových příznaků.

I přestože je dnes metoda LDA vnímána jako relativně stará statistická metoda, má překvapivě dobré výsledky. Ve výzkumu z roku 2000 se ukázala jako téměř nejlepší z hlediska zprůměrované klasifikační chyby [3]. Autoři výzkumu dospívají k závěru, že metoda LDA poskytuje prakticky příhodné měřítko pro porovnání s budoucími algoritmy vzhledem k tomu, že je rychlá, snadno implementovatelná a dostupná ve statistických knihovnách. Ve výzkumu z roku 2010 je metoda LDA uváděna jako atraktivní alternativa k lineární metodě podpůrných vektorů (SVM) kvůli lepší výpočetní složitosti, jednoduchému konceptu a rychlejšímu učení [4].

Z výše zmíněné informace vyplývá, že metoda LDA má stále potenciál kvůli své jednoduchosti na přípravu a používání. Zlepšení a rozšíření tradičního

[1] `angl.` Linear Discriminant Analysis
[2] `angl.` Regularized Discriminant Analysis

přístupu k metodě LDA například pomocí regularizace a implementačního postupu může udělat z daného modelu velmi silný nástroj strojového učení. Dodnes je metoda LDA široce používána v různých oblastech života. Mezi příklady patří ekonomické výzkumy [5], rozpoznání lidského hlasu [6], pohybu [7] a emocí [8], medicínské a biologické výzkumy [9], [6], [10], studie o Zemi [11], [12].

Daná práce je strukturována do kapitol. První popisuje cíle práce, druhá kapitola se věnuje analýze a návrhu řešeného problému, poskytuje teoretické zázemí pro pochopení konceptu vertikální inkrementace, regularizace a obnovení vnitřní struktury klasifikačního modelu. Třetí kapitola uvádí implementační informace. Poslední kapitola se zabývá testováním implementace na reálných a syntetických datech.

Kapitola 1

Cíl práce

Primárním cílem dané práce je analýza a implementace klasifikátoru na základě metody LDA, který by dokázal pracovat s více než dvěma třídami, t.j. provádět diskrétní[3] klasifikaci nad vstupními daty. Kromě toho se musí klasifikátor umět vypořádat s kolineárními (lineárně závislými) příznaky pomocí regularizace. Implementace by také měla fungovat ve dvou módech: standardním, též dávkovém, kdy od začátku jsou známy všechny příznaky vzorců, a inkrementálním. Inkrementální mód klasifikátoru se vyznačuje schopností modelu inkrementálně se přiučit na nových příznacích, kdy model přijme na vstupu vektor nového příznaku a obnoví vnitřní strukturu bez nutnosti učit celý model od začátku.

Dalším záměrem této práce je testování funkčnosti vytvořené implementace na základě přepracovaných a připravených dat: alespoň na 15 reálných datových sadách a syntetických datech. Testování by mělo zahrnovat porovnání tradičního dávkového modelu s inkrementální implementací z hlediska klasifikační přesnosti a časových ukazatelů.

[3] multiclass

Kapitola 2

Analýza a návrh

2.1 Diskriminační analýza

Jak vyplývá z názvu, lineární diskriminační analýza je jedním z druhů diskriminační analýzy – metody mnohorozměrné statistické analýzy, která umožňuje studovat rozdíly mezi dvěma a více skupinami objektů pro několik proměnných najednou [13]. Diskriminační analýza je obecný termín, který se týká několika úzce souvisejících statistických postupů, které mohou být rozděleny podle metod interpretace meziskupinových rozdílů na diskriminaci a klasifikaci pozorování. V této práci se zabývám jenom klasifikačními schopnostmi diskriminační analýzy.

Diskriminační analýza patří do skupiny metod učení s učitelem, tj. je užitečná, když příslušnost objektů ke skupině je předem známa. Na těchto předem známých datech probíhá učení modelu. Během učení na základě vstupních hodnot dochází ke konfiguraci vnitřní struktury modelu, která pak slouží ke klasifikaci. Výraznou vlastností diskriminační analýzy jako klasifikátoru je předem známý konstantní konečný počet disjunktních skupin nebo tříd. Klasifikace pak spočívá v určení třídy, do které patří pozorovaný objekt. Objekt nebo vzorek je představen reálným vektorem $x = (x_1, \ldots, x_d), x \in \mathbb{R}^d$, jehož každá složka označuje hodnotu naměřené vlastnosti, též příznaku tohoto objektu. Těmto hodnotám se také říká diskriminační proměnné. Algoritmus přiřazení objektu k jedné z definovaných tříd se nazývá rozhodovací pravidlo. Je to zobrazení z množiny příznaků $X \subset \mathbb{R}^d$ do množiny tříd K: $a(x) : X \Rightarrow K$. Diskriminanty nebo diskriminační funkce jsou funkce, které slouží k rozlišení mezi třídami. Diskriminační funkce definují polohu diskriminačních os, které rozdělují `d`-rozměrný prostor příznaků objektů tak, aby docházelo k minimálnímu překrývání tříd. Diskriminační funkce spolu s diskriminačními osami zajišťují možnost jednoznačného přiřazení vzorku během klasifikace v závislosti na hodnotách příznaků tohoto vzorku.

Jsou známy tři základní klasifikační postupy založené na diskriminačních

funkcích [14].

- Lineární klasifikátory, kde se ke každé třídě `k` konstruuje lineární funkce. Objekt je pak přiřazen ke `k`-té třídě, pokud hodnota aplikace `k`-té lineární funkce na vektor příznaků tohoto objektu je maximální.
- Vzdálenostní metody, kde objekt je přiřazen k třídě, která má nejmenší vzdálenost mezi středem této třídy a objektem. Nejčastější používanou vzdálenostní metrikou u klasifikátorů daného druhu je Mahalanobisova vzdálenost.
- Metody maximální věrohodnosti nebo pravděpodobnostní metody, kde objekt je přiřazen ke třídě s největší posteriorní pravděpodobností.

V rámci dané bakalářské práce se soustředím na třetí uvedený klasifikační postup.

2.2 Bayesovské rozhodovací kritérium

Jak jsem již zmínila, pravděpodobnostní klasifikační metody jsou založeny na posteriorní pravděpodobnosti. Posteriorní pravděpodobnost je podmíněná pravděpodobnost náhodného jevu, jejíž výsledek závisí na nějakých aposteriorních datech (experimentálně získané informace). Jinými slovy, je to pravděpodobnost jevu A za předpokladu platnosti jevu B. Posteriorní pravděpodobnost je těsně spojena s Bayesovou větou o podmíněné pravděpodobnosti [15]

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \text{ pokud } P(B) \neq 0 \tag{2.1}$$

V kontextu klasifikace počítáme posteriorní pravděpodobnost příslušnosti nějakého objektu k určité třídě $P(k|x), x \in X, x = (x_1, \ldots, x_d), k \in K$.

$$P(k|x) = \prod_{i=1}^{d} \frac{P(x_i|k)P(k)}{P(x_i)} \tag{2.2}$$

kde $P(k) = \pi_k$ je apriorní pravděpodobnost příslušnosti náhodného vzorku k třídě k, $P(x_i)$ je apriorní pravděpodobnost, že `i`-tý příznak náhodného objektu nabývá hodnoty x_i, $P(x_i|k)$ je pravděpodobnost, že náhodný vzorek příslušný k třídě k má hodnotu `i`-toho příznaku rovnou x_i. Hodnoty příznaků mohou nabývat jakýchkoliv hodnot, to znamená, že jde o spojité veličiny. Pokud příznaky v rámci jedné třídy dodržují stejné pravděpodobnostní rozdělení, posteriorní pravděpodobnost lze vyjádřit ve tvaru

$$P(k|x) = \frac{\rho(x|k)\pi_k}{\rho(x)} \tag{2.3}$$

kde $\rho(x|k) = \rho_k(x)$ je hustota pravděpodobnosti příznaků v rámci třídy k a $\rho(x)$ je apriorní hustota pravděpodobnosti veličiny x. Jelikož apriorní hustota pravděpodobnosti $\rho(x)$ je stejná pro všechny třídy, tj. nemá vliv na rozlišení mezi třídami, můžeme tento člen vynechat. Bayesovským rozhodovacím kritériem nebo pravidlem se pak nazývá klasifikační metoda, podle které je objekt zařazen do třídy s největší posteriorní pravděpodobností

$$a(x) = \arg\max_{k \in K} P(k|x) = \arg\max_{k \in K} (\pi_k \rho_k(x)) \tag{2.4}$$

Bayesovské rozhodovací kritérium dosahuje nejmenší střední ztráty vzniklou špatnou klasifikací $R(a)$ mezi všemi známými rozhodovacími pravidly [16].

2.3 Předpoklady a požadavky

2.3.1 Diskriminační analýza

Diskriminační analýza klade přísné požadavky na vstupní data [17].

- Data musí být rozdělena alespoň do dvou tříd (binární klasifikace).
- Každá třída musí být představena alespoň dvěma vzorky.
- Celkový počet příznaků (diskriminačních proměnných) nesmí překročit celkový počet objektů.
- Příznaky musí být kvantitativní a mezi příznaky musí být slabá korelace.
- Kovarianční matice všech tříd by měly být co nejvíce homogenní.
- Předpokládá se, že analyzované diskriminační proměnné představují výběrový soubor z vícerozměrného normálního rozdělení.

2.3.2 Kvadratická diskriminační analýza

Kvadratická diskriminační analýza rozšiřuje množinu základních předpokladů a požadavků diskriminační analýzy o předpoklad o normálním (Gaussově) rozdělení příznaků v jednotlivých třídách.

Hustota pravděpodobnosti normálního rozdělení $\mathcal{N}(\mu, \sigma^2)$, kde μ je střední hodnota a σ^2 je rozptyl, se rovná

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.5}$$

Analogicky, hustota pravděpodobnosti `d`-dimenzionálního normálního rozdělení $\mathcal{N}(\mu, \Sigma)$, kde μ je vektor středních hodnot a Σ je kovarianční matice, se rovná

$$p(x_1, \ldots, x_p) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{2.6}$$

kde $\det \Sigma$ značí determinant kovarianční matice. Aplikace logaritmické funkce nebude mít žádný negativní vliv na výsledek klasifikačního algoritmu (2.4), jelikož logaritmická funkce je monotonně rostoucí.

$$a(x) = \arg\max_{k \in K}(\pi_k \rho_k(x)) = \arg\max_{k \in K} \ln(\pi_k \rho_k(x)) = \arg\max_{k \in K} S_k(x) \tag{2.7}$$

kde $S_k(x)$ značí diskriminační skóre. Pomocí substitucí hustoty pravděpodobnosti `d`-dimenzionálního normálního rozdělení (2.6), logaritmování a odečtení konstanty lze nalézt explicitní vyjádření diskriminačního skóre pro QDA

$$S_k(x) = \ln(\pi_k \rho_k(x)) \tag{2.8a}$$

$$= \ln \pi_k + \ln\left(\frac{1}{\sqrt{(2\pi)^d \det \Sigma_k}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}\right) \tag{2.8b}$$

$$= \ln \pi_k - \frac{1}{2}\left(\ln(2\pi)^d + \ln(\det \Sigma_k)\right) - \frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) \tag{2.8c}$$

$$= \ln \pi_k - \frac{1}{2}(\det \Sigma_k) - \frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) \tag{2.8d}$$

2.3.3 Lineární diskriminační analýza

Lineární diskriminační analýza je speciálním případem kvadratické diskriminační analýzy, kde se předpokládá normální rozdělení příznaků v jednotlivých třídách $\mathcal{N}(\mu_k, \Sigma_k)$ se stejnou kovarianční maticí pro všechny třídy, tj. $\Sigma_k = \Sigma$ pro $\forall k \in K$ [18]. Právě tento předpoklad přináší linearitu do klasifikátoru. Na obrázku 2.1 z oficiálních webových stránek scikit-learn [19] lze vidět rozdíl mezi formou rozhodovacích hranic: kvadratické u QDA a lineární u LDA.

Roznásobením a vynecháním všech nezávislých na třídách členů lze odvodit explicitní vyjádření diskriminačního skóre pro LDA z vzorce diskriminačního skóre pro QDA (2.8d)

$$S_k(x) = \ln \pi_k - \frac{1}{2}(\det \Sigma_k) - \frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) \tag{2.9a}$$

$$= \ln \pi_k - \frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k) \tag{2.9b}$$

$$= \ln \pi_k - \frac{1}{2}x^T\Sigma^{-1}x + \frac{1}{2}x^T\Sigma^{-1}\mu_k + \frac{1}{2}\mu_k^T\Sigma^{-1}x - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k \tag{2.9c}$$

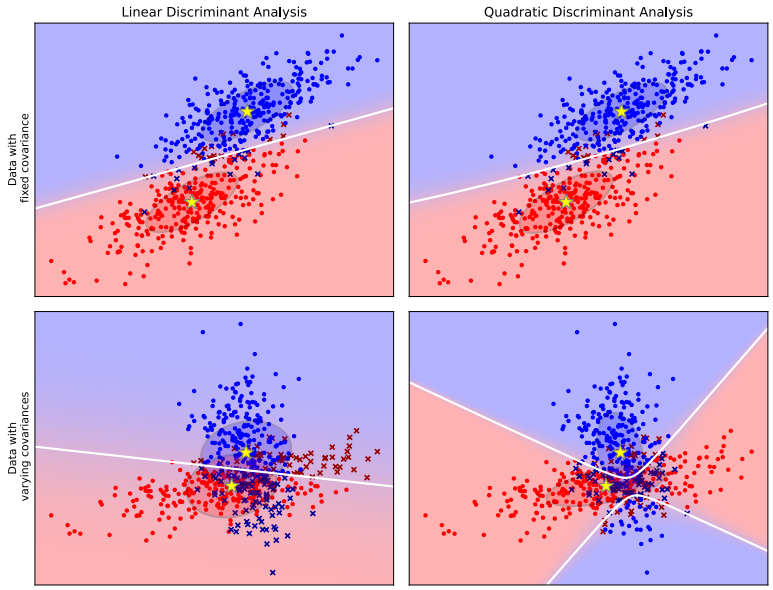
$$= \ln \pi_k + \frac{1}{2}x^T\Sigma^{-1}\mu_k + \frac{1}{2}\mu_k^T\Sigma^{-1}x - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k \tag{2.9d}$$

$$= \ln \pi_k + \frac{1}{2}x^T\Sigma^{-1}\mu_k + \frac{1}{2}x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k \tag{2.9e}$$

$$= \ln \pi_k + x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k \tag{2.9f}$$

Funkce $S_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \ln \pi_k$ se také nazývá lineární diskriminant pro třídu k a vzorek x [20].

Obrázek 2.1: Porovnání LDA a QDA



2.4 Odhad parametrů

Z vzorce diskriminačního skóre pro LDA (2.9f) lze vidět, že základní vnitřní struktura klasifikačního modelu by měla obsahovat následující prvky:

- třídní apriorní pravděpodobnosti $\pi_k, k \in K$;
- vektory středních hodnot anebo třídních průměrů $\mu_k, k \in K$;
- společná kovarianční matice Σ.

V praxi ale skutečné hodnoty těchto prvků nejsou známé, a proto se je snažíme odhadnout na základě dostupných trénovacích dat. Nechť $X \subset \mathbb{R}^d$ je konečná množina trénovacích dat, d je počet příznaků, $|X|$ je mohutnost trénovací množiny, tj. celkový počet vzorků, K je konečná množina tříd, $|K|$ je celkový počet tříd, $X_k \subset X$ je podmnožina prvků příslušných k třídě $k \in K$, $|X_k|$ je počet vzorků příslušných k třídě k, x_{ki} je `i`-tý vzorec v množině X_k, pak vychýlené[4] maximálně věrohodné odhady složek vnitřní struktury lze vyjádřit

[4] `angl.` biased

ve tvaru [20]

$$\hat{\pi}_k = \frac{|X_k|}{|X|} \tag{2.10a}$$

$$\hat{\mu}_k = \frac{1}{|X_k|} \sum_{i=1}^{|X_k|} x_{ik} \tag{2.10b}$$

$$\hat{\Sigma} = \frac{1}{\sum_{i=1}^{|K|} |X_i|} \sum_{i=1}^{|K|} \sum_{j=1}^{|X_i|} (x_{ij} - \hat{\mu}_i)(x_{ij} - \hat{\mu}_i)^T \tag{2.10c}$$

$$= \frac{1}{|X|} \sum_{i=1}^{|K|} \sum_{j=1}^{|X_i|} (x_{ij} - \hat{\mu}_i)(x_{ij} - \hat{\mu}_i)^T \tag{2.10d}$$

$$= \frac{1}{|X|} \sum_{i=1}^{|K|} |X_i| \frac{1}{|X_i|} \sum_{j=1}^{|X_i|} (x_{ij} - \hat{\mu}_i)(x_{ij} - \hat{\mu}_i)^T \tag{2.10e}$$

$$= \frac{1}{|X|} \sum_{i=1}^{|K|} |X_i| \hat{\Sigma}_i \tag{2.10f}$$

$$= \sum_{i=1}^{|K|} \hat{\pi}_i \hat{\Sigma}_i \tag{2.10g}$$

kde $\hat{\pi}_k$ je odhad apriorní pravděpodobnosti třídy k, $\hat{\mu}_k$ je vektor výběrových středních hodnot třídy k anebo výběrový průměr třídy k, $\hat{\Sigma}$ je vychýlený sdružený[5] odhad společné kovarianční matice, $\hat{\Sigma}_k$ je vychýlený odhad kovarianční matice třídy k. Společná kovarianční matice je v daném případě váženým průměrem třídních kovariančních matic. To znamená, že velké skupiny (třídy s větším počtem vzorků) ovlivní výslednou společnou kovarianční matici ve větší míře než malé skupiny.

2.5 Scikit-learn

V roli důležitého teoretického a praktického podkladu pro mě vystoupila svobodná softwarová knihovna scikit-learn se zaměřením na strojové učení pro programovací jazyk Python. Scikit-learn nabízí k používání implementaci klasifikátoru na základě lineární diskriminační analýzy, otevřený přístup ke zdrojovému kódu dané implementace a velmi kvalitní informativní dokumentaci. LDA v scikit-learn je představena třídou LinearDiscriminantAnalysis [21], která slouží jak pro klasifikační účely, tak i pro účely redukce dimenzionality. Níže je představen popis základních vstupních parametrů pro pochopení konceptu fungování dané implementace.

[5] `angl.` pooled

- **solver**: nabízí tři možnosti řešení klasifikačního problému:
  - „*svd*“ nebo singulární rozklad;
  - „*lsqr*“ nebo metoda nejmenších čtverců;
  - „*eigen*“ nebo dekompozice vlastních čísel.
- **shrinkage**: podporuje regularizaci, která je popsána v sekci 2.6, na základě hodnoty vstupního parametru:
  - „*None*“ znamená žádnou regularizaci;
  - „*auto*“ vede k použití vzorce Ledoit a Wolf [22] pro odhad shrinkage;
  - číslo typu float v intervalu (0,1) zaručí regularizaci s obdrženým koeficientem.
- **priors**: povoluje zadat třídní apriorní pravděpodobnosti, které se defaultně počítají na základě trénovacích dat;
- **n_components**: povoluje zadat počet komponent pro redukci dimenzionality;
- **store_covariance**: určuje, zda je potřeba uložit kovarianční matici;
- **tol**: parametr singulárního rozkladu.

Mezi dvě základní metody třídy LinearDiscriminantAnalysis nutné ke klasifikaci patří:

- **fit**, která slouží k učení modelu na dvou vstupních parametrech:
  - matice diskriminačních proměnných X, tj. matice hodnot příznaků objektů;
  - vektor vysvětlované proměnné y, tj. vektor příslušnosti objektů matice X k třídám.
- **predict**, která na základě hodnot příznaků vstupní matice objektů X vrátí predikci nebo klasifikaci těchto objektů do množiny tříd předem definované během učení.

Řešení „*lsqr*“ se vyznamenává použitím metody nejmenších čtverců [23] pro výpočet třídních koeficientů $w_k = \Sigma^{-1}\mu_k$, čímž se vyhýbá explicitnímu sestrojení inverzní matice Σ^{-1}. Toto zrychlení je velmi důležitým faktorem pro inkrementální přístup, a proto jsem použila právě toto řešení ze tří dostupných jako hlavní podklad během implementace mé vlastní verze LDA a také pro následující testovací účely v kapitole 4. Řešení LDA na základě dekompozice vlastních čísel („*eigen*“) spočívá v optimalizaci poměru matice mezitřídní variability k matici vnitrotřídní variability (Fisherovo rozhodovací kritérium

[24]). Slouží jak pro klasifikaci, tak i pro redukci dimenzionality, což však není tématem této práce. Navíc se toto řešení obrací k pomalejšímu explicitnímu postupu výpočtu inverze společné kovarianční matice. Řešení založené na singulárním rozkladu („*svd*“) na rozdíl od ostatních dvou vůbec nepoužívá kovarianční matici a z tohoto důvodu ani nepodporuje regularizaci, kvůli čemu také nevyhovuje cílům dané práce.

2.6 Regularizace

Jedním z důležitých požadavků diskriminační analýzy na vstupní data je slabá míra korelace. Skutečná data však nejsou většinou nezávislá a mezi příznaky objektů může existovat nějaká lineární závislost. V takovém případě mluvíme o multikolinearitě matice vstupních dat. Společná kovarianční matice $\hat{\Sigma}$ vypočítaná na základě multikolinearní trénovací množiny dat má determinant rovný nule. To znamená, že tato společná kovarianční matice není regulární, ale singulární, a nelze pro ni sestrojit inverzní matici $\hat{\Sigma}^{-1}$, potřebnou pro nalezení diskriminačního skóre $S_k(x)$ (2.9f). Stejný problém vzniká v případě, kdy celkový počet vlastností převyšuje celkový počet objektů $d > |X|$ [25].

Počet příznaků a jejich korelace však nejsou jedinými problematickými aspekty ovlivňujícími nalezení inverzní matice ke společné kovarianční matici. Důležitou charakteristikou matice je její podmíněnost anebo číslo podmíněnosti. Matice se nazývá špatně podmíněnou[6], pokud její číslo podmíněnosti je příliš velké. V extrémním případě, kdy číslo podmíněnosti se rovná nekonečnu, matice se stává singulární. Inverzní matice k špatně podmíněné matici je výpočetně nestabilní. Tato nestabilita se projevuje jako prakticky nepředvídatelná transformace $\hat{\Sigma}^{-1}$ při nepatrných variacích trénovacích dat. Z toho vyplývá, že podmíněnost matice $\hat{\Sigma}$ do značné míry ovlivňuje klasifikační přesnost celého modelu.

Existuje řada způsobů řešení problematiky singulární a špatně podmíněné matice. Jednou z velmi populárních strategií je již zmíněná v minulé sekci regularizace [26]. Regularizace je metoda úpravy původní nekorektní funkce na její zobecněnou podobu. V případě LDA regularizace je transformace špatně podmíněné a singulární společné kovarianční matice a následující nahrazení této matice jejím regulárním a dobře podmíněným maticovým analogem.

Hustota vícerozměrného normálního rozdělení geometricky tvoří elipsoidu nebo elipsu v případě dvourozměrného rozdělení, jak lze vidět na obrázku 2.2, který zobrazuje eliptickou představu dvourozměrné kovarianční matice [27]. Vlastní vektory $\vec{v_i}$ matice $\hat{\Sigma}$ udávají směr os jejího elipsoidu. Vlastní čísla λ_i určují „tloušťku“ elipsoidu podél jeho os. Při regularizaci dochází k mírnému zkreslení tvaru rozdělení, kvůli čemu se matice stává regulární a dobře podmíněnou. Nejjednodušším způsobem, jak zvětšit všechna vlastní čísla matice $\hat{\Sigma}$ o regularizační parametr α tak, aby vlastní vektory zůstaly

[6] `angl.` ill-conditioned

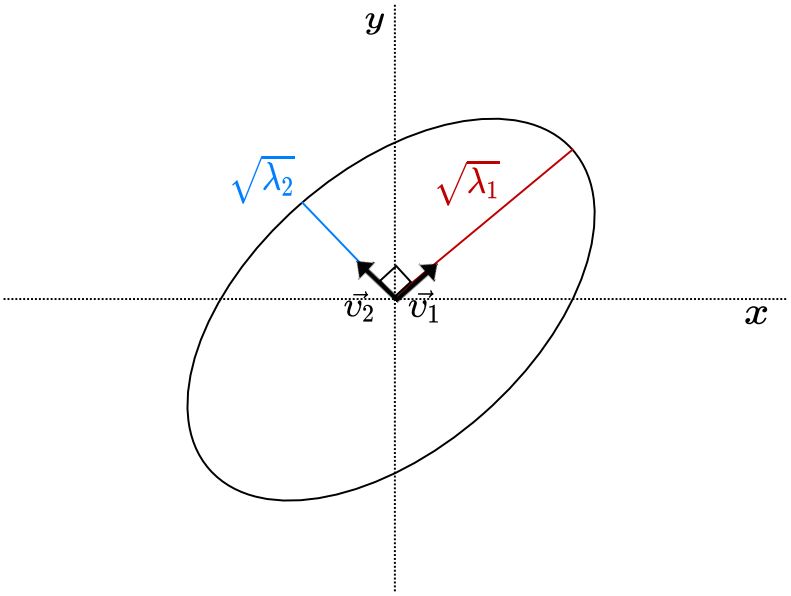
beze změny, je použití regularizačního vzorce [16]

$$\hat{\Sigma} = \hat{\Sigma} + \alpha\mathbb{I} \tag{2.11}$$

kde $\mathbb{I}$ je $d \times d$ jednotková matice, d značí počet příznaků a $0 < \alpha < 1$. Jestli v je vlastní vektor matice $\hat{\Sigma}$ a λ je tomuto vektoru příslušné vlastní číslo, tj. $\hat{\Sigma}v = \lambda v$, pak v je zároveň vlastním vektorem matice $\hat{\Sigma} + \alpha\mathbb{I}$ s jemu přidruženým vlastním číslem rovným $\lambda + \alpha$.

$$(\hat{\Sigma} + \alpha\mathbb{I})v = \hat{\Sigma}v + \alpha\mathbb{I}v = \lambda v + \alpha v = (\lambda + \alpha)v \tag{2.12}$$

Obrázek 2.2: Eliptická představa dvourozměrné kovarianční matice



Regularizace může také měnit, přesněji řečeno uměřeně zmenšovat prvky kovarianční matice mimo diagonálu. Takovou regularizaci lze vyjádřit ve tvaru

$$\hat{\Sigma} = (1 - \alpha)\hat{\Sigma} + \alpha\mathbb{I} \tag{2.13}$$

Regularizační vzorec, který se používá třídou LinearDiscriminantAnalysis, je jedním z hlavních konceptů regularizované diskriminační analýzy [16]:

$$\hat{\Sigma}_k = (1 - \alpha)\hat{\Sigma}_k + \frac{\alpha}{d}\,\mathrm{tr}(\hat{\Sigma}_k)\mathbb{I} \tag{2.14}$$

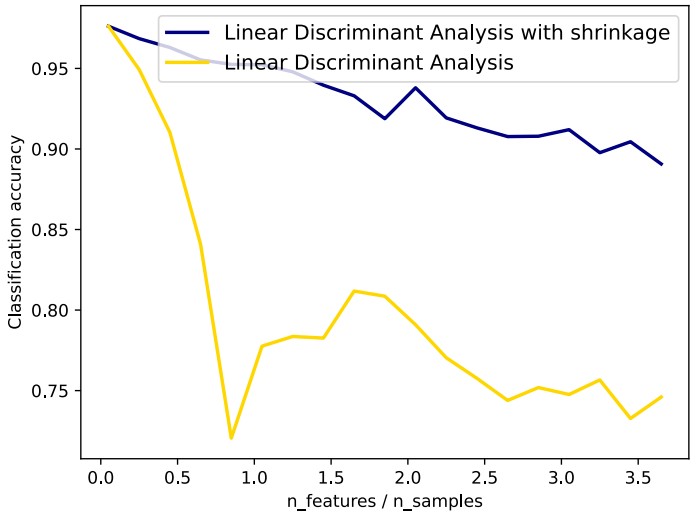
kde $\mathrm{tr}(\hat{\Sigma}_k)$ je stopa matice $\hat{\Sigma}_k$, $\hat{\Sigma}_k$ je kovarianční matice třídy k a koeficient α je nalezen pomocí algoritmu pro výpočet optimálního regularizačního parametru Ledoit a Wolf [28]. Regularizačnímu parametru α se v takovém případě říká shrinkage a $\hat{\Sigma}_k$ – shrinkage odhad, od anglického slovesa „shrink“, které jde

přeložit jako „srazit se, zmenšit se“. Odhad společné kovarianční matice je pak vypočítán pomocí vzorce (2.10g)

$$\hat{\Sigma} = \sum_{i=1}^{|K|} \hat{\pi}_i \left[(1-\alpha)\hat{\Sigma}_i + \frac{\alpha}{d} \operatorname{tr}(\hat{\Sigma}_i)\mathbb{I} \right] \tag{2.15}$$

Regularizace v oblasti strojového učení je často brána jako preventivní metoda proti přeučení modelu, které vede k výraznému zhoršení kvality predikce. Přeučení znamená přílišné přizpůsobení modelu trénovacím datům včetně přizpůsobení informačnímu hluku, který neodráží skutečnou strukturu dat, ale plní roli náhody. Pozitivní vliv regularizace na klasifikační přesnost LDA je zobrazen na obrázku 2.3 z oficiálních webových stránek knihovny scikit-learn [29].

Obrázek 2.3: Vliv regularizací na klasifikační přesnost LDA



Jiným způsobem řešení problému multikolinearity a špatné podmíněnosti matice je použití nějakého algoritmu redukce dimenzionality, například analýzy hlavních komponent. Tento způsob však není optimální, což ukázali ve své práci Ledoit a Wolf při porovnání PCA a regularizace kovarianční matice v analýze akciového trhu [28].

2.7 Inkrementální přístup

Tradiční přístup LDA vyžaduje dostupnost kompletní trénovací sady vzorků před začátkem učení. V reálném světě se však může stát, že celá sada trénovacích dat není k dispozici a bude se doplňovat o nové trénovací vzorky během učení. V takovém případě má smysl upravit algoritmus LDA tak, aby

bylo možné aktualizovat vnitřní strukturu modelu s již provedenými výpočty na základě hodnot nových vzorků bez potřeby opětného spuštění celého mechanismu. Takový přístup se nazývá inkrementální [30], [31] nebo online [30] a obvykle znamená postupné přidávání `d`-rozměrných vektorů nových objektů, buď jeden po druhém, nebo seskupeně. Takovou inkrementaci by se dalo pojmenovat „horizontální“ z hlediska směru rozšíření matice trénovacích dat, kde vzorky tvoří řádky této matice. Horizontální inkrementaci jde schematicky popsat pomocí obrázku 2.4, kde n je původní počet vzorků v matici trénovacích dat, y_i značí příslušnost `i`-toho vzorku k nějaké třídě, x_{ij} je hodnota `j`-tého příznaku `i`-tého vzorku.

Obrázek 2.4: Horizontální inkrementace

$$\left.\begin{bmatrix} y_1 & x_{11} & \dots & x_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ y_n & x_{n1} & \dots & x_{nd} \end{bmatrix}\right\} \text{původní matice}$$

$$\left.\begin{bmatrix} y_{n+1} & x_{(n+1)1} & \dots & x_{(n+1)d} \end{bmatrix}\right\} \text{přidané vzorky}$$
$$\left.\begin{bmatrix} y_{n+2} & x_{(n+2)1} & \dots & x_{(n+2)d} \end{bmatrix}\right\}$$

Nicméně získání nových vzorců může být velmi obtížné například z hlediska časové náročnosti, nebo může vyžadovat velké finanční prostředky. V takovém případě vhodnější volbou je studování již získaných vzorků a měření hodnot zatím neprozkoumaných příznaků těchto vzorků. Jako dobrý ilustrační příklad může posloužit abstraktní medicínský model, zabývající se klasifikací nemocí na základě dat o pacientech. Čekání na nového pacienta je v dané situaci podřadnější možností pro získání nových dat pro zlepšení kvality klasifikátoru než navazující zkoumání nemocných, které se již nacházejí pod lékařským dohledem.

Obrázek 2.5: Vertikální inkrementace

$$\underbrace{\begin{bmatrix} y_1 & x_{11} & \dots & x_{1d} \\ y_2 & x_{21} & \dots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ y_n & x_{n1} & \dots & x_{nd} \end{bmatrix}}_{\text{původní matice}} \underbrace{\begin{bmatrix} x_{1(d+1)} \\ x_{2(d+1)} \\ \vdots \\ x_{n(d+1)} \end{bmatrix} \begin{bmatrix} x_{1(d+2)} \\ x_{2(d+2)} \\ \vdots \\ x_{n(d+2)} \end{bmatrix}}_{\text{přidané příznaky}}$$

V dané bakalářské práci rozšiřuji standardní klasifikační model na základě LDA o inkrementaci na proudu příznaků, též o „vertikální“ inkrementaci, kdy dochází k přidávání n-rozměrných vektorů naměřených hodnot nových vlastností do již existujícího klasifikačního modelu. Vertikální inkrementaci jde schematicky popsat pomocí obrázku 2.5, kde d je původní počet příznaků v matici trénovacích dat, y_i značí příslušnost `i`-tého vzorku k nějaké třídě, x_{ij} je hodnota `j`-tého příznaku `i`-tého vzorku.

2.7.1 Konstantní regularizace

Regularizační metoda používaná v knihovně scikit-learn (2.15) není vhodná v případě LDA na proudu příznaků, jelikož přidání nové vlastnosti vede ke dvěma důležitým změnám: regularizačního parametru α na základě algoritmu Lediot a Wolf a stop třídních matic $\mathrm{tr}(\hat{\Sigma}_k)$. Tyto transformace znemožňují efektivní obnovení vnitřní struktury klasifikátoru, které je základním konceptem inkrementálního přístupu. Z toho vyplývá, že je potřebné použití konstantního regularizačního parametru α a zjednodušeného regularizačního vzorce (2.13), což povoluje vyjádřit odhad společné kovarianční matice ve tvaru

$$\hat{\Sigma} = \sum_{i=1}^{|K|} \hat{\pi}_i \left[(1-\alpha)\hat{\Sigma}_i + \alpha \mathbb{I} \right] \tag{2.16}$$

2.7.2 Obnovení základní vnitřní struktury

Při vstupu vektoru `(d+1)`-tého příznaku v_{d+1} do klasifikačního modelu se nejprve obnoví základní prvky vnitřní struktury.

- Pro každou třídu $k \in K$ se zvětší vektor výběrových průměrů $\hat{\mu}_k$ o jednu položku $\hat{\mu}_{k(d+1)}$, vypočítanou na základě $v_{k(d+1)}$, kde $v_{k(d+1)}$ značí část vektoru v_{d+1} příslušnou k třídě k.
- Odhad regularizované společné kovarianční matice se zvětší o jeden identický sloupec a řádek c_{d+1}, jelikož kovarianční matice je symetrická. Tento sloupec/řádek c_{d+1} lze vyjádřit ve tvaru

$$c_{d+1} = \sum_{i=1}^{|K|} \hat{\pi}_i \left[\frac{1-\alpha}{|X_i|} (v_{i(d+1)} - \hat{\mu}_{i(d+1)})(\hat{\Sigma}_i - \hat{\mu}_i) + \alpha\iota \right] \tag{2.17}$$

  kde ι značí poslední řádek $(d+1)\times(d+1)$ jednotkové matice, tj. vektor $[\underbrace{0 \quad \dots \quad 0}_{d \text{ krát}} \quad 1]$.

2.7.3 Choleského dekompozice

Časově nejnáročnější částí při výpočtu diskriminačního skóre (2.9f) je sestrojení inverzní matice $\hat{\Sigma}^{-1}$. Pro zrychlení a zjednodušení výpočtu inverzní matice

se hodí použití Choleského dekompozice [32]. Jestli reálná čtvercová matice A je pozitivně definitní, pak může být vyjádřena ve tvaru

$$A = LL^T \tag{2.18}$$

kde L je dolní trojúhelníková matice. Tomuto rozkladu se říká Choleského dekompozice nebo Choleského rozklad. Kovarianční matice je podle definice reálná, čtvercová, symetrická a pozitivně semidefinitní, tj. $\hat{\Sigma}$ může byt zapsána ve tvaru

$$\hat{\Sigma} = LL^T \tag{2.19}$$

Inverzní matice k odhadu společné kovarianční matice pak může být vypočítána pomocí vzorce

$$\hat{\Sigma}^{-1} = (LL^T)^{-1} = (L^T)^{-1}L^{-1} = (L^{-1})^T L^{-1} \tag{2.20}$$

Velkou výhodou a prvním ze dvou hlavních důvodů použití Choleského dekompozice pro inkrementální LDA je inkrementální struktura algoritmu tohoto rozkladu.

$$\begin{aligned} L_{i,j} &= \sqrt{\hat{\Sigma}_{i,j} - \sum_{k=1}^{j-1}(L_{i,k}^2)}, \text{ pokud } i = j \\ L_{i,j} &= \frac{1}{L_{j,j}}\left(\hat{\Sigma}_{i,j} - \sum_{k=1}^{j-1} L_{i,k}L_{j,k}\right), \text{ pokud } i \neq j \end{aligned} \tag{2.21}$$

kde $i = j, \ldots, d, j = 1, \ldots, d$, d je aktuální počet příznaků.

Druhým důvodem je způsobilost Choleského dekompozice při řešení soustavy lineárních rovnic $Ax = b$, kde A je symetrická a pozitivně definitní matice. V případě inkrementální LDA se tato schopnost Choleského dekompozice hodí pro nalezení $\hat{\Sigma}^{-1}\hat{\mu}_k$, jinými slovy pro nalezení řešení rovnice

$$\hat{\Sigma}\chi_k = \hat{\mu}_k \text{ pro } \forall k \in K \tag{2.22}$$

Rovnici (2.22) lze dle Choleského rozkladu (2.19) přepsat ve tvaru

$$LL^T\chi_k = \hat{\mu}_k \text{ pro } \forall k \in K \tag{2.23}$$

Nechť $L^T\chi_k = \gamma_k$ a $L\gamma_k = \hat{\mu}_k$, pak řešení rovnice $L\gamma_k = \hat{\mu}_k$ lze získat pomocí dopředné substituce[7]

$$\gamma_{ki} = \frac{1}{L_{i,i}}\left(\hat{\mu}_{ki} - \sum_{j=1}^{i-1}(L_{i,j}\gamma_{kj})\right) \text{ pro } i = 1, \ldots, d \tag{2.24}$$

[7] `angl.` forward substitution

Následovně pomocí získaného řešení γ_k a zpětné substituce[8] lze nalézt řešení rovnice $L^T\chi_k = \gamma_k$, které je zároveň řešením rovnice $\hat{\Sigma}\chi_k = \hat{\mu}_k$

$$\chi_{ki} = \frac{1}{L^T_{i,i}}\left(\gamma_{ki} - \sum_{j=1}^{i-1}(L^T_{i,j}\chi_{kj})\right) \text{ pro } i = 1, \ldots, d \tag{2.25}$$

kde γ_{ki}/χ_{ki} je `i`-tá položka vektoru γ_k/χ_k. Ve výsledku při vstupu vektoru `(d+1)`-tého příznaku v_{d+1} do klasifikačního modelu kromě základních prvků, popsaných v sekci 2.7.2, se obnoví následující elementy vnitřní struktury.

- Dolní trojúhelníková matice L se zvětší o jeden dolní řádek l_{d+1}, spočítaný na základě obnoveného odhadu společné kovarianční matice $\hat{\Sigma}$ pomocí inkrementálního algoritmu Choleského rozkladu (2.21).
- Pro každou třídu $k \in K$ se zvětší vektor $\gamma_k = (L^{-1})\hat{\mu}_k$ o jednu skalární položku, spočítanou na základě obnovené trojúhelníkové matice L a obnoveného vektoru $\hat{\mu}_k$ pomocí dopředné substituce (2.24).
- Pro každou třídu $k \in K$ se vypočte nový vektor $\chi_k = \hat{\Sigma}^{-1}\hat{\mu}_k$ na základě obnoveného vektoru γ_k a obnovené trojúhelníkové matice L pomocí zpětné substituce (2.25).
- Pro každou třídu $k \in K$ se vypočte nový skalár ω_k, potřebný pro výpočet diskriminačního skóre (2.9f).

$$\omega_k = \ln\hat{\pi}_k - \frac{1}{2}\hat{\mu}_k^T\hat{\Sigma}^{-1}\hat{\mu}_k = \ln\hat{\pi}_k - \frac{1}{2}\hat{\mu}_k^T\chi_k \tag{2.26}$$

V tabulce 2.1 lze pozorovat změnu rozměrů všech obnovujících prvků vnitřní strukturu.

Tabulka 2.1: Obnovení vnitřní struktury inkrementálního modelu

Prvek vnitřní struktury	Rozměr před obnovením	Rozměr po obnovení
$\hat{\mu}_k$	d	$d+1$
$\hat{\Sigma}$	$d \times d$	$(d+1) \times (d+1)$
L	$d \times d$	$(d+1) \times (d+1)$
$\gamma_k = L^{-1}\hat{\mu}_k$	d	$d+1$
$\chi_k = \hat{\Sigma}^{-1}\hat{\mu}_k$	d	$d+1$
$\omega_k = -\frac{1}{2}\hat{\mu}_k^T\hat{\Sigma}^{-1}\hat{\mu}_k + \ln\hat{\pi}_k$	1	1

Diskriminační skóre lze pak vyjádřit ve tvaru

$$S_k(x) = x^T\chi_k + \omega_k = x\chi_k^T + \omega_k \tag{2.27}$$

[8] `angl.` backward substitution

Kapitola 3

Implementace

3.1 Třída LDAClassifier

Jak jsem již zmínila, hlavním praktickým podkladem pro mě posloužila implementace LDA v knihovně scikit-learn. Moje implementace je také napsána v jazyce Python a je představena třídou LDAClassifier. Na stránce webové služby GitHub je umístěn repositář „LDA“, kde lze nalézt úplný zdrojový kód třídy LDAClassifier a testovací soubory. Tento repositář je dostupný na **github.com/ruslanasevera/LDA**. Mezi hlavní atributy třídy, které reprezentují vnitřní strukturu klasifikačního modelu, patří:

- **type** = druh klasifikátoru: „*offline*“ pro dávkovou verzi LDA, kdy všechny informace o objektech jsou dostupné před učením modelu, „*online*“ pro vertikální inkrementaci. Tato volba je zavedena především pro testovací účely a možnost porovnání výsledků.
- **regularization_coefficient** = regularizační parametr: „*None*“ pro klasifikaci bez použití regularizace, reálné číslo v intervalu (0,1) pro provedení konstantní regularizace. Defaultně je nastaven na 0.01.
- **max_num_of_features** = maximální počet příznaků v případě vertikální inkrementace. Slouží pro dopřednou alokaci pamětí, díky čemu se lze vyhnout časově nákladným funkcím změny velikosti pole jako třeba `numpy.insert`/`numpy.append`/`numpy.hstack`. Podobný princip fungování se používá například v Java ArrayList. Defaultně je nastaven na 50.
- **class_labels** = vektor původních názvů tříd $[k \in K]$.
- **num_classes** = počet tříd $[|K|]$.
- **X** = matice hodnot příznaků vzorků, na kterých probíhá učení modelu $[X_{\text{tren}}]$.

- **y** = vektor příslušnosti vzorků X_{tren} ke třídám $[y_{\text{tren}}]$.
- **overall_num_obs** = celkový počet vzorků $[|X_{\text{tren}}|]$.
- **num_obs_by_class** = vektor, `i`-tý prvek odpovídá počtu vzorků v `i`-té třídě. Pořadí tříd ve všech podobných atributech souhlasí s pořadím **class_labels** $[|X_k|, X_k \subset X_{\text{tren}}]$.
- **prob_classes** = vektor, `i`-tý prvek odpovídá odhadu apriorní pravděpodobnosti `i`-té třídy $[\hat{\pi}_k]$.
- **log_prob_classes** = vektor, `i`-tý prvek odpovídá přirozenému logaritmu z odhadu apriorní pravděpodobnosti `i`-té třídy $[\ln \hat{\pi}_k]$.
- **mean_vecs_by_class** = matice, `i`-tý řádek odpovídá vektoru výběrových průměrů `i`-té třídy $[\hat{\mu}_k]$.
- **total_cov_mat** = odhad společné kovarianční matice $[\hat{\Sigma}]$. Pro zjednodušení a zrychlení maticových výpočtů **total_cov_mat** ukládá jenom dolní trojúhelníkovou společnou kovarianční matici, jelikož je symetrická.
- **num_features** = aktuální počet příznaků $[d]$.
- **L** = dolní trojúhelníková matice dle Choleského rozkladu společné kovarianční matice $[L]$.
- **L_inv_mv_by_class** = matice, `i`-tý řádek odpovídá součinu inverze dolní trojúhelníkové matice a vektoru výběrových průměrů `i`-té třídy, výpočet pomocí dopředné substituce $[\gamma_k = L^{-1}\hat{\mu}_k]$.
- **total_cov_mat_inv_mv_by_class** = matice, `i`-tý řádek odpovídá součinu inverze odhadu společné kovarianční matice a vektoru výběrových průměrů `i`-té třídy, výpočet pomocí zpětné substituce $[\chi_k = \hat{\Sigma}^{-1}\hat{\mu}_k]$.
- **mv_total_cov_mat_inv_m_by_class** = vektor, `i`-tý prvek odpovídá součtu přirozeného logaritmu z odhadu apriorní pravděpodobnosti `i`-té třídy a součinu (-0.5) transponovaného vektoru výběrových průměrů `i`-té třídy, inverze odhadu společné kovarianční matice a stejného vektoru výběrových průměrů `i`-té třídy $[\omega_k = -\frac{1}{2}\hat{\mu}_k^T\hat{\Sigma}^{-1}\hat{\mu}_k + \log\hat{\pi}_k]$.
- **scores** = matice, `i`-tý sloupec odpovídá diskriminačnímu skóre `i`-té třídy na vstupních datech X_{test} $[S_k]$.

Mezi nejpodstatnější metody třídy LDAClassifyer patří:

- **fit**, která slouží k naučení modelu na vstupních datech X_{tren} a y_{tren}.

- **add_feature**, která povoluje přidat vektor příznaku do klasifikačního modelu a tím vertikálně zvětšit X_{tren}. Po přidání příznaku proběhne obnovení vnitřní struktury modelu dle algoritmů popsaných v sekcích 2.7.2 a 2.7.3.

- **predict**, která provádí samotnou klasifikaci, tj. vytváří predikci příslušnosti vzorků $x \in X_{\text{test}}$ k třídám na základě vnitřní struktury modelu pomocí diskriminačního skóre (2.27).

3.2 Použité funkce

Jedním z důležitých aspektů implementace jakéhokoliv algoritmu je použití vhodných a dostatečně rychlých funkcí. V dané práci jako hlavní výpočetní zdroj pro mou implementaci vystupuje knihovna se zaměřením na lineární algebru, která je součástí svobodné a otevřené softwarové knihovny SciPy. Velkou nevýhodou dokumentace SciPy je její malá informativnost. Z toho důvodu jsem se rozhodla krátce popsat mnou použité funkce z knihovny SciPy, abych nastínila jejich výpočetní role v mé implementaci.

3.2.1 scipy.linalg.blas.dsyrk

Vstupní parametry:

- **alpha** = skalár α;

- **a** = trojúhelníková matice A;

- **lower** = „*True*“, jestli A je dolní trojúhelníková matice, „*False*“, pokud A je horní trojúhelníková matice.

Vrací: výsledek součinu αAA^T.

3.2.2 scipy.linalg.blas.dtrsv

Vstupní parametry:

- **a** = trojúhelníková matice A;

- **x** = vektor x;

- **lower** = „*True*“, jestli A je dolní trojúhelníková matice, „*False*“, pokud A je horní trojúhelníková matice.

Vrací: výsledek součinu $A^{-1}x$, jinými slovy řešení y rovnice $Ay = x$.

3.2.3 scipy.linalg.cholesky

Vstupní parametry:

- **A** = hermitovská pozitivně definitní matice A;
- **x** = vektor x;
- **lower** = „*True*“, jestli výsledek musí být dolní trojúhelníkovou maticí, „*False*“, pokud výsledek musí být horní trojúhelníkovou maticí.

Vrací: dolní nebo horní trojúhelníková matice podle Choleského dekompozice $A = LU = LL^H = UU^H$, kde L^H znamená hermitovsky sdruženou matici anebo transponovanou, pokud A je reálná matice.

3.2.4 scipy.linalg.blas.dgemv

Vstupní parametry:

- **alpha** = skalár α;
- **a** = matice A;
- **x** = vektor x.

Vrací: výsledek součinu αAx.

3.2.5 scipy.linalg.blas.dgemm

Vstupní parametry:

- **alpha** = skalár α;
- **a** = matice A;
- **b** = matice B.

Vrací: výsledek součinu αAB.

Kapitola 4

Testovaní

Tato část bakalářské práce se věnuje testování implementace popsané v předešlé kapitole. Testování je založeno na porovnání klasifikačního modelu LDA s vertikální inkrementací s dvěma klasifikačními dávkovými verzemi LDA.

- **Scikit-learn** = třída LinearDiscriminantAnalysis se vstupním parametrem **solver**=„*lsqr*“, dávková klasifikace.
- **Offline** = třída LDAClassifier se vstupním parametrem **type**=„*offline*“, dávková klasifikace.
- **Online** = třída LDAClassifier se vstupním parametrem **type**=„*online*“, inkrementální klasifikace.

4.1 Datové sady

Pro selekci vhodných datových sad jsem použila veřejně přístupnou datovou platformu OpenML (Open Machine Learning). Pro filtraci datových sad jsem nastavila následující podmínky:

- počet vzorků v rozmezí od 20 do 1000;
- počet tříd v uzavřeném intervalu ode 2 do 10;
- maximální počet vlastností rovný 30;
- žádné chybějící hodnoty;
- pouze číselné hodnoty příznaků.

Na základě výše popsaných podmínek bylo odfiltrováno 30 datových sad. Popis vybraných datových sad je uveden v tabulce 4.1. Kromě základních veličin, jako jsou identifikátor datové sady, název datové sady, počet vzorků,

počet příznaků a počet tříd, obsahuje tabulka 4.1 maximální absolutní hodnotu Pearsonova korelačního koeficientu, tj. veličinu největší lineární závislosti mezi dvěma různými vektory příznaků.

Tabulka 4.1: Popis vybraných datových sad

ID	Název	Počet vzorků	Počet příznaků	Počet tříd	Pearsonův kor. koef.
11	balance-scale	625	4	3	0.0
37	diabetes	768	8	2	0.544
41	glass	214	9	6	0.81
54	vehicle	846	18	4	0.996
61	iris	150	4	3	0.963
187	wine	178	13	3	0.865
329	hayes-roth	160	4	3	0.054
464	prnn_synth	250	2	2	0.196
468	confidence	72	3	6	0.895
472	lupus	87	3	2	0.818
683	sleuth_ex2015	60	7	2	0.797
694	diggle_table_a2	310	8	9	0.998
874	rabe_131	50	5	4	0.629
894	rabe_148	66	5	2	0.75
969	iris	150	4	3	0.963
973	wine	178	13	3	0.865
974	hayes-roth	160	4	3	0.054
994	vehicle	846	18	4	0.996
997	balance-scale	625	4	3	0.0
1005	glass	214	9	6	0.81
1015	confidence	72	3	6	0.895
1048	jEdit_4.2_4.3	369	8	2	0.925
1060	ar3	63	29	2	1.0
1061	ar4	107	29	2	1.0
1062	ar5	36	29	2	1.0
1063	kc2	522	21	2	0.997
1064	ar6	101	29	2	1.0
1073	jEdit_4.0_4.2	274	8	2	0.949
1075	datatrieve	130	8	2	0.998
1117	desharnais	81	11	3	0.986

4.2 Příprava a rozdělení dat

Přepracování dat před testováním často zahrnuje standardizaci dat. Nejčastějším způsobem standardizace je standardizace směrodatnou odchylkou dle

vzorce

$$z = \frac{x - \hat{\mu}}{\hat{\delta}} \tag{4.1}$$

kde $\hat{\mu}$ je výběrový průměr a $\hat{\delta}$ je výběrová směrodatná odchylka. LDA je však na standardizaci invariantní [33], tj. výsledky na standardizovaných a nestandardizovaných datech jsou stejné. Z tohoto důvodu může být tento krok vynechán.

Dalším bodem je rozdělení dat. Z důvodu docela velkého počtu datových sad a specifiky inkrementálního přístupu bylo rozhodnuto použít nejjednodušší rozdělení dat pomocí `sklearn.model_selection.train_test_split`, tj. rozdělení jenom na dvě části:

1. trénovací množinu (tvoří 70% původních dat);

2. testovací množinu (tvoří 30% původních dat).

4.3 Regularizace

Pro získání porovnatelných výsledků jsem musela změnit zdrojový kód třídy LinearDiscriminantAnalysis tak, aby se regularizační vzorec používaný danou třídou shodoval se vzorcem používaným v mé implementaci (2.13). Jak vyplývá ze sekcí 2.6 a 2.7.3, primárním účelem regularizace v mé bakalářské práci je získání pozitivně definitního odhadu společné kovarianční matice pro splnění podmínek použití Choleského dekompozice, a proto daný parametr nemusí být optimalizován pro konkrétní datovou sadu. To znamená, že kvalita takové regularizace nezávisí na hodnotě regularizačního parametru, a proto stačí, aby regularizační parametr nabýval jakékoliv hodnoty z otevřeného intervalu $(0, 1)$, podmíníme-li případy, kde $\alpha > 0$, ale je tak malé, že kvůli výpočetním nepřesnostem regularizovaná matice stále nebude pozitivně definitní. Dalším zaměřením regularizace je získání dobře podmíněného odhadu společné kovarianční matice, abych se mohla vyhnout výpočetním chybám kvůli nestabilitě. Experimentálně bylo zjištěno, že regularizační parametr rovný **0.02** vyhovuje oběma cílům regularizace, a proto právě tato hodnota byla využita pro všechna testovaná data.

4.4 Měření času

Hlavní výhodou použití inkrementálního přístupu je časová „výhra“, která je založená na předpokladu, že obnovení vnitřní struktury modelu je časově efektivnější než opakované celkové přeučení modelu od začátku. Měření času je důležitým ukazatelem kvality inkrementálního klasifikátoru. V dané práci používám dvě časové metriky:

- doba trvání učení modelu na trénovacích datech, tj. součet doby vykonání metody **fit** a doby vykonání metody **add_feature**;
- doba trvání vytvoření predikce na trénovacích datech, tj. doba vykonání metody **predict**.

Obrázek 4.1: Algoritmus pro výpočet časových ukazatelů pro inkrementální klasifikaci

```
fit_time = 0
predict_time = 0
tmp = time.process_time()
model = model.fit(X_train[:, 0], y_train)
fit_time += time.process_time() - tmp
tmp = time.process_time()
model.predict(X_train[:, 0])
predict_time += time.process_time() - tmp
for i in range(1, X_train.shape[1]):
    tmp = time.process_time()
    model = model.add_feature(X_train[:, i])
    fit_time += time.process_time() - tmp
    tmp = time.process_time()
    model.predict(X_train[:, :i + 1])
    predict_time += time.process_time() - tmp
```

Obrázek 4.2: Algoritmus pro výpočet časových ukazatelů pro dávkovou klasifikaci

```
fit_time = 0
predict_time = 0
for i in range(0, X_train.shape[1]):
    tmp = time.process_time()
    model = model.fit(X_train[:,:i+1], y_train)
    fit_time += time.process_time() - tmp
    tmp = time.process_time()
    predicted_training = model.predict(X_train[:,:i+1])
    predict_time += time.process_time() - tmp
```

Samotné měření času se realizuje prostřednictvím algoritmů na obrázcích 4.1 a 4.2. Po zopakování daných algoritmů 10× konečné hodnoty časových

ukazatelů, tj. celková doba učení a celková doba vytvoření predikce, jsou vypočteny jako průměry z mezivýsledků pro daný klasifikační model.

4.5 Metriky

Kromě časových ukazatelů jsem použila následující metriky pro posouzení kvality implementovaného klasifikátoru [34].

- Klasifikační přesnost[9] pro binární klasifikaci uvádí poměr mezi vzorky, které byly predikovány jako kladné, a skutečnými pozitivními vzorky. V případě diskrétní klasifikace s předpokladem rovnováhy tříd je použitá mikroprůměrná přesnost $\frac{TP}{TP+FP}$, kde TP je celkový počet skutečných pozitivních hodnot pro všechny třídy a FP je celkový počet hodnot falešně predikovaných jako pozitivní pro všechny třídy.

- Výtěžnost[10] je míra podobná přesností, která pro binární klasifikaci počítá podíl skutečných pozitivních vzorků, které byly správně klasifikovány. V případě diskrétní klasifikace s předpokladem rovnováhy tříd je analogicky aplikován vzorec pro mikroprůměrnou výtěžnost $\frac{TP}{TP+FN}$, kde TP je celkový počet skutečných pozitivních hodnot pro všechny třídy a FP je celkový počet hodnot falešně predikovaných jako negativní pro všechny třídy.

- F_1 skóre závisí na hodnotách klasifikační přesnosti a výtěžnosti. Tuto závislost lze vyjádřit pomocí vzorce $F_1 = 2 * \frac{precision * recall}{precision + recall}$. Mikroprůměrné F_1 skóre se počítá stejně a ukazuje míru rovnováhy mezi mikroprůměrnou přesností a mikroprůměrné výtěžností.

- Cohenův koeficient kappa κ v podstatě vyjadřuje, kolikrát hodnocený klasifikátor podává lepší výkon než náhodný klasifikátor, který rozhoduje pouze na základě četnosti výskytu vzorců v každé třídě.

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{4.2}$$

  kde p_o je získaná hodnota a p_e je očekávaná hodnota. Klasifikátor s $\kappa \leq 0$ je nepoužitelný.

- AUC-ROC skóre měří plochu pod křivkou, která zobrazuje vztah mezi pravděpodobnostmi správné (osa y) a chybné (osa x) klasifikace skutečného pozitivního vzorku (výtěžnost a falešný poplach). Jinými slovy, AUC-ROC skóre určuje schopnost modelu rozlišovat mezi třídami. Pro diskrétní klasifikaci AUC-ROC skóre je počítán jako průměr ze všech možných kombinací tříd. AUC-ROC skóre musí nabývat alespoň 0.5, aby kvalita klasifikačního modelu byla považována za přijatelnou.

[9] `angl.` precision
[10] `angl.` recall

- Brierovo skóre je kalibrační metrika, která počítá střední kvadratickou chybu mezi predikovanými posteriorními pravděpodobnostmi a očekávanými hodnotami. Nejlepší hodnota Brierova skóre se rovná 0. Brierův skóre pro diskrétní klasifikační model lze vyjádřit ve tvaru

$$BS = \frac{1}{|X_{\text{test}}|} \sum_{i=1}^{|X_{\text{test}}|} \sum_{j=1}^{|K|} (p_{ij} - o_{ij})^2 \tag{4.3}$$

  kde $|X_{\text{test}}|$ znamená počet vzorků v testovací množině dat a p_{ij} znamená posteriorní pravděpodobnost, že vzorek x_i patří do třídy j.

$$o_{ij} = \begin{cases} 1 & \text{jestli } j = k \\ 0 & \text{jinak} \end{cases}$$

  Posteriorní pravděpodobnost p_{ij} nebo $p_k(x)$ lze vypočítat na základě vráceného diskriminačního skóre $S_k(x)$ pro jednotlivé třídy $k \in K$ pomocí aplikace exponenciální funkce.

$$p_k(x) = \frac{e^{S_k(x)}}{\sum_{i=1}^{|K|} e^{S_i(x)}} \tag{4.4}$$

Pro všechny výše popsané metriky s výjimkou Brierova skóre platí, že větší hodnota znamená lepší kvalitu klasifikátoru.

4.6 Výsledky testování na datových sadech

Níže jsou zobrazeny tabulky jednotlivých metrik popsaných v sekci 4.5. Každá tabulka obsahuje naměřené hodnoty odpovídající příslušnému klasifikátoru a identifikační údaje odfiltrovaných datových sad. Mikroprůměrná klasifikační přesnost na trénovacích datech, mikroprůměrná klasifikační přesnost na testovacích datech, mikroprůměrná výtěžnost na testovacích datech, mikroprůměrné F_1 skóre na testovacích datech, Cohenův koeficient kappa na testovacích datech, Brierovo skóre na testovacích datech a AUC-ROC skóre na testovacích datech se shodují pro jednotlivé testované implementace. To znamená, že inkrementální klasifikátor neustupuje dávkovému klasifikátoru z hlediska hodnocení predikčních schopností. Je důležité poznamenat, že konkrétní výsledky jsou velmi závislé na kvalitě datové sady, tj. nakolik daná datová sada splňuje požadavky LDA na vstupní data a nakolik je vhodná pro klasifikaci modelem LDA. Dobrým příkladem je tabulka 4.6, ze které lze vyčíst, že datová sada s identifikátorem 1062 není dobrou volbou pro klasifikátor na základě lineární diskriminační analýzy.

Dále jsou uvedeny dvě tabulky hodnot časových ukazatelů, naměřených dle algoritmů popsaných v sekci 4.4. Jelikož obnovení vnitřní struktury probíhá jenom během učení modelu, největší časovou „výhru“ inkrementálního

klasifikátoru nad dávkovou verzí lze zaznamenat v tabulce 4.9. Dle časových výsledků a popisu datových sad lze předpokládat, že velikost časové „výhry“ je spojená s celkovým počtem příznaků. Tento předpoklad je následovně ověřen na syntetických datech v sekci 4.7.

4.7 Syntetická data

Syntetická data jsou uměle vygenerovaná data pro specifické účely, která nebyla získána experimentálním změřením. V kontextu LDA lze mluvit o vygenerovaných datech, vhodných pro klasifikaci daným klasifikačním modelem. Výhodou syntetických dat jsou jejich v podstatě neomezené rozměry, nevýhodou – jejich umělá struktura a neschopnost odrazit složitou organizaci reálného světa. Knihovna scikit-learn nabízí funkci `sklearn.datasets.make_classification` k vytvoření matice dat na základě normálního rozdělení pro testování klasifikátorů. Mezi základní vstupní parametry patří:

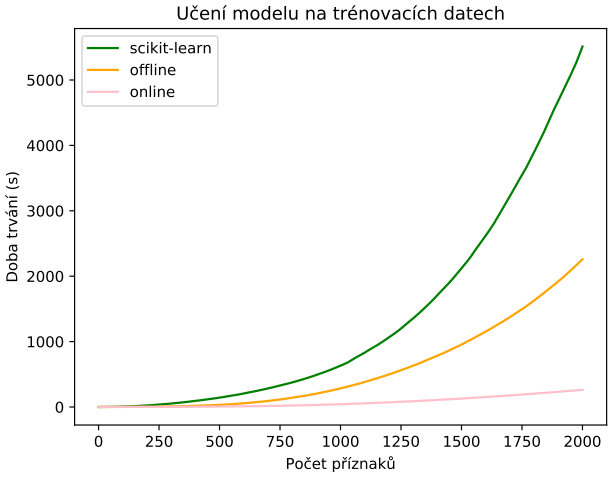
- **n_samples** = počet vzorků;
- **n_features** = počet příznaků;
- **n_informative** = počet informativních příznaků;
- **n_redundant** = počet přebytečných příznaků, které jsou lineárními kombinacemi náhodně vybraných informativních příznaků;
- **n_repeated** = počet duplicitních příznaků, které jsou náhodně vybrány z informativních a přebytečných příznaků;
- **n_classes** = počet tříd.

Na základě synteticky vygenerovaných datových sad jsem vytvořila dva grafy, které odráží závislost časových ukazatelů na počtu příznaků při fixním počtu vzorců (obrázky 4.3 a 4.4), dále jsem vytvořila dva grafy, které odráží závislost časových ukazatelů na počtu vzorků při fixním počtu příznaků (obrázky 4.5 a 4.6). Tyto grafy popisují časovou „výhru“ inkrementálního klasifikačního modelu nad dávkovou verzí. Podle naměřených hodnot dokáže vertikální inkrementace průměrně 10× zrychlit učení klasifikačního modelu. Například, při 2000 vzorcích a 500 příznacích „online“ LDA model provádí 5× rychlejší trénovaní oproti „offline“ LDA modelu a 15× rychlejší trénovaní oproti „sklearn“ LDA modelu. Při 3000 vzorcích a 2000 příznacích „online“ LDA model provádí 7× rychlejší trénovaní oproti „offline“ LDA modelu a 17× rychlejší trénovaní oproti „sklearn“ LDA modelu. Datové sady, na kterých bylo provedeno dané měření, byly vytvořeny pomocí příkazů:

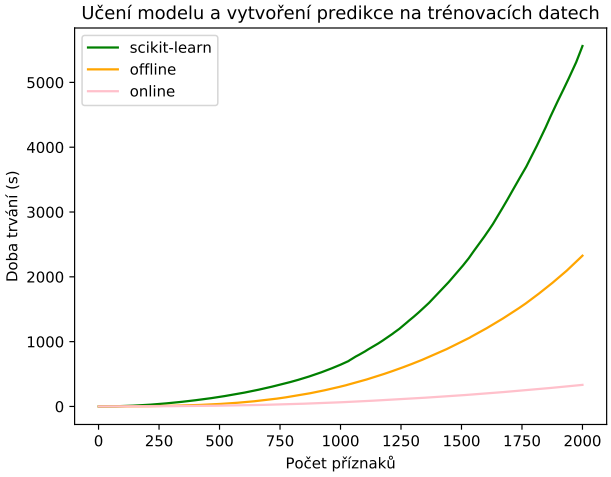
```
dataset_features = make_classification(n_samples=3000,
n_features=2000, n_informative=1500, n_redundant=300,
```

```
n_repeated=100, n_classes=15)
dataset_samples = make_classification(n_samples=2000,
n_features=500, n_informative=400, n_redundant=25,
n_repeated=25, n_classes=15)
```

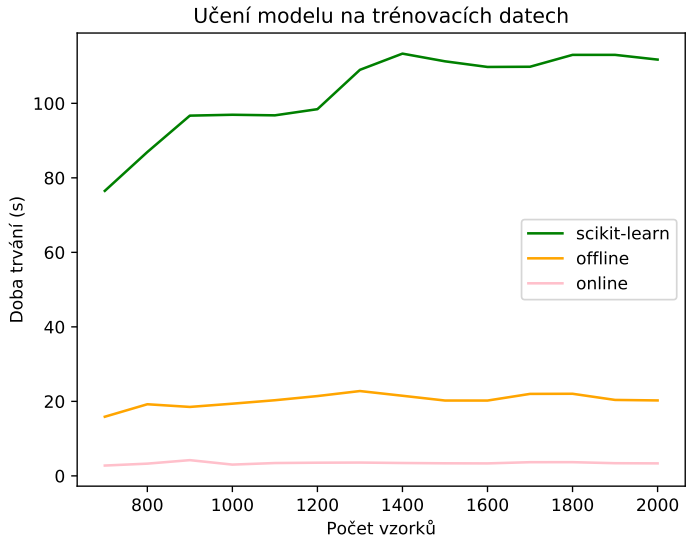
Obrázek 4.3: Porovnání časových výsledků (učení) LDA modelů v závislosti na množství příznaků



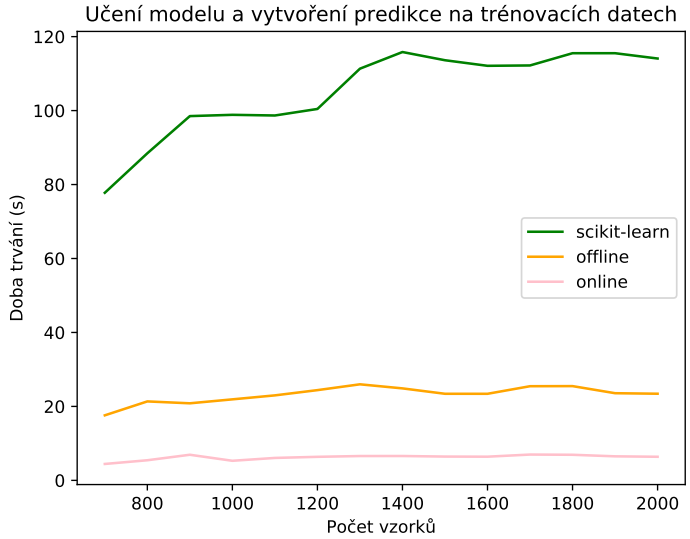
Obrázek 4.4: Porovnání časových výsledků (učení a vytvoření predikce) LDA modelů v závislosti na množství příznaků



Obrázek 4.5: Porovnání časových výsledků (učení) LDA modelů v závislosti na množství vzorků



Obrázek 4.6: Porovnání časových výsledků (učení a vytvoření predikce) LDA modelů v závislosti na množství vzorků



Tabulka 4.2: Mikroprůměrná klasifikační přesnost na trénovacích datech

ID	Název	Scikit-learn	Offline	Online
11	balance-scale	0.89016	0.89016	0.89016
37	diabetes	0.79143	0.79143	0.79143
41	glass	0.63087	0.63087	0.63087
54	vehicle	0.8125	0.8125	0.8125
61	iris	0.97143	0.97143	0.97143
187	wine	1.0	1.0	1.0
329	hayes-roth	0.61607	0.61607	0.61607
464	prnn_synth	0.85143	0.85143	0.85143
468	confidence	0.94	0.94	0.94
472	lupus	0.76667	0.76667	0.76667
683	sleuth_ex2015	0.78571	0.78571	0.78571
694	diggle_table_a2	1.0	1.0	1.0
874	rabe_131	0.94286	0.94286	0.94286
894	rabe_148	0.97826	0.97826	0.97826
969	iris	0.97143	0.97143	0.97143
973	wine	1.0	1.0	1.0
974	hayes-roth	0.61607	0.61607	0.61607
994	vehicle	0.8125	0.8125	0.8125
997	balance-scale	0.89016	0.89016	0.89016
1005	glass	0.63087	0.63087	0.63087
1015	confidence	0.94	0.94	0.94
1048	jEdit_4.2_4.3	0.64341	0.64341	0.64341
1060	ar3	0.95455	0.95455	0.95455
1061	ar4	0.87838	0.87838	0.87838
1062	ar5	1.0	1.0	1.0
1063	kc2	0.86849	0.86849	0.86849
1064	ar6	0.91429	0.91429	0.91429
1073	jEdit_4.0_4.2	0.70681	0.70681	0.70681
1075	datatrieve	0.92308	0.92308	0.92308
1117	desharnais	0.83929	0.83929	0.83929

Tabulka 4.3: Mikroprůměrná klasifikační přesnost na testovacích datech

ID	Název	Scikit-learn	Offline	Online
11	balance-scale	0.87766	0.87766	0.87766
37	diabetes	0.74459	0.74459	0.74459
41	glass	0.66154	0.66154	0.66154
54	vehicle	0.75591	0.75591	0.75591
61	iris	1.0	1.0	1.0
187	wine	0.98148	0.98148	0.98148
329	hayes-roth	0.54167	0.54167	0.54167
464	prnn_synth	0.82667	0.82667	0.82667
468	confidence	0.72727	0.72727	0.72727
472	lupus	0.77778	0.77778	0.77778
683	sleuth_ex2015	0.83333	0.83333	0.83333
694	diggle_table_a2	1.0	1.0	1.0
874	rabe_131	0.93333	0.93333	0.93333
894	rabe_148	0.8	0.8	0.8
969	iris	1.0	1.0	1.0
973	wine	0.98148	0.98148	0.98148
974	hayes-roth	0.54167	0.54167	0.54167
994	vehicle	0.75591	0.75591	0.75591
997	balance-scale	0.87766	0.87766	0.87766
1005	glass	0.66154	0.66154	0.66154
1015	confidence	0.72727	0.72727	0.72727
1048	jEdit_4.2_4.3	0.59459	0.59459	0.59459
1060	ar3	0.84211	0.84211	0.84211
1061	ar4	0.90909	0.90909	0.90909
1062	ar5	0.63636	0.63636	0.63636
1063	kc2	0.79618	0.79618	0.79618
1064	ar6	0.80645	0.80645	0.80645
1073	jEdit_4.0_4.2	0.71084	0.71084	0.71084
1075	datatrieve	0.92308	0.92308	0.92308
1117	desharnais	0.6	0.6	0.6

Tabulka 4.4: Mikroprůměrná výtěžnost na testovacích datech

ID	Název	Scikit-learn	Offline	Online
11	balance-scale	0.87766	0.87766	0.87766
37	diabetes	0.74459	0.74459	0.74459
41	glass	0.66154	0.66154	0.66154
54	vehicle	0.75591	0.75591	0.75591
61	iris	1.0	1.0	1.0
187	wine	0.98148	0.98148	0.98148
329	hayes-roth	0.54167	0.54167	0.54167
464	prnn_synth	0.82667	0.82667	0.82667
468	confidence	0.72727	0.72727	0.72727
472	lupus	0.77778	0.77778	0.77778
683	sleuth_ex2015	0.83333	0.83333	0.83333
694	diggle_table_a2	1.0	1.0	1.0
874	rabe_131	0.93333	0.93333	0.93333
894	rabe_148	0.8	0.8	0.8
969	iris	1.0	1.0	1.0
973	wine	0.98148	0.98148	0.98148
974	hayes-roth	0.54167	0.54167	0.54167
994	vehicle	0.75591	0.75591	0.75591
997	balance-scale	0.87766	0.87766	0.87766
1005	glass	0.66154	0.66154	0.66154
1015	confidence	0.72727	0.72727	0.72727
1048	jEdit_4.2_4.3	0.59459	0.59459	0.59459
1060	ar3	0.84211	0.84211	0.84211
1061	ar4	0.90909	0.90909	0.90909
1062	ar5	0.63636	0.63636	0.63636
1063	kc2	0.79618	0.79618	0.79618
1064	ar6	0.80645	0.80645	0.80645
1073	jEdit_4.0_4.2	0.71084	0.71084	0.71084
1075	datatrieve	0.92308	0.92308	0.92308
1117	desharnais	0.6	0.6	0.6

Tabulka 4.5: F_1 skóre

ID	Název	Scikit-learn	Offline	Online
11	balance-scale	0.87766	0.87766	0.87766
37	diabetes	0.74459	0.74459	0.74459
41	glass	0.66154	0.66154	0.66154
54	vehicle	0.75591	0.75591	0.75591
61	iris	1.0	1.0	1.0
187	wine	0.98148	0.98148	0.98148
329	hayes-roth	0.54167	0.54167	0.54167
464	prnn_synth	0.82667	0.82667	0.82667
468	confidence	0.72727	0.72727	0.72727
472	lupus	0.77778	0.77778	0.77778
683	sleuth_ex2015	0.83333	0.83333	0.83333
694	diggle_table_a2	1.0	1.0	1.0
874	rabe_131	0.93333	0.93333	0.93333
894	rabe_148	0.8	0.8	0.8
969	iris	1.0	1.0	1.0
973	wine	0.98148	0.98148	0.98148
974	hayes-roth	0.54167	0.54167	0.54167
994	vehicle	0.75591	0.75591	0.75591
997	balance-scale	0.87766	0.87766	0.87766
1005	glass	0.66154	0.66154	0.66154
1015	confidence	0.72727	0.72727	0.72727
1048	jEdit_4.2_4.3	0.59459	0.59459	0.59459
1060	ar3	0.84211	0.84211	0.84211
1061	ar4	0.90909	0.90909	0.90909
1062	ar5	0.63636	0.63636	0.63636
1063	kc2	0.79618	0.79618	0.79618
1064	ar6	0.80645	0.80645	0.80645
1073	jEdit_4.0_4.2	0.71084	0.71084	0.71084
1075	datatrieve	0.92308	0.92308	0.92308
1117	desharnais	0.6	0.6	0.6

Tabulka 4.6: Cohenův koeficient kappa na testovacích datech

ID	Název	Scikit-learn	Offline	Online
11	balance-scale	0.77261	0.77261	0.77261
37	diabetes	0.40931	0.40931	0.40931
41	glass	0.50977	0.50977	0.50977
54	vehicle	0.67437	0.67437	0.67437
61	iris	1.0	1.0	1.0
187	wine	0.97214	0.97214	0.97214
329	hayes-roth	0.28649	0.28649	0.28649
464	prnn_synth	0.65389	0.65389	0.65389
468	confidence	0.67488	0.67488	0.67488
472	lupus	0.5	0.5	0.5
683	sleuth_ex2015	0.65823	0.65823	0.65823
694	diggle_table_a2	1.0	1.0	1.0
874	rabe_131	0.90909	0.90909	0.90909
894	rabe_148	0.61538	0.61538	0.61538
969	iris	1.0	1.0	1.0
973	wine	0.97214	0.97214	0.97214
974	hayes-roth	0.28649	0.28649	0.28649
994	vehicle	0.67437	0.67437	0.67437
997	balance-scale	0.77261	0.77261	0.77261
1005	glass	0.50977	0.50977	0.50977
1015	confidence	0.67488	0.67488	0.67488
1048	jEdit_4.2_4.3	0.1999	0.1999	0.1999
1060	ar3	0.31325	0.31325	0.31325
1061	ar4	0.67327	0.67327	0.67327
1062	ar5	-0.22222	-0.22222	-0.22222
1063	kc2	0.37216	0.37216	0.37216
1064	ar6	0.13889	0.13889	0.13889
1073	jEdit_4.0_4.2	0.42461	0.42461	0.42461
1075	datatrieve	0.0	0.0	0.0
1117	desharnais	0.32615	0.32615	0.32615

Tabulka 4.7: Brierovo skóre na testovacích datech

ID	Název	Scikit-learn	Offline	Online
11	balance-scale	0.15436	0.15436	0.15436
37	diabetes	0.33019	0.33019	0.33019
41	glass	0.51159	0.51159	0.51159
54	vehicle	0.31514	0.31514	0.31514
61	iris	0.00727	0.00727	0.00727
187	wine	0.01923	0.01923	0.01923
329	hayes-roth	0.49973	0.49973	0.49973
464	prnn_synth	0.2305	0.2305	0.2305
468	confidence	0.46499	0.46499	0.46499
472	lupus	0.27272	0.27272	0.27272
683	sleuth_ex2015	0.34395	0.34395	0.34395
694	diggle_table_a2	0.0	0.0	0.0
874	rabe_131	0.08475	0.08475	0.08475
894	rabe_148	0.27958	0.27958	0.27958
969	iris	0.00727	0.00727	0.00727
973	wine	0.01923	0.01923	0.01923
974	hayes-roth	0.49973	0.49973	0.49973
994	vehicle	0.31514	0.31514	0.31514
997	balance-scale	0.15436	0.15436	0.15436
1005	glass	0.51159	0.51159	0.51159
1015	confidence	0.46499	0.46499	0.46499
1048	jEdit_4.2_4.3	0.46599	0.46599	0.46599
1060	ar3	0.29615	0.29615	0.29615
1061	ar4	0.12699	0.12699	0.12699
1062	ar5	0.72727	0.72727	0.72727
1063	kc2	0.32556	0.32556	0.32556
1064	ar6	0.28615	0.28615	0.28615
1073	jEdit_4.0_4.2	0.33367	0.33367	0.33367
1075	datatrieve	0.13403	0.13403	0.13403
1117	desharnais	0.61983	0.61983	0.61983

Tabulka 4.8: AUC-ROC skóre na testovacích datech

ID	Název	Scikit-learn	Offline	Online
11	balance-scale	0.93782	0.93782	0.93782
37	diabetes	0.32856	0.32856	0.32856
41	glass	0.87285	0.87285	0.87285
54	vehicle	0.94094	0.94094	0.94094
61	iris	1.0	1.0	1.0
187	wine	1.0	1.0	1.0
329	hayes-roth	0.80611	0.80611	0.80611
464	prnn_synth	0.48629	0.48629	0.48629
468	confidence	0.96667	0.96667	0.96667
472	lupus	0.4321	0.4321	0.4321
683	sleuth_ex2015	0.5	0.5	0.5
694	diggle_table_a2	1.0	1.0	1.0
874	rabe_131	0.98611	0.98611	0.98611
894	rabe_148	0.54167	0.54167	0.54167
969	iris	1.0	1.0	1.0
973	wine	1.0	1.0	1.0
974	hayes-roth	0.80611	0.80611	0.80611
994	vehicle	0.94094	0.94094	0.94094
997	balance-scale	0.93782	0.93782	0.93782
1005	glass	0.87285	0.87285	0.87285
1015	confidence	0.96667	0.96667	0.96667
1048	jEdit_4.2_4.3	0.61776	0.61776	0.61776
1060	ar3	0.77083	0.77083	0.77083
1061	ar4	0.37143	0.37143	0.37143
1062	ar5	0.66667	0.66667	0.66667
1063	kc2	0.29252	0.29252	0.29252
1064	ar6	0.34259	0.34259	0.34259
1073	jEdit_4.0_4.2	0.39744	0.39744	0.39744
1075	datatrieve	0.17593	0.17593	0.17593
1117	desharnais	0.57315	0.57315	0.57315

Tabulka 4.9: Doba trvání učení modelu na trénovacích datech

ID	Název	Scikit-learn	Offline	Online
11	balance-scale	0.00833	0.00955	0.00783
37	diabetes	0.02598	0.01706	0.01133
41	glass	0.02175	0.04089	0.02868
54	vehicle	0.05778	0.11008	0.07379
61	iris	0.00592	0.00905	0.00756
187	wine	0.02733	0.04107	0.02326
329	hayes-roth	0.0078	0.01124	0.00721
464	prnn_synth	0.00308	0.00379	0.00276
468	confidence	0.00608	0.01435	0.01083
472	lupus	0.00403	0.00536	0.00377
683	sleuth_ex2015	0.00967	0.01391	0.00754
694	diggle_table_a2	0.02329	0.05111	0.03594
874	rabe_131	0.00658	0.01896	0.01339
894	rabe_148	0.00507	0.00957	0.00765
969	iris	0.00551	0.00906	0.00739
973	wine	0.01913	0.03135	0.02323
974	hayes-roth	0.00558	0.00908	0.00768
994	vehicle	0.05873	0.10198	0.06469
997	balance-scale	0.00822	0.00971	0.00687
1005	glass	0.02096	0.04571	0.03591
1015	confidence	0.01203	0.02506	0.01301
1048	jEdit_4.2_4.3	0.0113	0.03461	0.02464
1060	ar3	0.04126	0.08774	0.04487
1061	ar4	0.03918	0.08579	0.04315
1062	ar5	0.03541	0.06527	0.04341
1063	kc2	0.05562	0.04998	0.03218
1064	ar6	0.04019	0.05929	0.05036
1073	jEdit_4.0_4.2	0.01072	0.02426	0.01179
1075	datatrieve	0.01131	0.01496	0.0117
1117	desharnais	0.01505	0.03347	0.0226

Tabulka 4.10: Doba trvání vytvoření predikce modelem na trénovacích datech

ID	Název	Scikit-learn	Offline	Online
11	balance-scale	0.00052	0.00024	0.00028
37	diabetes	0.00173	0.00055	0.00058
41	glass	0.00119	0.00072	0.00069
54	vehicle	0.00295	0.00391	0.00399
61	iris	0.00048	0.00018	0.00021
187	wine	0.00223	0.00092	0.00074
329	hayes-roth	0.00056	0.00022	0.0002
464	prnn_synth	0.00025	0.0001	9e-05
468	confidence	0.00035	0.00022	0.00021
472	lupus	0.00041	0.00013	0.00011
683	sleuth_ex2015	0.00096	0.00034	0.00026
694	diggle_table_a2	0.00101	0.00088	0.00087
874	rabe_131	0.00046	0.00026	0.00026
894	rabe_148	0.00054	0.00019	0.00022
969	iris	0.00045	0.00019	0.00021
973	wine	0.00151	0.00069	0.00081
974	hayes-roth	0.00038	0.00019	0.00021
994	vehicle	0.00299	0.00364	0.00361
997	balance-scale	0.00056	0.00024	0.00024
1005	glass	0.00128	0.00084	0.00096
1015	confidence	0.00069	0.00045	0.00027
1048	jEdit_4.2_4.3	0.00111	0.00085	0.00075
1060	ar3	0.00456	0.00198	0.00142
1061	ar4	0.00412	0.00185	0.0014
1062	ar5	0.00373	0.00117	0.00112
1063	kc2	0.00385	0.00193	0.00179
1064	ar6	0.00403	0.00122	0.00164
1073	jEdit_4.0_4.2	0.00109	0.00071	0.00037
1075	datatrieve	0.00107	0.00038	0.00046
1117	desharnais	0.00139	0.00052	0.00051

Závěr

Hlavním cílem této práce bylo navrhnout a implementovat diskrétní inkrementální klasifikátor na základě lineární diskriminační analýzy, který by se dokázal přiučit na nových příznacích. Dalším požadavkem na klasifikátor byla jeho schopnost poradit si s problémem multikolinearity. Pro naplnění tohoto cíle byla provedena analýza existujícího dávkového řešení. Na základě této analýzy byl navržen algoritmus efektivního obnovení vnitřní struktury klasifikačního modelu pomocí Choleského dekompozice s použitím konstantní regularizace. Analýza a navržený inkrementální postup jsou nejdůležitějšími částmi této práce, protože povolují vytvořit vertikálně inkrementální klasifikátor LDA prostřednictvím jakékoliv vhodné technologie.

Následně byla vytvořená implementace v jazyce Python a bylo provedeno testování této implementace na reálných datových sadách a syntetických datech. Výsledky testování ukazují, že vertikální inkrementace v kontextu LDA má potenciál, jelikož může nabídnout 10× rychlejší učení klasifikačního modelu oproti dávkovému analogu při stejné kvalitě predikčních schopností. Nevýhodou vertikálně inkrementální LDA jsou přísné požadavky na vstupní data, zdědění všech nedostatků dávkové verze a omezení využití konceptu regularizace. Cíle definované v kapitole 1 byly splněny.

Bibliografie

1. HORÁKOVÁ, Eliška. *Robustní metody v diskriminační analýze* [online]. Brno, 2008 [cit. 2020-04-01]. Dostupné z: `https://is.muni.cz/th/c92a7/Diplomka.pdf`. Diplomová práce. Masarykova Univerzita, Přírodov˘edecká fakulta. Vedoucí práce Marie FORBELSKÁ.
2. NAVRÁTILOVÁ, Simona. *Moderní metody diskriminační analýzy* [online]. Brno, 2019 [cit. 2020-04-01]. Dostupné z: `https://is.muni.cz/th/h55rf/DP_navratilova_simona.pdf`. Diplomová práce. Masarykova Univerzita, Přírodov˘edecká fakulta, Ústav matematiky a statistiky. Vedoucí práce Radim NAVRÁTIL.
3. TJEN-SIEN, Lim; WEI-YIN, Loh; SHIH, Yu-Shan. A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. *Machine Learning* [online]. 2000, roč. 3, č. 40, s. 203–228 [cit. 2020-04-05]. ISSN 1573-0565. Dostupné z DOI: `10.1023/A:1007608224229`.
4. MISAKI, M.; KIM, Y.; BANDETTINI, P. A.; KRIEGESKORTE, N. Comparison of multivariate classifiers and response normalizations for pattern-information fMRI. *NeuroImage* [online]. 2010, roč. 53, č. 1, s. 103–118 [cit. 2020-04-05]. ISSN 10538119. Dostupné z DOI: `10.1016/j.neuroimage.2010.05.051`.
5. ÜNSAL, Mehmet Güray; NAZMAN, Ezgi. Investigating socio-economic ranking of cities in Turkey using data envelopment analysis (DEA) and linear discriminant analysis (LDA). *Annals of Operations Research* [online]. 2018, s. 1–15 [cit. 2020-03-30]. ISSN 15729338. Dostupné z DOI: `10.1007/s10479-017-2748-0`.
6. HUANG, Mei Ling; CHEN, Hsin Yi; HUANG, Wei Cheng; TSAI, Yi Yu. Linear discriminant analysis and artificial neural network for glaucoma diagnosis using scanning laser polarimetry-variable cornea compensation measurements in Taiwan Chinese population. *Graefe's Archive for Cli-*

nical and Experimental Ophthalmology [online]. 2010, roč. 248, č. 3, s. 435–441 [cit. 2020-03-30]. ISSN 0721832X. Dostupné z DOI: 10.1007/s00417-009-1259-3.

7. SU, Yuting; LI, Yang; LIU, Anan. Open-view human action recognition based on linear discriminant analysis. *Multimedia Tools and Applications* [online]. 2019, roč. 78, č. 1, s. 767–782 [cit. 2020-03-30]. ISSN 15737721. Dostupné z DOI: 10.1007/s11042-018-5657-6.

8. SIDDIQI, Muhammad Hameed; ALI, Rahman; KHAN, Adil Mehmood; PARK, Young Tack; LEE, Sungyoung. Human Facial Expression Recognition Using Stepwise Linear Discriminant Analysis and Hidden Conditional Random Fields. *IEEE Transactions on Image Processing* [online]. 2015, roč. 24, č. 4, s. 1386–1398 [cit. 2020-03-30]. ISSN 10577149. Dostupné z DOI: 10.1109/TIP.2015.2405346.

9. WU, Michael C.; ZHANG, Lingsong; WANG, Zhaoxi; CHRISTIANI, David C.; LIN, Xihong. Sparse linear discriminant analysis for simultaneous testing for the significance of a gene set/pathway and gene selection. *Bioinformatics* [online]. 2009, roč. 25, č. 9, s. 1145–1151 [cit. 2020-03-30]. ISSN 13674803. Dostupné z DOI: 10.1093/bioinformatics/btp019.

10. YANG, Jung Gi; KIM, Jae Kwon; KANG, Un Gu; LEE, Young Ho. Coronary heart disease optimization system on adaptive-network-based fuzzy inference system and linear discriminant analysis (ANFIS-LDA). *Personal and Ubiquitous Computing* [online]. 2014, roč. 18, č. 6, s. 1351–1362 [cit. 2020-03-30]. ISSN 16174909. Dostupné z DOI: 10.1007/s00779-013-0737-0.

11. TUNG, Ka Kit; CAMP, Charles D. Solar cycle warming at the Earth's surface in NCEP and ERA-40 data: A linear discriminant analysis. *Journal of Geophysical Research Atmospheres* [online]. 2008, roč. 113, č. 5, s. 1–11 [cit. 2020-03-30]. ISSN 01480227. Dostupné z DOI: 10.1029/2007JD009164.

12. VÍTKOVÁ, Gabriela; NOVOTNÝ, Karel; PROKEŠ, Lubomír; HRDLIČKA, Aleš; KAISER, Jozef; NOVOTNÝ, Jan; MALINA, Radomír; PROCHAZKA, David. Fast identification of biominerals by means of stand-off laser-induced breakdown spectroscopy using linear discriminant analysis and artificial neural networks. *Spectrochimica Acta - Part B Atomic Spectroscopy* [online]. 2012, roč. 73, s. 1–6 [cit. 2020-03-30]. ISSN 05848547. Dostupné z DOI: 10.1016/j.sab.2012.05.010.

13. HÄRDLE, Wolfgang Karl; SIMAR, Léopold. Discriminant Analysis. In: *Applied Multivariate Statistical Analysis* [online]. 4. vyd. Berlin, Heidelberg: Springer, 2015, s. 407–425 [cit. 2020-03-15]. ISBN 978-3-662-45171-7. Dostupné z DOI: 10.1007/978-3-662-45171-7.

14. RADHAKRISHNA RAO, C. Discriminatory Analysis (Identification). In: *Linear Statistical Inference and its Applications*. 2. vyd. USA: John Wiley & Sons, 2002, s. 407–425. Wiley Series in Probability and Statistics. ISBN 978-0-471-21875-3.

15. RIPLEY, B. D. Bayes rules for known distributions. In: *Pattern Recognition and Neural Networks*. Cambridge, United Kingdom: Cambridge University Press, 1996, s. 18–26. ISBN 0-521-46086-7.

16. FRIEDMAN, Jerome H. Regularized discriminant analysis. *Journal of the American Statistical Association* [online]. 1989, roč. 84, č. 405, s. 165–175 [cit. 2020-04-05]. ISSN 1537274X. Dostupné z DOI: 10.1080/01621459.1989.10478752.

17. DUDA, Richard O.; HART, Peter E.; STORK, David G. Multiple Discriminant Analysis. In: *Pattern Classificatio*. 2. vyd. New York: Wiley-Interscience, 2000, s. 47–51. ISBN 978-0-471-05669-0.

18. RIPLEY, B. D. Linear Discriminant Analysis. In: *Pattern Recognition and Neural Networks*. Cambridge, United Kingdom: Cambridge University Press, 1996, s. 91–121. ISBN 0-521-46086-7.

19. *Linear and Quadratic Discriminant Analysis with covariance ellipsoid* [online]. 2007 - 2019, scikit-learn developers (BSD License) [cit. 2020-04-12]. Dostupné z: https://scikit-learn.org/stable/auto_examples/classification/plot_lda_qda.html.

20. HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. Linear Discriminant Analysis. In: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* [online]. 2. vyd. New York: Springer, 2009, s. 106–113 [cit. 2020-04-10]. Springer Series in Statistics. ISBN 978-0-387-84858-7. Dostupné z DOI: 10.1007/978-0-387-84858-7.

21. *sklearn.discriminant_analysis.LinearDiscriminantAnalysis* [online]. 2007 - 2019, scikit-learn developers (BSD License) [cit. 2020-04-12]. Dostupné z: https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html.

22. LEDOIT, Olivier; WOLF, Michael. Honey, I Shrunk the Sample Covariance Matrix. *The Journal of Portfolio Management* [online]. 2003, roč. 4, č. 30, s. 110–119 [cit. 2020-04-05]. Dostupné z DOI: 10.2139/ssrn.433840.

23. DUDA, Richard O.; HART, Peter E.; STORK, David G. Minimum Squared Error Procedures. In: *Pattern Classificatio*. 2. vyd. New York: Wiley-Interscience, 2000, s. 28–37. ISBN 978-0-471-05669-0.

24. FISHER, Ronald Aylmer. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* [online]. 1936, č. 7, s. 179–188 [cit. 2020-04-05]. Dostupné z DOI: 10.1111/j.1469-1809.1936.tb02137.x.

25. KALINA, Jan; VALENTA, Zdeněk; TEBBENS, Jurjen Duintjer. Computation of Regularized Linear Discriminant Analysis. *Comput. Stat. Int. Conf.* [online]. 2015, s. 128–133 [cit. 2020-04-05]. Dostupné z: http://www.cs.cas.cz/duintjertebbens/pubs/Compstat2014.pdf.

26. YAQIAN GUO; TREVOR HASTIE; ROBERT TIBSHIRANI. Regularized Discriminant Analysis and Its Application in Microarrays. *Biostatistics* [online]. 2005, roč. 1, č. 1, s. 1–18 [cit. 2020-04-05]. Dostupné z DOI: 10.1155/2015/797280.

27. *How to Draw Ellipse of Covariance Matrix* [online] [cit. 2020-06-20]. Dostupné z: https://cookierobotics.com/007/.

28. LEDOIT, Olivier; WOLF, Michael. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance* [online]. 2001, roč. 5, č. 10, s. 603–621 [cit. 2020-04-05]. ISSN 09275398. Dostupné z DOI: 10.1016/S0927-5398(03)00007-0.

29. *Normal and Shrinkage Linear Discriminant Analysis for classification* [online]. 2007 - 2019, scikit-learn developers (BSD License) [cit. 2020-06-15]. Dostupné z: https://scikit-learn.org/stable/auto_examples/classification/plot_lda.html.

30. PANG, Shaoning; OZAWA, Seiichi; KASABOV, Nikola. Incremental linear discriminant analysis for classification of data streams. *IEEE Trans. Syst. Man. Cybern.* [online]. 2005, roč. 35, č. 5, s. 905–914 [cit. 2020-04-07]. ISSN 10834419. Dostupné z DOI: 10.1109/TSMCB.2005.847744.

31. CHU, Delin; LIAO, Li Zhi; NG, Michael Kwok Po; WANG, Xiaoyan. Incremental Linear Discriminant Analysis: A Fast Algorithm and Comparisons. *IEEE Transactions on Neural Networks and Learning Systems* [online]. 2015, roč. 26, č. 11, s. 2716–2735 [cit. 2020-03-30]. ISSN 21622388. Dostupné z DOI: 10.1109/TNNLS.2015.2391201.

32. KRISHNAMOORTHY, Aravindh; MENON, Deepak. Matrix inversion using Cholesky decomposition. *Signal Processing - Algorithms, Architectures, Arrangements, and Applications Conference Proceedings, SPA* [online]. 2013, č. 3, s. 70–72 [cit. 2020-04-05]. ISSN 23260262. Dostupné z: https://arxiv.org/pdf/1111.4144.pdf.

33. RASCHKA, Sebastian. *Linear Discriminant Analysis: Bit by Bit* [online]. 2013-2020 Sebastian Raschka, 2014 [cit. 2020-07-05]. Dostupné z: https://sebastianraschka.com/Articles/2014_python_lda.html#lda-in-5-steps.

34. *Metrics and scoring: quantifying the quality of predictions* [online]. 2007 - 2019, scikit-learn developers (BSD License) [cit. 2020-07-15]. Dostupné z: https://scikit-learn.org/stable/modules/model_evaluation.html.

Příloha **A**

Seznam použitých zkratek

DA Diskriminační analýza

LDA Lineární diskriminační analýza

QDA Kvadratická diskriminační analýza

RDA Regularizovaná diskriminační analýza

PCA Analýza hlavních komponent

Příloha B

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `LDAClassifier.py` ..... zdrojový kód implementace klasifikátoru
  - `tests_openml.py` ..... zdrojový kód testování na datových sadech
  - `tests_synt.py` ..... zdrojový kód testování na syntetických datech
  - `tests_openml_assert.py` .. zdrojový kód porovnání vnitřní struktury
  - `thesis.tex` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF