Pipeline di Data Quality e Machine Learning

Bishara Giovanni -
Singh Probjot - 869434

4 giugno 2025

Sommario

Questa relazione descrive una pipeline completa per la gestione della qualità dei dati e l'implementazione di modelli di machine learning. Il sistema combina fasi di preprocessing, trasformazione, verifica della qualità dei dati, addestramento di modelli predittivi (SVM e Decision Tree) e valutazione delle performance. L'intero processo è orchestrato utilizzando il framework Luigi, che garantisce l'esecuzione ordinata delle task e la gestione automatica delle dipendenze.

Indice

1 Introduzione

1.1 Obiettivi della Pipeline

Questa pipeline è stata progettata per:

- Verificare la qualità dei dati attraverso 4 dimensioni fondamentali
- Addestrare modelli di classificazione per prevedere il tipo di vino (rosso/bianco)
- Valutare oggettivamente le prestazioni dei modelli
- Fornire un framework riproducibile e scalabile

1.2 Architettura Generale

Il processo si articola in tre fasi principali:

1. **Preprocessing Dati**: Pulizia, trasformazione e riduzione dimensionalità
2. **Modellazione ML**: Addestramento di SVM e Decision Tree
3. **Verifica DQ**: Controlli di qualità su 4 dimensioni

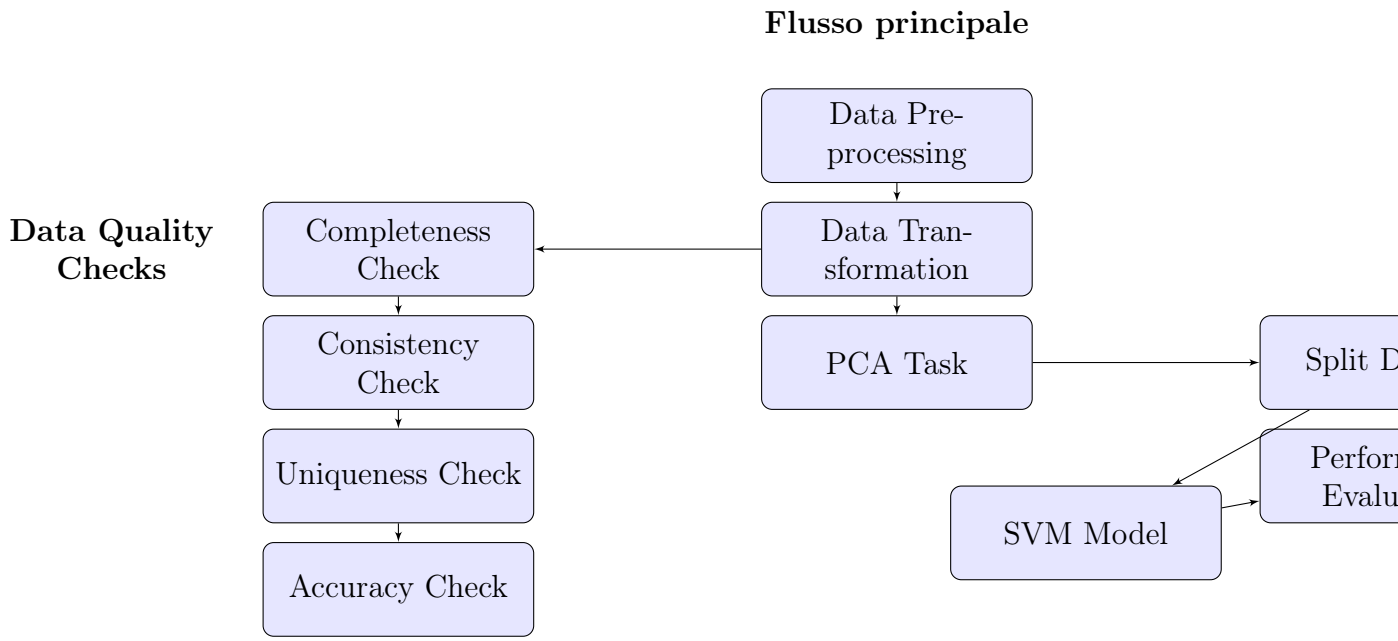

Figura 1: Architettura completa della pipeline

2 Teoria: Le 4 Dimensioni della Data Quality

Data Quality si riferisce alla capacità dei dati di soddisfare i requisiti del loro utilizzo previsto. È composta da quattro dimensioni fondamentali:

2.1 Completezza (Completeness)

Misura la presenza di valori mancanti nel dataset. Un dato è completo quando tutti i valori attesi sono presenti.

Soglia: $< 3\%$ dei record totali

$$\text{Completeness Score} = 1 - \frac{\text{Numero valori mancanti}}{\text{Totale valori attesi}}$$

2.2 Consistenza (Consistency)

Verifica la coerenza logica dei dati e l'assenza di valori anomali. Utilizza tre approcci:

1. **Controllo dominio**: Valori devono essere in range fisici/chimici accettabili
2. **Metodo statistico**: Identifica outlier con media $\pm\, 3\sigma$
3. **Metodo IQR**: $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$

2.3 Unicità (Uniqueness)

Garantisce che non ci siano duplicati e che tutte le feature siano informative:

- Record duplicati: Righe identiche
- Feature non informative: Colonne con un solo valore distinto

2.4 Accuratezza (Accuracy)

Verifica la correttezza dei tipi di dato e la conformità agli schemi attesi:

- `type`: Booleano (0=rosso, 1=bianco)
- Altre feature: Numeriche

3 Implementazione della Pipeline

3.1 Preprocessing Dati

3.1.1 DataPreprocessing

- **Input**: `winetype.csv`
- **Output**: Dataset pulito (`winetype_cleaned.csv`)
- **Operazioni**:

1. Rimozione record con valori mancanti
2. Eliminazione duplicati
3. Simulazione dati "sporchi" per test di qualità

3.1.2 DataTransformation

- **Input**: Dataset pulito
- **Output**: Dataset trasformato (`winetype_transformed.csv`)
- **Operazioni**:
  1. Encoding variabile target "type" (rosso=0, bianco=1)
  2. Rimozione feature "quality" (irrilevante per classificazione)

3.1.3 PCATask

- **Input**: Dataset trasformato
- **Output**: Dataset ridotto dimensionalmente (`winetype_pca.csv`)
- **Operazioni**:
  1. Standardizzazione delle feature
  2. Riduzione dimensionalità con PCA (11 feature $\rightarrow$ 5 componenti principali)

3.1.4 SplitDataset

- **Input**: Dataset PCA
- **Output**: Training set (80%) e Test set (20%)
- **Note**: Split stratificato per mantenere la distribuzione delle classi

3.2 Modellazione Machine Learning

3.2.1 SVMModel

- **Algoritmo**: Support Vector Machine
- **Kernel**: Lineare
- **Output**: Modello serializzato (`svm_model.pkl`)
- **Parametri**:
  - C=1.0
  - random_state=42

3.2.2 DTCModel

- **Algoritmo**: Decision Tree Classifier
- **Output**: Modello serializzato (`dtc_model.pkl`)
- **Parametri**:
  - max_depth=5
  - min_samples_split=10
  - random_state=42

3.2.3 PerformanceEval

- **Input**: Modelli addestrati e test set
- **Output**: Report metriche (`metrics.csv`)
- **Metriche calcolate**:
  - Accuratezza (Accuracy)
  - Precisione (Precision)
  - Recall (Sensibilità)
  - F1-score
- **Intervalli confidenza**: Calcolati con cross-validation stratificata a 10 fold

Formula intervallo confidenza:

$$\bar{x} \pm t\frac{s}{\sqrt{n}}$$

Dove:

- $\bar{x}$: Media delle metriche
- t: Valore t-distribuzione per $\alpha = 0.05$
- s: Deviazione standard
- n: Numero di campioni

3.3 Verifica Data Quality

4 Esecuzione e Risultati

4.1 Comando di Esecuzione

La pipeline si avvia con:

```
python3 -m luigi --module pipeline FullPipeline --local-scheduler
```

Task	Soglia	Metodo
Completeness	<3% valori mancanti	Conteggio valori nulli
Consistency	<3% outlier	3 metodi: dominio, std, IQR
Uniqueness	<3% duplicati	Conteggio record duplicati
Accuracy	100% tipi corretti	Verifica tipi di dato

Tabella 1: Soglie e metodi per i controlli di qualità

4.2 Output Attesi

- `datasets/`: File CSV trasformati
- `models/`: Modelli serializzati (SVM e Decision Tree)
- `reports/metrics.csv`: Report prestazioni modelli

```
model_name,accuracy,precision,recall,f1_score
SVM,0.92,0.93,0.91,0.92
DecisionTree,0.88,0.87,0.89,0.88

```

- `pipeline.log`: Log dettagliato dell'esecuzione

4.3 Interpretazione Risultati

- **Accuratezza $>$ 90%**: Modello eccellente
- **Accuratezza 80-90%**: Modello buono
- **Accuratezza $<$ 80%**: Necessità di miglioramento
- **F1-score**: Media armonica tra precisione e recall (ottimo indicatore per dataset sbilanciati)

5 Conclusioni

Questa pipeline rappresenta un framework completo per:

- Garantire dati affidabili attraverso controlli di qualità automatizzati
- Sviluppare modelli ML robusti per la classificazione
- Valutare oggettivamente le prestazioni
- Documentare l'intero processo tramite logging strutturato

L'approccio modulare e configurabile lo rende adattabile a diversi contesti e tipi di dataset, non solo al dominio enologico. L'utilizzo di Luigi garantisce l'esecuzione ordinata delle task e la gestione automatica delle dipendenze.

Riferimenti bibliografici

[1] Spotify. (2012). *Luigi: Python package for building complex pipelines*. GitHub repository.

[2] DAMA International. (2017). *DAMA-DMBOK: Data Management Body of Knowledge* (2nd ed.).

[3] Pedregosa, F. et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research.

A Codice Configurazione Luigi

```
[DataPreprocessing]
input_csv = datasets/winetype.csv
cleaned_csv = datasets/cleaned_data.csv

[DataTransformation]
transformed_csv = datasets/transformed_data.csv

[PCATask]
pca_csv = datasets/pca_data.csv

[SplitDataset]
train_csv = datasets/train_data.csv
test_csv = datasets/test_data.csv

[SVMModel]
svm_model_file = models/svm_model.pkl

[DTCModel]
dtc_model_file = models/dtc_model.pkl

[PerformanceEval]
metrics_csv = reports/metrics.csv
```

Listing 1: Esempio file luigi.cfg

B Schema Dataset Originale

Feature	**Tipo**	**Descrizione**
type	categorico	Rosso (0) o Bianco (1)
fixed acidity	numerico	Acidità fissa (g/dm^3)
volatile acidity	numerico	Acidità volatile (g/dm^3)
citric acid	numerico	Acido citrico (g/dm^3)
residual sugar	numerico	Zuccheri residui (g/dm^3)
chlorides	numerico	Cloruri (g/dm^3)
free sulfur dioxide	numerico	SO_2 libero (mg/dm^3)
total sulfur dioxide	numerico	SO_2 totale (mg/dm^3)
density	numerico	Densità (g/cm^3)
pH	numerico	pH
sulphates	numerico	Solfati (g/dm^3)
alcohol	numerico	Alcol (% vol)
quality	categorico	Qualità percepita (0-10)

Tabella 2: Schema del dataset originale del vino