**OrangeSelection**

Cistulli Domenico

**Introduzione**

🔷 Il progetto **OrangeSelection** è stato ideato per analizzare e classificare la qualità delle arance basandosi su caratteristiche come dimensione, peso, dolcezza, acidità, ecc. La missione principale è sviluppare un sistema integrato che utilizzi tecniche di Machine Learning e Deep Learning, supportate da una base di conoscenza semantica, per classificare la qualità delle arance in modo accurato e scalabile.

**Struttura del Progetto**

**1. Dataset**

* **data/raw/**: Contiene il dataset grezzo.
  + Preprocessed\_Orange\_Data.csv
* **data/processed/**: Contiene i dataset preprocessati e arricchiti.
  + Preprocessed\_Orange\_Data.csv
  + Enriched\_Orange\_Data.csv

**2. Script**

* **src/dataprocessing/preprocess\_data.py**: Script per preprocessare il dataset. Include:
  + Rimozione di duplicati.
  + Imputazione di valori mancanti.
  + Salvataggio del dataset preprocessato.
* **src/kbs/knowledge\_base.pl**: Base di conoscenza semantica che include fatti e regole sulle arance, integrata con RDF per il Web Semantico.
* **src/kbs/knowledge\_base.py**: Script Python per caricare i dati arricchiti e gestire i modelli di classificazione basati su SVM e Random Forest.
* **src/ml/models.py**: Implementazione di modelli di Machine Learning per classificare la qualità delle arance:
  + SVM
  + Random Forest
* **src/neural\_network/neural\_network.py**: Implementazione di una rete neurale con PyTorch per la classificazione delle arance.

**3. Notebook**

* **notebook/eda\_analysis.ipynb**: Notebook Jupyter che fornisce un'analisi esplorativa dettagliata del dataset.

**4. Script di Automazione**

* **scripts/run\_pipeline.py**: Automatizza l'intero processo:
  + Preprocessamento.
  + Addestramento dei modelli ML.
  + Addestramento della rete neurale.
* **scripts/generate\_report.py**: Genera un report con le metriche di valutazione dei modelli.

**5. Report Generato**

* **results/report.txt**: Contiene i risultati dettagliati dei modelli addestrati.

**Metodologia**

**1. Preprocessamento dei Dati**

1. Rimozione dei valori duplicati.
2. Imputazione dei valori mancanti con strategie di fallback.
3. Normalizzazione delle variabili numeriche e encoding delle variabili categoriche.

**2. Base di Conoscenza**

1. Utilizzo di Prolog per rappresentare la qualità delle arance con regole logiche.
2. Collegamenti RDF per accedere a conoscenze esterne.

**3. Modelli ML**

1. SVM e Random Forest sono stati implementati per la classificazione, utilizzando tecniche come:
   * **SMOTE** per bilanciare le classi.
   * Grid Search per l'ottimizzazione degli iperparametri.

**4. Rete Neurale**

1. Una rete neurale con tre layer completamente connessi è stata sviluppata per migliorare la classificazione delle arance, raggiungendo il 72.5% di accuratezza.

**Risultati**

**Modelli di Machine Learning**

| **Modello** | **Accuratezza** | **Precision** |
| --- | --- | --- |
| **SVM** | 70% | 71% |
| **Random Forest** | 82.5% | 83% |

**Rete Neurale**

| **Modello** | **Accuratezza** |
| --- | --- |
| **Neural Network** | 72.5% |

**Come Eseguire il Progetto**

**Prerequisiti**

1. Installare le dipendenze da requirements.txt:
2. pip install -r requirements.txt

**Esecuzione**

1. Eseguire la pipeline completa:
2. python scripts/run\_pipeline.py
3. Generare il report:
4. python scripts/generate\_report.py

**BIBLIOGRAFIA**

**SVILUPPI FUTURI**