**Čo sa študent naučí**

* načítať reálne dáta do pandas
* urobiť EDA (exploratory data analysis): head(), info(), describe()
* vyčistiť dáta (duplicity, chýbajúce hodnoty, typy)
* spraviť základné výpočty v **NumPy** (min–max normalizácia, percentuálna zmena)
* filtrovať, groupby a vytvoriť základné grafy v **matplotlib**
* natrénovať a porovnať **3 klasifikátory**: Logistic Regression, Random Forest, SVC
* vyhodnotiť model (accuracy, classification report, confusion matrix, CV)
* urobiť **malú predikciu** a uložiť výstupy

**Dataset a súbor**

* dátový súbor: **db1.csv** (polož k skriptu)
* skript: **analysis\_db1.py**
* výstupy: priečinok **outputs/** (vytvorí sa automaticky)

Ak má tvoj dataset iné názvy stĺpcov, upravíš to v časti rename\_map\_candidates.  
Cieľový stĺpec (label) môžeš nastaviť priamo (TARGET\_COL = "moj\_label"), alebo ho skript skúsí uhádnuť

**Celý postup (podľa skriptu)**

**1) Načítanie a EDA**

* pd.read\_csv("db1.csv") načíta dáta.
* head() ukáže prvých 5 riadkov (rýchla vizuálna kontrola).
* info() zobrazí typy stĺpcov a počty neprázdnych hodnôt.
* describe(include='all') dá základné štatistiky (pre číselné aj nečíselné stĺpce).

**Čo študentom zdôrazniť:**

* Pozerať sa na **typy** (object vs. float/int) a **chýbajúce hodnoty** hneď na začiatku.
* Pomenovať, čo je **riadok** (záznam) a čo je **stĺpec** (premenná / feature).

**2) Čistenie a príprava**

* **Premenovania**: rename\_map\_candidates (ilustratívne mapovanie, ktoré môžeš upraviť pre svoj dataset).
* **Duplicitné riadky**: drop\_duplicates() — skript vypíše, koľko ich odstránil.
* **Chýbajúce hodnoty**: číselné stĺpce sa doplnia **mediánom**.
* (Voliteľne) konverzia dátumu (pd.to\_datetime) podľa potrieb.

**Poznámky pre výučbu:**

* Prečo **medián**? Je robustnejší voči **extrémnym hodnotám** ako priemer.
* Ak je cieľový stĺpec textový (napr. mená tried), nechajte ho tak — ML časť si ho neskôr zakóduje.

**3) Základné štatistiky + NumPy transformácie**

* df[numeric\_cols].describe() — rýchly prehľad priemerov, smerodajnej odchýlky atď.
* **Min–max normalizácia** (NumPy funkcia minmax\_normalize) — vytvorí \*\_norm stĺpce.
* **Percentuálna zmena** (medzi susednými riadkami) v referenčnom stĺpci — vytvorí \*\_pct\_change.

**Didaktické body:**

* Prečo normalizovať? Niektoré modely (LR, SVC) sú citlivé na škálu.
* V skripte normalizácia slúži **len na EDA** (nie je použitá vo featurách pre model) — aby sme sa **vyhli úniku informácií**. Skutočné škálovanie pre ML je v **Pipeline** (nižšie).

**4) Filtrovanie, groupby a vizualizácie**

* Filtrovanie podľa priemeru v referenčnom stĺpci (ukážka praktického dotazu).
* groupby('species').mean() (ilustrácia agregácie — zmeň podľa tvojho datasetu).
* Grafy (každý samostatne, bez definovania farieb):
  + Histogram (distribúcia jednej premennej)
  + Boxplot (medián, kvartily, extrémy)
  + Scatter (vzťah dvoch číselných premenných)

**Čo vysvetliť pri grafoch:**

* Histogram: tvar distribúcie (symetrická, šikmá, multimodálna?).
* Boxplot: **rozptyl** a **outliery**.
* Scatter: možné **korelačné** vzťahy.

**5) Klasifikácia (ML)**

**a) Výber cieľa (target)**

* TARGET\_COL vieš nastaviť ručne, inak sa hľadá medzi ["target","label","class","species","y"].
* Ak nič z toho neexistuje, skript sa pozrie na **posledný stĺpec**: ak je textový alebo má málo unikátnych hodnôt, skúsi ho.

**b) Train/Test split**

* train\_test\_split(..., stratify=y) — **stratifikácia** zachová pomery tried v train/test.
* **RANDOM\_STATE** (42) — kvôli **reprodukovateľnosti**.

**c) Pipelines a škálovanie**

* Pre LR a SVC je v pipeline StandardScaler() — **škáluje len tréning** a aplikuje na test (správne pravidlo).
* RandomForest škálovanie nepotrebuje (stromy sú na škálu necitlivé).

**d) Modely, ktoré porovnávame**

* **LogisticRegression** — lineárny model, rýchly baseline.
* **RandomForest** — nelineárny, často silný na default nastaveniach, dá aj **feature importances**.
* **SVC (RBF)** — výkonný nelineárny klasifikátor, citlivý na škálu.

**e) Metriky a vysvetlenie**

* **Accuracy** — jednoduché, ale pozor na **nevyvážené dáta**.
* **Classification report** — Precision, Recall, F1 pre každú triedu.
* **Confusion matrix** — vizuálne, kde sa model mýli (riadky = skutočné, stĺpce = predikované).

**f) Cross-validation (CV)**

* cross\_val\_score(..., cv=5) — overí stabilitu modelu na viacerých rezoch dát.
* Využi na diskusiu o **variance** a **generalizácii**.

**g) Výber najlepšieho modelu**

* Porovnávame accuracy na test sete; vypíšeme názov modelu a skóre.
* Ak vyhrá **RandomForest**, zobrazíme **feature importances** (stĺpce, ktoré najviac pomohli).

**h) Malá predikcia**

* Zobrazíme **5 predikcií** z testu (skutočný vs. predikovaný).
* Uložíme celý test s predikciami do outputs/predictions\_test.csv.

**Dôležité metodické upozornenie:**

* Featury do modelu berieme z **pôvodných číselných stĺpcov**.  
  Stĺpce ako \*\_norm a \*\_pct\_change **nepoužívame** (zabránime úniku informácií a zmiešaniu EDA s tréningom).

**6) Uloženie výstupov**

* Očistené dáta: outputs/db1\_clean.csv
* Grafy: hist\_\*.png, boxplot.png, scatter.png, cm\_\*.png, prípadne feature\_importance\_rf.png
* Predikcie: outputs/predictions\_test.csv

**Ako prispôsobiť inému datasetu**

1. **Premenovania stĺpcov** – v rename\_map\_candidates si nadefinuj vlastné mapovanie.
2. **Cieľový stĺpec** – nastav TARGET\_COL = "moj\_label".
3. **Kategórie vo featurách** – skript teraz používa len **číselné** featury.
   * Jednoduchá voľba: pd.get\_dummies(df, drop\_first=True) pred ML časťou.
   * “Čistejšie” riešenie: ColumnTransformer + OneHotEncoder v pipeline (pokročilejšie).
4. **Nerovnováha tried** – pri silne nevyvážených dátach uprednostni **recall**, **F1**, alebo použij class\_weight='balanced' (LR/SVC/RF to podporujú).
5. **Tuning** – na ďalšiu hodinu: GridSearchCV alebo RandomizedSearchCV.

**Časté chyby a prevencia**

* **Únik informácií** (data leakage): škálovanie/transformácie robiť **len v pipeline** na train a aplikovať na test. (V skripte je to vyriešené.)
* **Bez stratifikácie**: pri nerovnováhe tried sa rozpadne pomer v train/test → skreslené výsledky.
* **Chýbajúce hodnoty**: neignorovať; mať jasnú stratégiu (medián/mean/imputer).
* **Nereprodukovateľnosť**: nastav random\_state.
* **Použitie accuracy pri extrémnej nerovnováhe**: radšej F1, ROC AUC (pri binárnych problémoch).

**Knižnice a ich funkcia:**

| **Knižnica** | **Popis** |
| --- | --- |
| **os** | práca so súbormi a priečinkami (napr. os.makedirs) |
| **sys** | ukončenie programu (sys.exit) |
| **numpy (np)** | rýchle numerické výpočty, práca s poľami (np.nanmin, np.empty\_like) |
| **pandas (pd)** | práca s tabuľkovými dátami (DataFrame), analýza a čistenie dát |
| **matplotlib.pyplot (plt)** | kreslenie grafov (histogramy, scatter, boxplot) |
| **sklearn.model\_selection** | rozdelenie dát (train\_test\_split), krížová validácia (cross\_val\_score) |
| **sklearn.preprocessing** | škálovanie (StandardScaler), kódovanie textu (LabelEncoder) |
| **sklearn.pipeline** | spájanie viacerých krokov do pipeline |
| **sklearn.metrics** | vyhodnocovanie modelov (accuracy\_score, classification\_report) |
| **sklearn.linear\_model**, **ensemble**, **svm** | rôzne typy klasifikátorov (LogisticRegression, RandomForestClassifier, SVC) |

**Premenné konfigurácie:**

| **Premenná** | **Význam** |
| --- | --- |
| DATA\_PATH | názov vstupného CSV súboru s dátami |
| OUTPUT\_DIR | názov priečinka, kam sa uložia výsledky (grafy, CSV) |
| TARGET\_COL | cieľový stĺpec (label), ktorý chceme predikovať |
| TARGET\_CANDIDATES | zoznam názvov, ktoré skript skúša automaticky nájsť ako cieľový stĺpec |
| RANDOM\_STATE | semienko náhody pre opakovateľné výsledky |

**Funkcie z Pandas:**

| **Funkcia** | **Popis** |
| --- | --- |
| pd.read\_csv("db1.csv") | načíta CSV súbor ako DataFrame |
| df.head() | ukáže prvých 5 riadkov tabuľky |
| df.info() | vypíše informácie o typoch a počte neprázdnych hodnôt |
| df.describe() | základné štatistiky číselných stĺpcov |
| df.drop\_duplicates() | odstráni duplicitné riadky |
| df.isnull().sum() | zistí, koľko chýbajúcich hodnôt je v jednotlivých stĺpcoch |
| df.fillna(median) | vyplní chýbajúce hodnoty mediánom |
| df.rename(columns={...}) | premenuje názvy stĺpcov |
| df.groupby('column').mean() | vypočíta priemery po skupinách |
| df.to\_csv("output.csv") | uloží DataFrame do CSV súboru |

**Funkcie z NumPy:**

| **Funkcia** | **Popis** |
| --- | --- |
| np.nanmin(x) / np.nanmax(x) | minimum a maximum poľa, ignoruje NaN |
| np.empty\_like(x) | vytvorí prázdne pole rovnakého tvaru |
| np.where(condition, A, B) | podmienka: ak True → A, inak → B |

**Vizualizačné funkcie:**

| **Funkcia** | **Popis** |
| --- | --- |
| df[column].hist() | histogram hodnôt v jednom stĺpci |
| df.boxplot(column=[...]) | boxplot – ukáže rozptyl a extrémy |
| plt.scatter(x, y) | scatter graf – vzťah dvoch premenných |
| plt.savefig("output.png") | uloží graf do obrázka |

**Funkcie zo sklearn – strojové učenie:**

| **Funkcia** | **Popis** |
| --- | --- |
| train\_test\_split(X, y, test\_size=0.25) | rozdelenie dát na trénovacie a testovacie |
| StandardScaler() | štandardizuje dáta (stred 0, smerodajná odchýlka 1) |
| LabelEncoder() | prevedie textové triedy na číselné |
| Pipeline([...]) | spojí viac krokov (škálovanie + model) do jedného |
| LogisticRegression() | lineárny klasifikátor |
| RandomForestClassifier() | stromový model, robustný a výkonný |
| SVC() | Support Vector Machine klasifikátor |
| accuracy\_score(y\_true, y\_pred) | presnosť predikcie |
| classification\_report() | precision, recall, F1-score a pod. |
| confusion\_matrix() | matica chýb – kde sa model mýli |
| cross\_val\_score(model, X, y, cv=5) | krížová validácia s 5 rozdeleniami dát |

**Vlastné pomocné funkcie v skripte:**

| **Funkcia** | **Popis** |
| --- | --- |
| minmax\_normalize(x) | prevedie hodnoty do rozsahu 0–1 |
| autodetect\_target(df, candidates) | pokúsi sa nájsť cieľový stĺpec automaticky podľa názvov |
| inv\_label(vals) | prevedie číselné predikcie späť na pôvodné textové triedy |

**Dôležité premenné počas analýzy:**

| **Premenná** | **Význam** |
| --- | --- |
| df | hlavný DataFrame – tabuľka s dátami |
| numeric\_cols | zoznam číselných stĺpcov v dátach |
| ref\_col | referenčný stĺpec – prvý číselný, používa sa napr. na percentuálnu zmenu |
| filtered | filtrovaný DataFrame podľa určitej podmienky |
| grouped | zoskupené dáta podľa kategórie |
| X, y | vstupy (features) a výstup (target) pre model |
| X\_train, X\_test, y\_train, y\_test | trénovacie a testovacie množiny |
| models | slovník troch modelov (LogReg, RandomForest, SVC) |
| results | zoznam výsledkov presnosti pre všetky modely |
| best\_model | model s najlepšou presnosťou |
| preview\_df | tabuľka so skutočnými a predikovanými triedami |

**Dôležité pojmy (vysvetlenie pre študentov):**

* **EDA (Exploratory Data Analysis)** – úvodná fáza analýzy, kde dáta skúmame, vizualizujeme a zisťujeme ich kvalitu.
* **Feature** – vstupná premenná pre model (napr. dĺžka okvetného lístka).
* **Label (Target)** – výstup, ktorý chceme predikovať (napr. druh kvetu).
* **Pipeline** – reťaz krokov, ktoré sa vykonajú v správnom poradí (napr. škálovanie → trénovanie modelu).
* **Stratifikácia** – zachovanie rovnakého pomeru tried v tréningovej aj testovacej množine.
* **Accuracy** – podiel správnych predikcií zo všetkých predikcií.
* **Confusion Matrix** – ukazuje, ktoré triedy si model pomýlil.