# IZVEŠTAJ O REALIZOVANIM IMPLEMENTACIJAMA I EVALUACIJI MODELA

## 1. Uvod i opšti opis projekta

Tokom realizacije projekta **„Pronalaženje skrivenog znanja“**, sproveden je kompletan proces prikupljanja, obrade i analize podataka o turističkim aranžmanima, sa fokusom na automatsko preuzimanje javno dostupnih ponuda sa sajta **Kontiki.rs** i izgradnju sopstvene baze podataka pogodne za dalju analitiku i mašinsko učenje.

**Cilj sistema**: 1. Automatski preuzima i parsira turističke ponude (destinacija, hotel, broj zvezdica, tip usluge, broj noćenja, datum, cena, valuta, jedinica naplate). 2. Skladišti podatke u **MySQL** relacionu bazu, u normalizovanom formatu (tabela arrangements). 3. Omogućava **vizuelnu i analitičku obradu**, kao i implementaciju algoritama **linearne regresije** i **KNN klasifikacije** sa evaluacijom.

Implementacije su rađene u **Python 3.13** uz **Playwright (async)**, **SQLAlchemy**, **Matplotlib** i **Flask**.

## 2. Zadatak 1 i 2 — Prikupljanje i obrada podataka

### 2.1. Scraper i ključni izazovi

Razvijen je asinhroni Playwright scraper, sa iterativnim unapređenjima: - **Dinamički sadržaj u iframe-u** (iframe#cruisepool\_iframe): dodato eksplicitno prebacivanje konteksta i višestruko čekanje na alternativne selektore (.property-list, .property-name, .no-rooms-text, #SearchResult). - **Timeout greške** (npr. Frame.wait\_for\_selector): uvedena strategija “čekaj-na-više-selektora” i fallback grananja (dijagnostika *no rooms* vs. *has table*), plus exponential backoff i retry. - **Nedostajuće/nekonzistentne cene**: regex parsiranje price\_text, normalizacija valute (EUR default), razlikovanje **po osobi** vs **po sobi**. - **Paginacija i swiper datumi**: stabilizovano kruženje kroz datume i strane (swiper + next-page), uz limiter noćenja i datumske opsege.

**Struktura JSON izlaza** (primer):

{  
 "rogla": {  
 "by\_date": {  
 "2025-12-10": {  
 "7": [  
 { "name": "Hotel Rogla", "city": "Rogla", "stars": 3, "price\_num": 425.0, "currency": "EUR", "unit": "po osobi" }  
 ]  
 }  
 }  
 }  
}

### 2.2. Parsiranje i ubacivanje u bazu

ETL skript **load\_winter\_json.py** obavlja: - Rekurzivno razlaganje hijerarhije (različiti *legacy* i *winter* formati po destinacijama). - Preskakanje ponuda bez cene i nesigurnih metapodataka. - **Nasumičan izbor 1000 neduplikata** po lokaciji (ograničenje volumena podataka u jednom prolazu). - **Skaliranje cena na dve osobe** kada je cena izražena “po sobi”; sve vrednosti izražene u **EUR**. - Generisanje **identifikacionog ključa** za deduplikaciju i SQL UPSERT u arrangements.

**Deduplikacija u bazi**: korišćenje ROW\_NUMBER() po skupu kolona (site, url, naziv, lokacija, zvezdice, datum, noćenja, soba, usluga, AI-flag, stare i nove cene), brisanje redova sa rn > 1 (ostavlja se najnoviji zapis).

**Stanje podataka nakon čišćenja**: ~**7.5–8k** jedinstvenih aranžmana (varira po tranche-u), usklađene valute i jedinice, uniformisani datumi i noćenja.

## 3. Zadatak 3 — Vizuelizacija podataka

Iz pročišćene baze generisani su sledeći prikazi (Matplotlib, eksport u PNG za izveštaj): - **Top-10 destinacija** po broju aranžmana (bar chart). - **Broj aranžmana po mestu** (bar chart; kompletna distribucija). - **Distribucija hotela po zvezdicama** (pie chart; 2–5\*; procentualni odnos). - **Cenovni opsezi**: ≤500, 501–1500, 1501–3000, ≥3000 EUR (stacked bar/pie). - **Usluge (board)**: room only, self-catering, B&B, half board, full board, all inclusive (bar/pie + procenat).

## 4. Zadatak 4 — Linearna regresija (predviđanje cene)

### 4.1. Implementacija

Ručna implementacija **višestruke linearne regresije** sa **gradijentnim spustom**: - Ručna **One-Hot** kodifikacija za kategorije (city, board, room\_type; opcioni hotel\_name). - **Sezona** predstavljena ciklično: month\_sin, month\_cos. - **Log-transformacija cilja** log(price\_eur) radi stabilizacije varijanse. - **Flask UI**: forma za unos (grad, hotel, zvezdice, usluga, noćenja, mesec) i prikaz predikcije.

### 4.2. Problemi i rešenja

* **Preučenost na diskretnim kategorijama** → uvedene one-hot + regularizacija ulaza (standardizacija).
* **Mala korist od country** → atribut isključen iz obuke i UI.
* **Razlike u jedinici naplate** → svi zapisi prevedeni na **cenu za dve osobe**.
* **Ekstremi** (>3000 EUR) povećavaju RMSE → log-target + robustnija stopa učenja.

## 5. Zadatak 5 — K-Najbližih suseda (klasifikacija cene)

### 5.1. Implementacija

Ručna implementacija **KNN**: 1. **Min-Max** skaliranje ulaznih atributa. 2. **Euklidska distanca**. 3. **Majority voting** nad K najbližih.

**Klase cilja** (u skladu sa 3.d):

| Oznaka | Opis |
| --- | --- |
| 1 | ≤ 500 EUR |
| 2 | 501–1500 EUR |
| 3 | 1501–3000 EUR |
| 4 | ≥ 3000 EUR |

### 5.2. Eksperimenti

* Testiran **K ∈ {3, 5, 7, 9,11}**; najbolji za **K = 11**.
* Uvođenje hotel\_name donelo je **malo poboljšanje** ali nije ključno.
* Analiziran balans klasa i uticaj sezonskih feature-a.

### 5.3. Rezultati (test skup)

| Metrika | Vrednost |
| --- | --- |
| **Accuracy** | **0.880** |
| **Macro-F1** | **0.811** |
| **Train/Test** | **5862/ 1466** |
| **#Features** | **150** |

|  |  |
| --- | --- |
|  |  |

**Konfuziona matrica (red = istina, kolona = predikcija)**:

|  | C1 | C2 | C3 | C4 |
| --- | --- | --- | --- | --- |
| **C1 (≤500)** | 124 | 13 | 1 | 0 |
| **C2 (501–1500)** | 32 | 84 | 17 | 0 |
| **C3 (1501–3000)** | 1 | 17 | 226 | 51 |
| **C4 (≥3000)** | 0 | 0 | 44 | 856 |

**Napomena**: najviše zabuna je između **C2** i **C3** (granični aranžmani), što je očekivano zbog preklapanja u cenama.

## 6. Zaključak

Realizovan je kompletan **ETL + ML** tok: scraping → parsiranje → čišćenje/normalizacija → skladištenje → vizuelizacije → modeli → evaluacija. Ključni praktični izazovi (asinh­rono učitavanje u iframe-u, neuniformni JSON formati, skaliranje cena po jedinici naplate, deduplikacija i normalizacija valuta) rešeni su ciljanim inženjerskim pristupima i iterativnim unapređenjima.

**Rezime**: - **Linearni model** pruža upotrebljive predikcije (R² ≈ 0.64) i jasnu interpretaciju. - **KNN klasifikator** dostiže **≈90% tačnosti** uz stabilan Macro-F1. - **Flask UI** omogućava interaktivnu upotrebu modela i demonstraciju rezultata.

**Smernice za dalji rad**: - Uvesti **regularizaciju** (L2) i **polinomske interakcije** za regresiju (ručna implementacija) radi hvatanja nelinearnosti. - Istražiti **učene distance** ili **ponderisani KNN** (npr. 1/d) za granične slučajeve između C2–C3. - Proširiti skup izvora (npr. rapsodytravel.rs) za veću generalizaciju i deblje repove distribucije cena.

## 7. Prilozi (predlog)

* PNG grafici: *top10\_mesta.png*, *zvezdice\_pie.png*, *cenovni\_opsezi.png*, *board\_podela.png*, *reg\_pred\_vs\_true.png*, *reg\_residuals.png*, *knn\_confusion\_heatmap.png*, *knn\_acc\_k.png*.
* SQL skripte: *create\_table.sql*, *dedupe.sql*, *analytics\_queries.sql*.
* ETL: *scrape\_kontiki\_winter.py*, *load\_winter\_json.py*.
* ML/Flask: *regresion.py*, *app\_knn\_db.py* (ili *app\_knn\_db*).