

Omega

Projekt o tréninku AI a jeho použití v praxi.

Projektová dokumentace: WristWatch AI

Informace o projektu

- **Název projektu:** WristWatch AI (Omega)
 - **Autor:** Jakub Hofman (C4c)
 - **Kontakt:** hofmjakub@gmail.com
 - **Datum dokončení:** 6. 4. 2025 (Neděle)
 - **Škola:** SPŠE Ječná (Česká republika, Praha)
 - **Typ projektu:** Školní
-

Popis

WristWatch AI je webová aplikace, která umožňuje využití modelů umělé inteligence pro analýzu a vyhodnocování náramkových hodinek. Projekt obsahuje dva modely:

1. **Model pro predikci ceny** – odhaduje **tržní hodnotu hodinek** na základě designu, technických specifikací a vlastností.
2. **Model pro klasifikaci použití** – určuje **pravděpodobné použití hodinek** (např. Sportovní, Vojenské, Společenské) podle konstrukce a designu.

Uživatelé mohou po přihlášení přes jednoduché webové rozhraní **vložit parametry svých vysněných hodinek** a ihned získat výsledek pomocí umělé inteligence.

Požadavky na uživatele

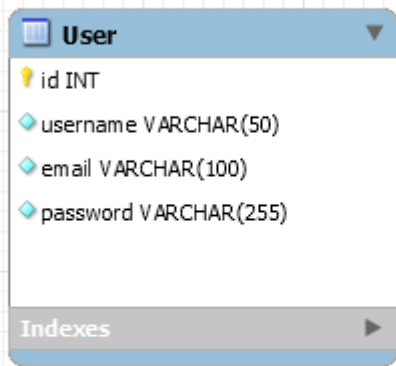
- **Python:** verze 3.10 nebo novější
- **Pip / Virtuální prostředí** (doporučeno)
- **MySQL Server:** verze 8.0 nebo novější
- **Webový prohlížeč:** Chrome, Firefox, Opera apod.
- **Operační systém:** Linux (doporučen Ubuntu), Windows

Potřebné knihovny:

- `pandas`
 - `scikit-learn`
 - `flask`
 - `pymysql`
 - `pickle` nebo `joblib`
 - `request`
 - `json`
-

Databáze

E-R Model:



Architektura a návrh

Projekt WristWatch AI je navržený, aby umožnil snadnou údržbu a rozšíření.

- **Frontend (HTML, CSS)**
 - uživatelské rozhraní pro zadávání dat
- **Backend (Flask)**
 - vytvořeno v Pythonu
 - zpracovává požadavky, transformuje data, volá modely a vrací výsledky
- **Machine Learning:**
 - Předzpracování dat - podpora pro single i multi-labeled hodnot
 - Kódování kategorií - `OneHotEncoder`, `MultiLabelBinarizer`
 - Modely:
 - `RandomForestRegressor` pro cenu
 - `RandomForestClassifier` pro použití
- **Databáze: MySQL**
 - pro přihlášení uživatelů a správu účtů
- Serializace **modelů a encoderů** pomocí `pickle`

Návrhy/Vzory:

- **ORM**
 - Oddělení crawlingu s daty, trénování a API
-

Jak to funguje?

1. Uživatel přejde na webovou stránku **WristWatch AI**
 - URL: <https://138.3.255.133>
 2. Vytvoří si účet či se rovnou přihlásí
 3. Na hlavní stránce **vyplní formulář s parametry hodinek** (značka, pohon, tvar, materiály atd.) a klikne na **Předvídat**
 4. Následně:
 - Data jsou převedena pomocí encoderů
 - Jsou předána do odpovídajícího modelu:
 - **Cena** => regresní model
 - **Použití** => klasifikační model
 - Výsledek se vrátí a okamžitě se zobrazí ve webovém rozhraní
-

Trénink AI

Při vývoji modelu pro odhad ceny a určení způsobu použití hodinek jsem experimentoval s několika typy modelů strojového učení. Cílem bylo najít modely, které dosahují co nejlepších výsledků se zachováním použitelnosti a robustnosti vůči různorodým vstupním datům.

1. Testování různých modelů

Nejprve jsem zpracoval data a následně otestoval různé modely:

- **Linear Regression / Logistic Regression**
- **Decision Tree**

- **Gradient Boosting** (např. XGBoost)

Volba Random Forest

Nakonec jsem zvolil **Random Forest**, protože:

- poskytuje **stabilní výsledky** i na menších nebo nevyvážených datech
- efektivně zachytává **nelineární vztahy** mezi atributy
- je snadno exportovatelný a připravený k nasazení v produkčním prostředí

Model pro predikci ceny (regrese)

Pro regresní úlohu jsem použil `RandomForestRegressor`. Model dostává jako vstupy:

- **numerické atributy**: vodotěsnost, rozměry, hmotnost,
- **vícehodnotové funkce hodinek**, převedené pomocí `MultiLabelBinarizer`
- **kategoriální atributy** (značka, určení, materiály...), zakódované přes `OneHotEncoder`

Evaluace proběhla pomocí:

- MAE (Mean Absolute Error)
- MSE (Mean Squared Error)
- R^2 Score

Model dosáhl velmi uspokojivých výsledků při testování a byl následně exportován (`rfr_wristwatch_price_model.pkl`).

Model pro predikci vhodného použití (klasifikace)

Pro klasifikační úlohu jsem zvolil `RandomForestClassifier`, který na podobném vektorovém vstupu predikuje:

- jeden nebo více štítků „**Použití**“ (např. sportovní, společenské, potápěčské...).

Výstup je **multi-labeled klasifikace**, vyhodnocoval jsem přes:

- **F1 skóre** (mikro průměr)
- **přesnost (Accuracy)**
- **klasifikační report**

Trénovaný model (`rfc_wristwatch_usage_model.pkl`) také zahrnuje všechny použité kodéry, aby bylo možné snadno zpracovávat nové vstupy.

Jak aplikaci sám nastavím?

Na GitHubu najdete podrobný postup v `README.txt` souboru.

- GitHub: <https://github.com/Mithynite/Omega>
-

Third-Party Libraries

- **scikit-learn** - pro trénink a použití ML modelů
 - **pandas** - pro zpracování a transformaci dat
 - **Flask** - pro webový backend
 - **mysql** - pro komunikaci s MySQL databází
 - **pickle** - pro ukládání modelů a encoderů
 - **json, csv** - pro práci se soubory v CSV a JSON
 - **request** - pro crawling stránek
-

Závěr

Projekt **WristWatch AI** spojuje svět chytrých algoritmů a klasických hodinek. Nabízí intuitivní platformu pro predikci ceny a rozpoznání účelu hodinek na základě jejich parametrů. Díky jednoduché architektuře a přehlednému API je tento systém snadno rozšiřitelný, testovatelný a vhodný i pro reálné použití nebo jako výuková pomůcka.

Sources

- <https://pandas.pydata.org/docs/reference/api/pandas.concat.html>
- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>
- <https://www.kaggle.com/code/swatisinghalmav/best-of-8-regression-models-to-predict-strength>
- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- <https://www.kaggle.com/code/junkal/selecting-the-best-regression-model>
- <https://www.youtube.com/watch?v=QhD015WUMxE>
- https://scikit-learn.org/stable/modules/naive_bayes.html
- <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier>
- <https://www.youtube.com/watch?v=fCUkvL0mbxI>
- https://www.youtube.com/watch?v=Z1RJmh_QgeA