Zadání Cvičení #11

Popis dat: Pracovní data jsou uložena v souboru data.csv, který je k dispozici ke stažení na Moodle stránce tohoto předmětu, ve složce příslušného cvičení. Pro načtení dat do Matlabu využijte funkce `readtable`.

Data jsou ve formátu tabulky, která obsahuje data ***pouze*** od pacientů s Parkinsonovou nemocí (label **PD**). Tabulka obsahuje ID kódy subjektů a soubor parametrů, které již znáte z minulých cvičení:

- Klinická hodnocení neurologem prostřednictvím škály UPDRS III (bez jednotky):
  - **UPDRS III axial**: souhrnné skóre z položek, které hodnotí axiální[1] motoriku.
  - **UPDRS III bradykinesia**: souhrnné skóre z položek, které hodnotí *bradykinezii*[2].
  - **UPDRS III tremor:** souhrnné skóre z položek, které hodnotí klidový třes.
  - **UPDRS III rigidity:** souhrnné skóre z položek, které hodnotí ztuhlost.

Všechny parametry byly naměřeny či vyhodnoceny před (**label U** – *Untreated*) a po podání léku (**label T** – *Treated*). Všechny 4 parametry mají vyšší hodnotu, pokud byly příznaky nemoci, popisované jednotlivými parametry, u pacienta vyhodnoceny jako horší, tj. vyšší hodnota – horší stav.

Tohle je poslední cvičení, díky všem za dobrou práci po celý semestr! 😊

[1] Axiální – týkající se těch částí těla, které se nachází na svislé ose (hlava, krk, hrudník, trup, …), opak je apendikulární (týkající se končetin)

[2] *Bradykinezie* – zpomalení pohybů, snížení rozsahu a rychlosti pohybů, těžší cílená iniciace pohybů

Zadání úlohy	body
Použijte všechny 4 dostupné hodnocení UPDRS pro rozdělení dostupných dat do dvou skupin – clusterů. Cílem by bylo např. charakterizovat skupinu s dobou a špatnou odezvou na léky. • Tyto clustery by měly reprezentovat odpověď na léčbu, proto pro každé hodnocení vypočtěte **rozdíl hodnot před a po podání léku (Δ delta)**. • Pro *clustering* použijte metodu K-means. Neprogramujte ji manuálně (to si můžete vyzkoušet v bonusu), **využijte funkci `kmeans`**. • Algoritmus K-means **inicializujte náhodně**, a proveďte **5 samostatných iterací** celého clusteringu, ze kterých pak vyberete nejlepší výsledek. ○ *Nápověda*: vše se dá zvládnout jen pomocí funkce `kmeans` s parametry. • Výsledné labely bodů a souřadnice centroidů vložte společně se vstupními daty do funkce `visualizeKmeans` (k dispozici ke stažení na Moodlu), která provede vizualizaci dat jak pomocí 3D scatter grafu všech permutací 4 parametrů, tak pomocí 2D matice grafů všech kombinací parametrů. **Stručně slovy zhodnoťte výsledky clusteringu (separovatelnost dat, velikost výsledných clusterů, rozdíly mezi iteracemi algoritmu, …).** **Odpovězte také na tyto otázky:** • *Proč je dobré celý clusterovací algoritmus několikrát opakovat?* • *Jak byste manuálně nastavili počáteční podmínky algoritmu, tak aby algoritmus co nejrychleji zkonvergoval s co nejlepšími výsledky?* • *Jak byste nastavili hranici pro úspěšné skončení clusterovacího algoritmu?*	1

Vytvořte *Gaussian Mixture Model* (GMM), který bude modelovat skupiny ve vašich datech pomocí směsi (dvou) Gaussovských rozdělení.
Pro další pokračování si vyberte libovolně **2 z poskytnutých parametrů**, se kterými budete dále pracovat (z důvodů snadné 2D vizualizace).
Použijte opět **delta hodnoty Δ parametrů**, které budou, stejně jako v předchozím bodě, reprezentovat efekt léků.
• Na základě delta hodnot vybraných 2 parametrů připravte GMM. **Neprogramujte proces manuálně** (to si lze opět vyzkoušet v bonusové úloze), použijte funkce:
  ○ `fitgmdist` pro výpočet modelu (pomocí EM-algoritmu a maximalizace likelihoodu)
  ○ `cluster` pro přiřazení labelů k jednotlivým datovým bodům na základě vypočteného GMM
• Stejně jako u K-means celý proces několikrát opakujte (např. 10x) s **náhodnou inicializací** a vyberte nejlepší výsledek.
  ○ Opět se vše dá zvládnout pouze pomocí argumentů funkce `fitgmdist`.
	1.5

- Vyzkoušejte si při výpočtu GMM různá nastavení kovariančních matic pro komponenty GMM: diagonální/plná či sdílená/nesdílená matice.
- V případě, že by měl algoritmus pro návrh modelu problémy s konvergencí (poznali byste s varovných výpisů v konzoli), nastavte parametr `RegularizationValue` na hodnotu 0.01. Tato malá hodnota se přičte k diagonále kovarianční matice tak, aby mohla být pro potřeby algoritmu vypočtena její inverze.

Pro vybraná data také vypočítejte také clustery pomocí K-means.
Výsledky clusteringu pro obě metody zobrazte vedle sebe do jednoho obrázku.

- Vhodným způsobem vyznačte příslušnost jednotlivých bodu k jedné či druhé skupině tak, jak jí přiřadil clustering.
- Vykreslete do obrázku také centroidy obou clusterů.
- Pro Gaussovský model také vykreslete do obrázku kontury jednotlivých pravděpodobnostních úrovní, případně můžete vykreslit celý obrázek ve 3D.
    - Pro vizualizaci pravděpodobnostního rozdělení z GMM použijte funkci `fcontour`. Příklady použití najdete v dokumentaci k `fitgmdist` a `cluster`.
- Pokuste se nastavit kovarianční matici Gaussovského modelu tak, aby výsledek clusteringu co nejvíce odpovídal výsledku K-means.

Odpovězte také slovy na tyto otázky:

- *Jaké jsou hlavní rozdíly mezi clusteringem pomocí K-means a pomocí GMM?*
- *Jaká jsou úskalí pro jednotlivá nastavení kovarianční matice při výpočtu modelu?*

Zkombinujte vědomosti z tohoto a z předešlého cvičení a proveďte klasifikační experiment. Zkuste predikovat, jak bude pacient odpovídat na léčbu jen na základě hodnot všech UPDRS parametrů vyhodnocených <u>před podáním léku.</u>
• Použijte clustering pro **rozřazení dat pro trénování SVM modelu do dvou skupin:** a) Skupiny, která reaguje na léčbu dobře, b) do skupiny, která na léčbu reaguje špatně (nebo vůbec).
  ○ Reakci na léčbu berte opět **na základě delta Δ hodnot** všech UPDRS hodnocení (rozdíly mezi hodnotami před a po podání léku).
  ○ Pro clusterování opět využijte k-means: 2 skupiny, náhodná inicializace, min. 5 replikací.
• S pomocí olabelovaných dat **natrénujte SVM model s RBF kernelem** – pro trénování použijte hodnocení UPDRS **<u>před podáním léku</u>** (aby model mohl klasifikovat pacienty pomocí klinického vyšetření předtím, než dostanou lék, a my bychom tak mohli vědět, zdali na něj budou reagovat dobře, či ne).
  ○ Optimální parametry SVM modelu (*C* a *σ*) zjistěte pomocí techniky *Grid Search* na cross-validovaných datech metodou *leave-one-out*, stejně jako na minulém cvičení.
  ○ Hyperparametry RBF kernelu *BoxConstraint* a *KernelScale* procházejte opět na mřížce mezi hodnotami 0.1 až 3.01 s krokem 0.1.
  ○ Pro vizualizaci a nalezení optimálních hodnot hyperparametrů z mřížky můžete opět použít přiloženou funkci `plotGridSearch`.
Pro optimálně nastavený SVM s RBF kernelem vypočítejte hodnoty senzitivity, specificity a accuracy pro cross-validovaný model metodou *leave-out-out*. Výsledky také můžete vykreslit pomocí `confusionchart`. Tyto hodnoty zapište a stručně slovně zhodnoťte výsledek klasifikačního experimentu.
	1.5

## Nepovinný bonus: **BUĎ: (za 0.5 bodu)** Implementujte manuálně algoritmus K-means, který bude fungovat ve více dimenzích. Algoritmus otestujte na datasetu bonus.csv (k dispozici ke stažení na Moodlu). **NEBO: (za 1 bod)** Implementujte manuálně algoritmus EM pro vícedimenzionální data, obsahující směs dvou normálních rozdělení – bonus.csv. Pokud chcete, můžete z dat použít pouze libovolné dvě dimenze a výsledek si pak vykreslit pomocí grafů `scatter` a `contour`.	0.5/1