UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

A.A. 2023/2024
Laurea Magistrale
Informatica

Progetto Data Quality

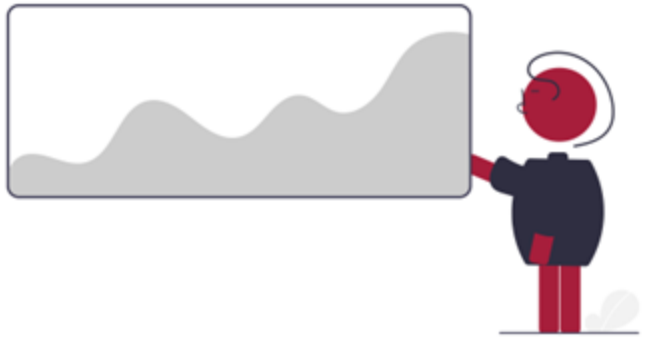
Architetture Dati – Giugno 2024

Realizzato da:

Cavaleri Matteo - 875050
Gargiulo Elio - 869184
Piacente Cristian - 866020

Introduzione del Progetto

- Machine Learning: Analisi, Modelli e Performance
- Dimensioni di Qualità
- Data Explainability
- Analisi Esperimenti di Data Quality
- Considerazioni e Conclusioni

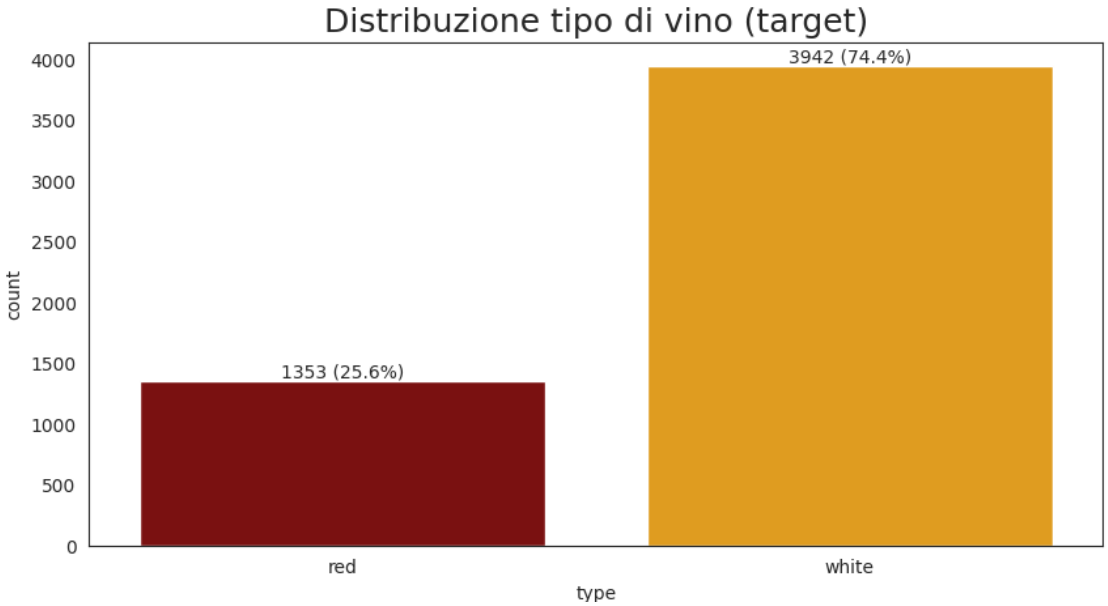


Features Dataset e Target

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	white	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	white	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	white	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	white	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	white	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

- Target: type (binario)
- 11 features continue e 1 categorica (quality intero tra 0 e 10) che è stata droppata + pulizia iniziale

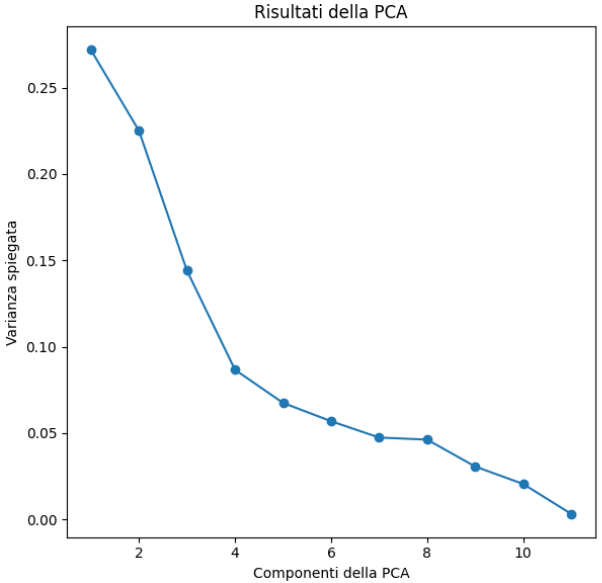
Analisi del Target

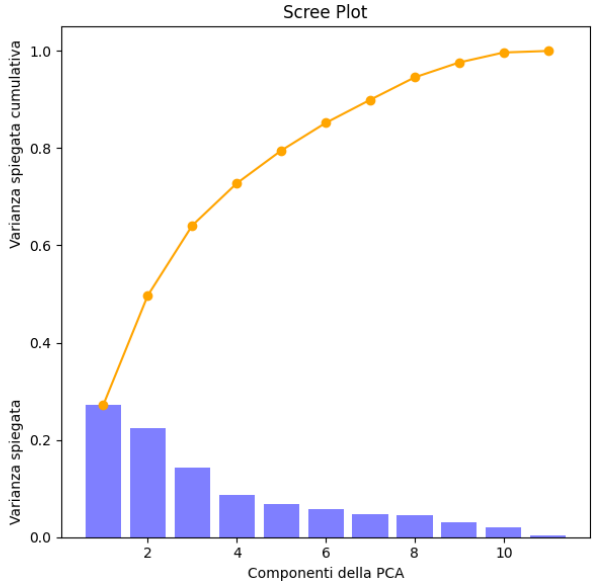


- Target abbastanza sbilanciato

Principal Component Analysis

	Eigenvalue	Variance Percent	Cumulative Variance Percent
Comp 1	2.991077	27.186472	27.186472
Comp 2	2.476404	22.508515	49.694986
Comp 3	1.585096	14.407246	64.102232
Comp 4	0.953458	8.666165	72.768397
Comp 5	0.742378	6.747617	79.516014





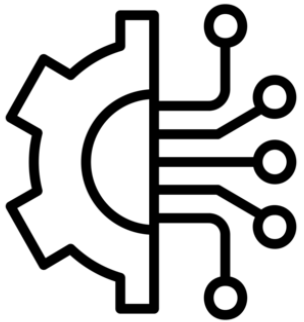
- 5 componenti spiegano ~ 80% varianza

Modelli di Apprendimento

- Rete Neurale
- Support Vector Machines
- Albero di Decisione

Approccio Naive:
Immediato → Iperparametri di default/comuni

Suddivisione del dataset in:
- 80% Training Set
- 20% Test Set



Performance Iniziali

Model Performance Metrics

Model Name	Accuracy	Precision	Recall	F1 Score
Neural Network	0.983	0.9923	0.9847	0.9885
SVM	0.9858	0.9886	0.9923	0.9905
Decision Tree	0.9783	0.981	0.9898	0.9854

Ottime Performance di Partenza su metriche globali ~ 0.98 per metrica

Stratified 10 Fold Cross Validation

Model Performance Intervals

Model Name	Accuracy Interval Lower	Accuracy Interval Upper	Precision Interval Lower	Precision Interval Upper	Recall Interval Lower	Recall Interval Upper	F1 Score Interval Lower	F1 Score Interval Upper
Neural Network	0.9811	0.9879	0.9855	0.9923	0.9865	0.9942	0.9873	0.9919
SVM	0.9836	0.9866	0.9772	0.984	0.9777	0.9829	0.9785	0.9823
Decision Tree	0.9751	0.9826	0.9677	0.9756	0.966	0.9799	0.9672	0.9773

Mostra generalmente migliori metriche con intervalli di confidenza del 95% ~ 0.97/0.99

Data Quality: Dimensioni

Garanzia che i dati siano **di qualità**

- **Completezza** → Valori Mancanti
- **Consistenza** → Inconsistenze, Outliers
- **Unicità** → Righe Duplicate, Distribuzione Valori
- **Accuratezza** → Tipi Trattati

Implementate e applicate sul Dataset Pulito **(Pre PCA)** al fine di avere una conferma sulla pulizia

In questo modo abbiamo un punto di partenza per i successivi esperimenti sulla qualità dei dati

Data Quality: Completezza

Output della Pipeline che mostrano gli esiti sul Dataset Pulito

```
type                  0
fixed acidity         0
volatile acidity      0
citric acid           0
residual sugar        0
chlorides             0
free sulfur dioxide   0
total sulfur dioxide  0
density               0
pH                    0
sulphates             0
alcohol               0
dtype: int64
```

Conteggio Valori Mancanti

```
type                  0.0
fixed acidity         0.0
volatile acidity      0.0
citric acid           0.0
residual sugar        0.0
chlorides             0.0
free sulfur dioxide   0.0
total sulfur dioxide  0.0
density               0.0
pH                    0.0
sulphates             0.0
alcohol               0.0
dtype: float64
```

Percentuale Valori Mancanti

Data Quality: Consistenza

```
feature_ranges = {
        'fixed acidity': (0, None),  # Check for non-negative values
        'volatile acidity': (0, None),  # Check for non-negative values
        'citric acid': (0, None),  # Check for non-negative values
        'residual sugar': (0, None),  # Check for non-negative values
        'chlorides': (0, None),  # Check for non-negative values
        'free sulfur dioxide': (0, None),  # Check for non-negative values
        'total sulfur dioxide': (0, None),  # Check for non-negative values
        'density': (0, None),  # Check for non-negative values
        'pH': (0, 14),  # Default Range for pH
        'sulphates': (0, None),  # Check for non-negative values
        'alcohol': (0, 100),  # % Alcohol goes necessarily from 0 to 100%
}
```

```
============================================================
CONSISTENCY - STD BOUNDS RESULTS:
============================================================
fixed acidity: (0.6145579802193915, 13.821457128373623)
volatile acidity: (0, 1.1852054165644146)
citric acid: (0, 1.0543419012093667)
residual sugar: (0, 27.554235702957847)
chlorides: (0, 0.2411949011056968)
free sulfur dioxide: (0, 119.18259086148798)
total sulfur dioxide: (0, 398.0541620677567)
density: (0.9796917405510556, 1.0093805049635807)
pH: (2.4236118482077655, 4.02515869003586)
sulphates: (0, 1.2824521302170306)
alcohol: (4.617486789639095, 16.482820418411897)
```

Range di Dominio da Rispettare:

- **Es. pH da 0 a 14**
- **Es. Valori non negativi**

Ranges per gli Outliers:

- Media ± 5 Deviazione Standard
- IQR con threshold = 4

L'Esempio raffiguato utilizza il primo approccio

Data Quality: Consistenza

```
============================================================
CONSISTENCY - INCONSISTENT VALUES DEFAULT RESULTS:
============================================================
fixed acidity: PASSED
volatile acidity: PASSED
citric acid: PASSED
residual sugar: PASSED
chlorides: PASSED
free sulfur dioxide: PASSED
total sulfur dioxide: PASSED
density: PASSED
pH: PASSED
sulphates: PASSED
alcohol: PASSED
```

- Ogni Valore **DEVE** essere nel range di dominio, ovvero con l'esito equivalente a **PASSED**

```
============================================================
CONSISTENCY - OUTLIERS STD INFO RESULTS:
============================================================
Feature: fixed acidity
Count of outliers: 8
Rows with outliers: [1273, 4156, 4270, 4327, 4415, 4425, 4427, 4507]

Feature: volatile acidity
Count of outliers: 4
Rows with outliers: [4056, 4057, 4524, 5048]

Feature: citric acid
Count of outliers: 2
Rows with outliers: [615, 2568]

Feature: residual sugar
Count of outliers: 2
Rows with outliers: [1378, 2278]
```

- **Tolleranza del 3%** sul totale dei records del Dataset per gli Outliers identificati dai Ranges

Data Quality: Unicita'

Output della Pipeline che mostrano gli esiti di Unicità

```
type                    2
fixed acidity         106
volatile acidity      187
citric acid            89
residual sugar        315
chlorides             214
free sulfur dioxide   135
total sulfur dioxide  276
density               996
pH                    108
sulphates             111
alcohol               111
dtype: int64
```

Conteggio Valori Unici

```
Duplicated Records: 0
```

Nessun Duplicato

```
type                   0.037771
fixed acidity          2.001889
volatile acidity       3.531634
citric acid            1.680831
residual sugar         5.949008
chlorides              4.041549
free sulfur dioxide    2.549575
total sulfur dioxide   5.212465
density               18.810198
pH                     2.039660
sulphates              2.096317
alcohol                2.096317
dtype: float64
```

Percentuale Valori Unici

Data Quality: Accuratezza

Output della Pipeline che mostrano gli esiti sul Tipo Corretto

```
# Feature: type that has to be respected
expected_types = {
        'fixed acidity': 'float64',
        'volatile acidity': 'float64',
        'citric acid': 'float64',
        'residual sugar': 'float64',
        'chlorides': 'float64',
        'free sulfur dioxide': 'float64',
        'total sulfur dioxide': 'float64',
        'density': 'float64',
        'pH': 'float64',
        'sulphates': 'float64',
        'alcohol': 'float64',
        'type': 'bool'  # Target is true or false
        }
```

Tipi Attesi:

- **float64** per variabili continue
- **bool** per il target

```
fixed acidity: PASSED
volatile acidity: PASSED
citric acid: PASSED
residual sugar: PASSED
chlorides: PASSED
free sulfur dioxide: PASSED
total sulfur dioxide: PASSED
density: PASSED
pH: PASSED
sulphates: PASSED
alcohol: PASSED
type: PASSED
```

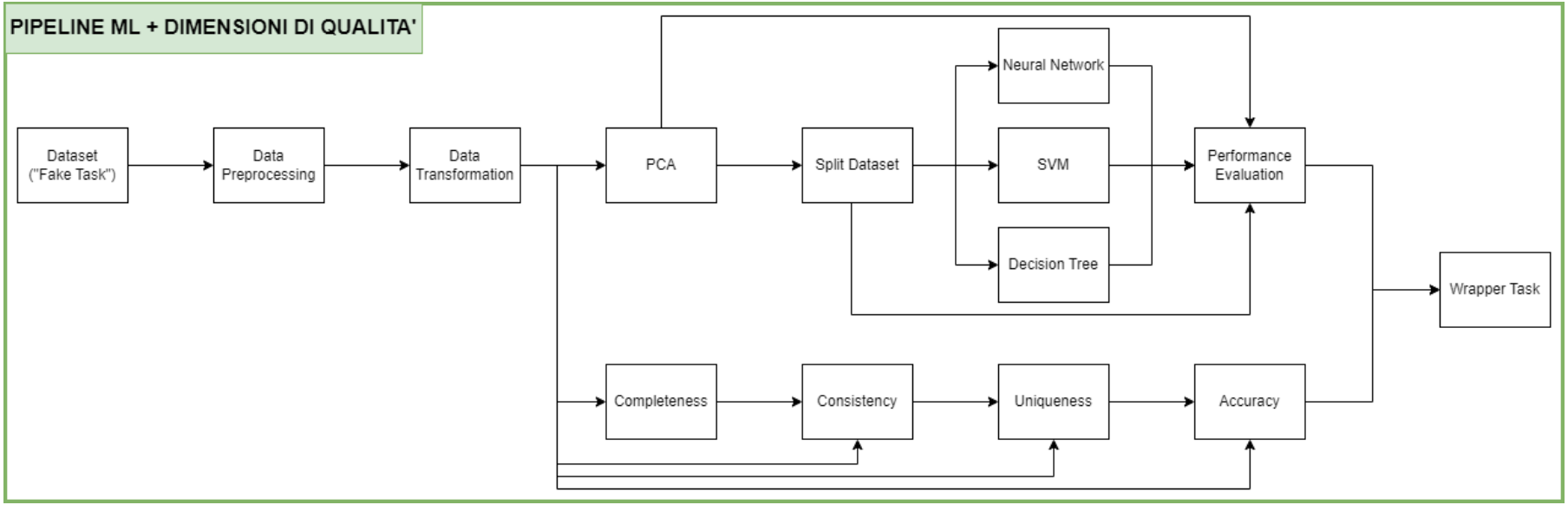
PASSED indica se il tipo è correttamente utilizzato

Struttura del progetto

- Necessari **scalabilità** e **automatizzazione** per far fronte ~~colab~~ ad un numero elevato di esperimenti

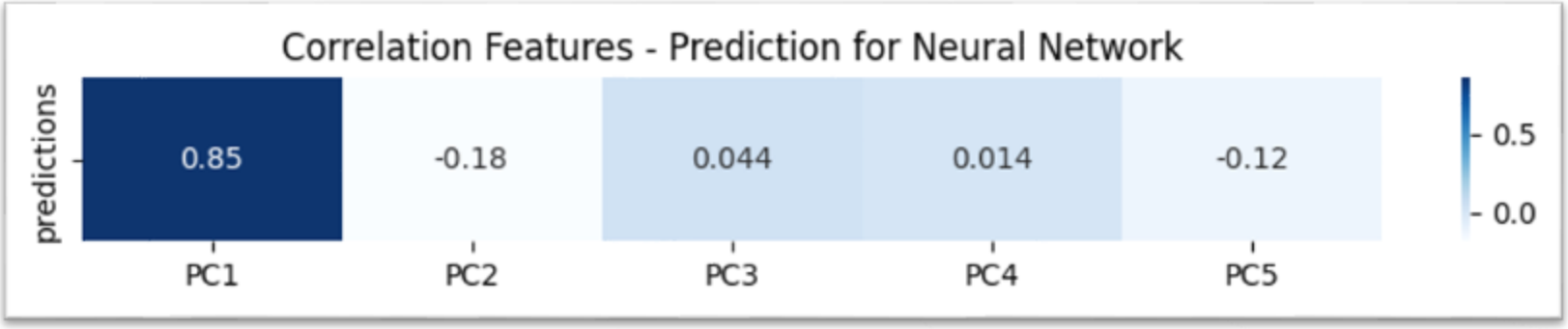
- Il progetto è organizzato come segue:
  - Pipeline Python (libreria Luigi)
    - **ML** e check **Dimensioni di Qualità**
    - **Esperimenti**
  - Jupyter notebooks
    - **ML** ed **Explainability**
    - **Esperimenti**

Pipeline Machine Learning



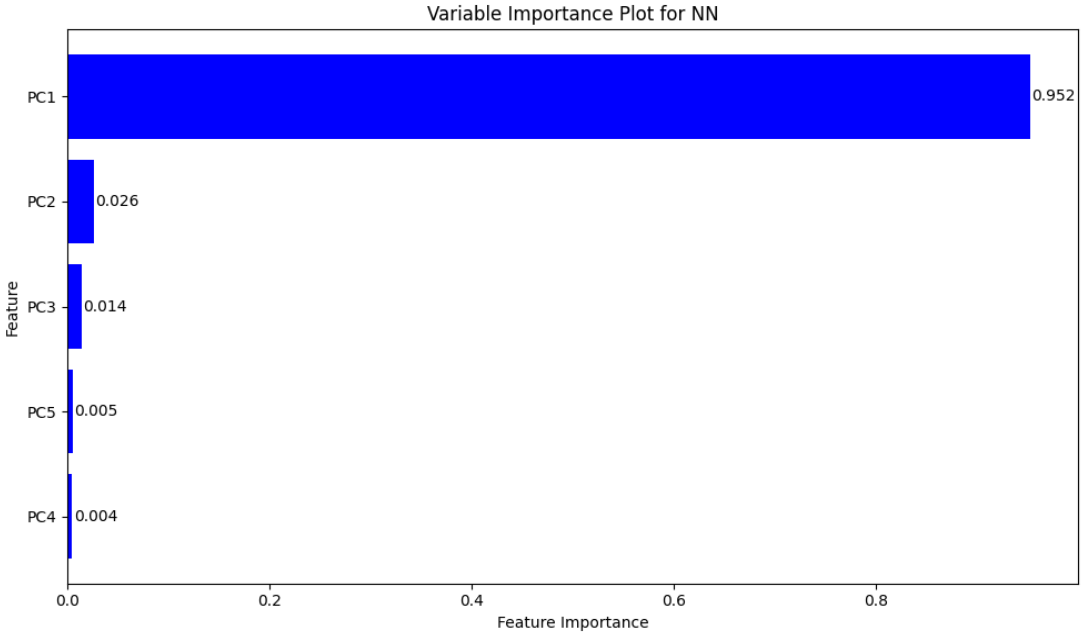
"Completeness", "Consistency", "Uniqueness" e "Accuracy" non producono file.

Data Explainability (1)



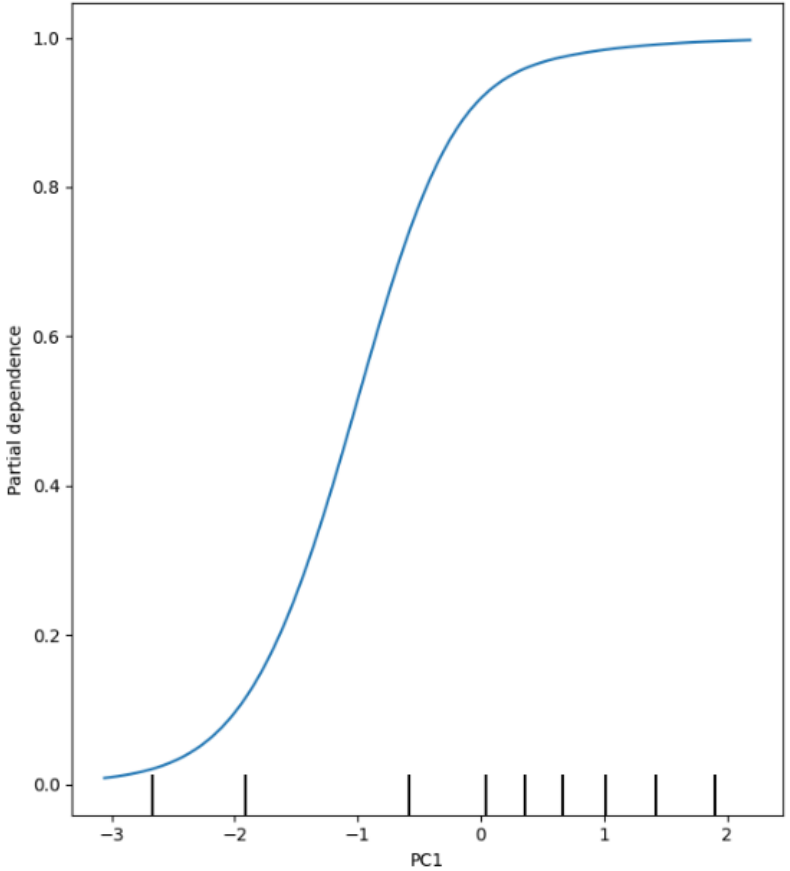
- Correlazione delle predizioni con ogni componente della PCA per la rete neurale.

Data Explainability (2)



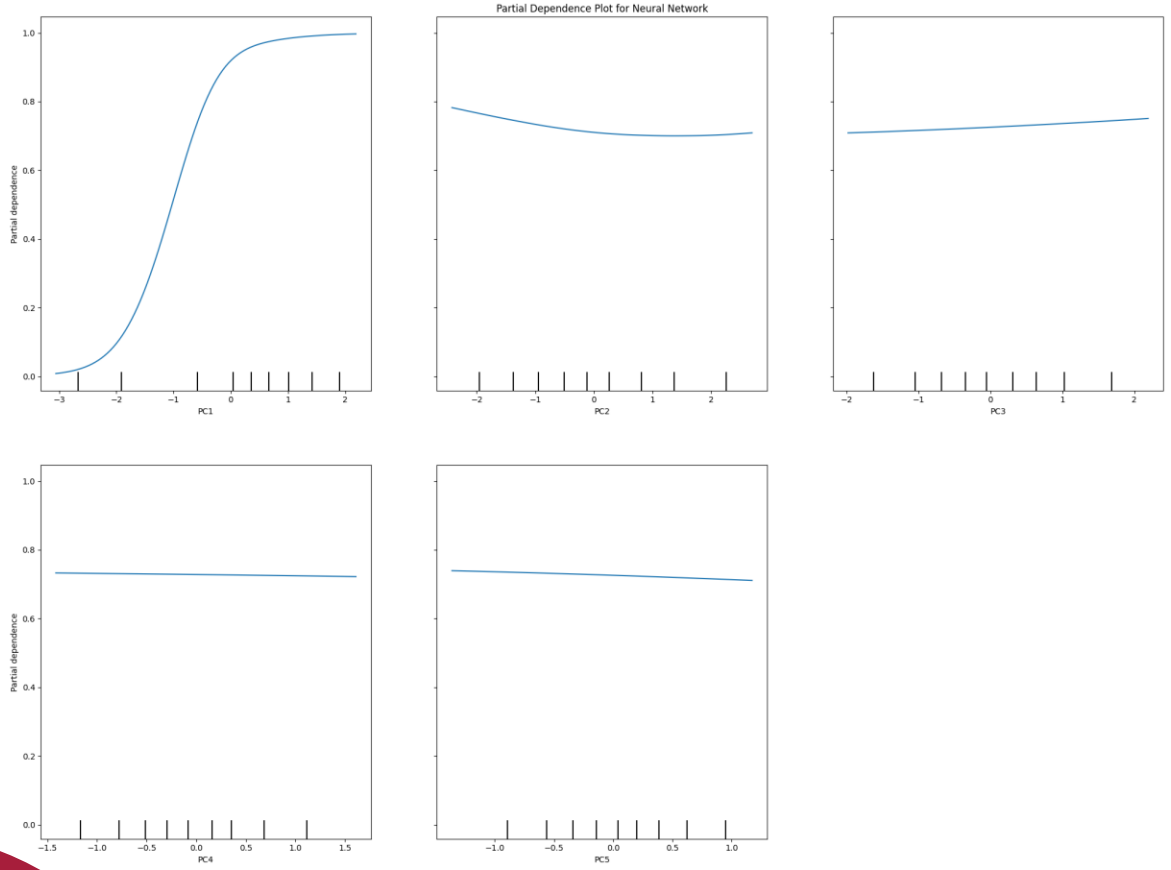
- Il grafico di Variable Importance è coerente con le correlazioni

Data Explainability (3)

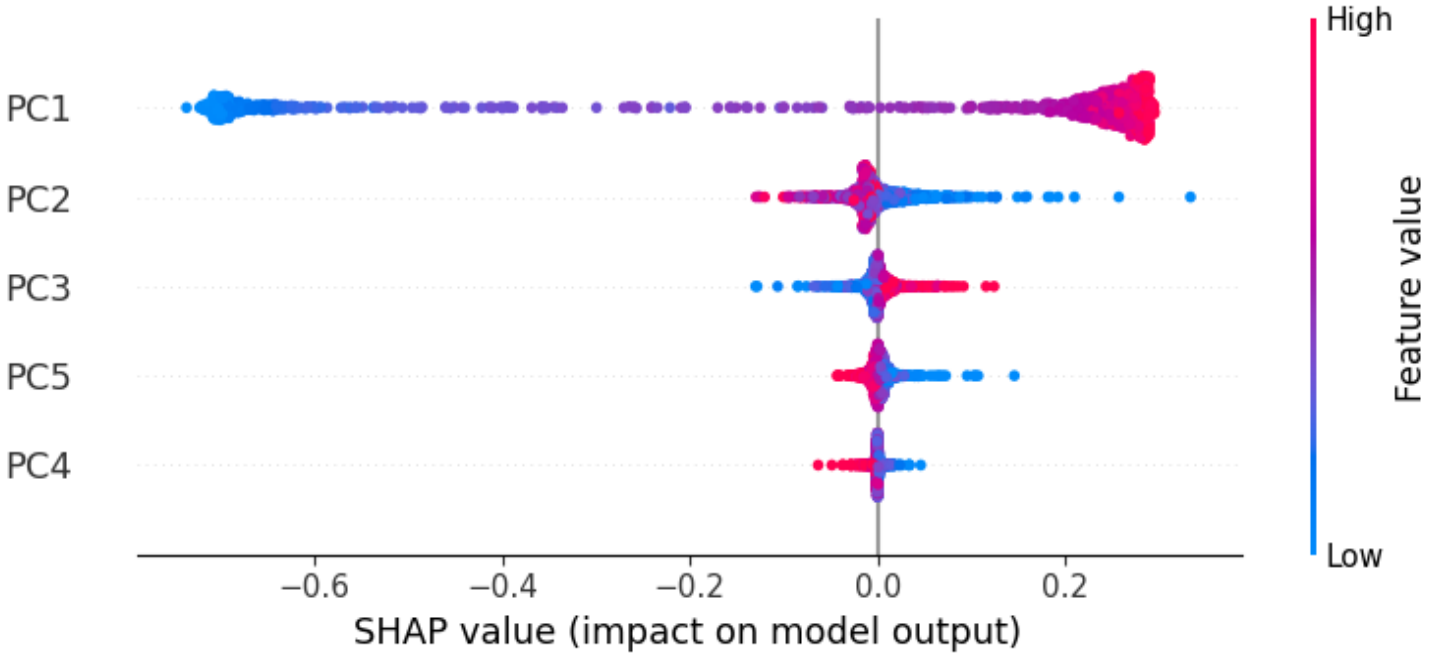


- -1 risulta essere il valore di frontiera per determinare la classificazione di un tipo di vino.

Data Explainability (4)

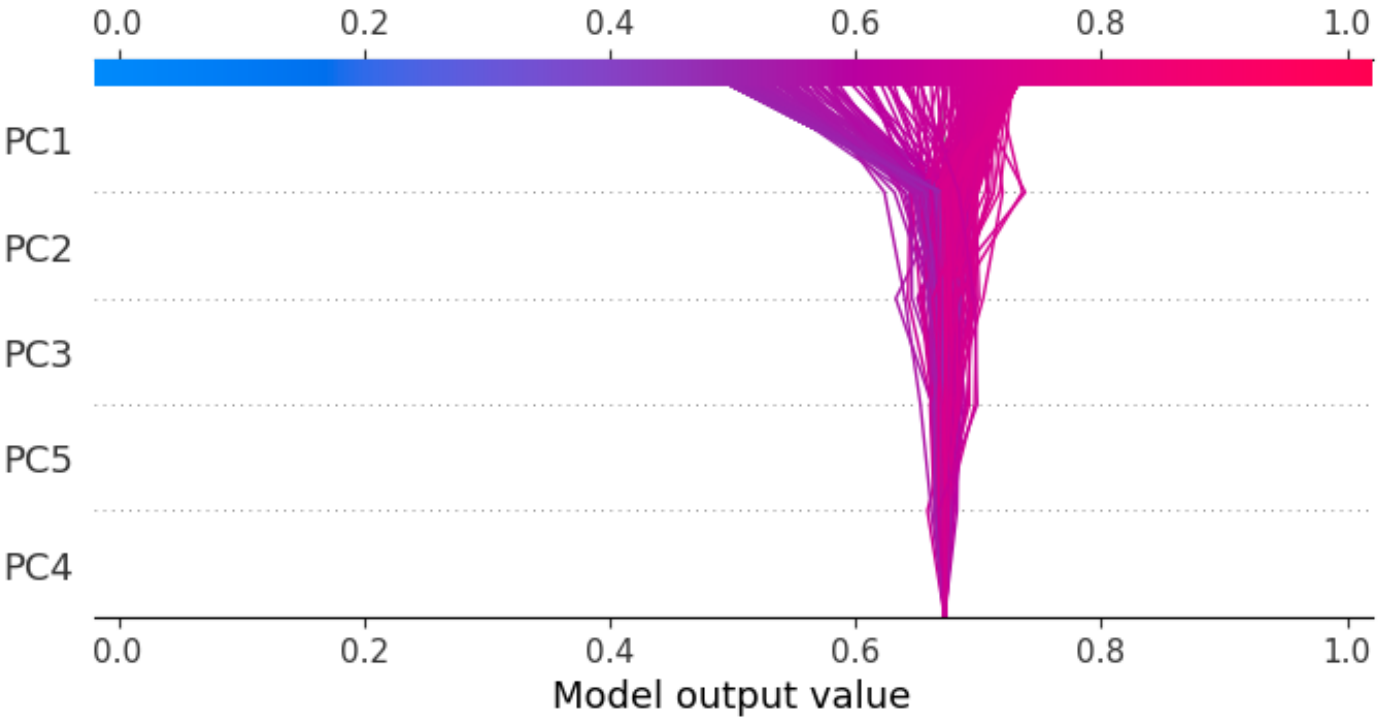


Data Explainability (5)

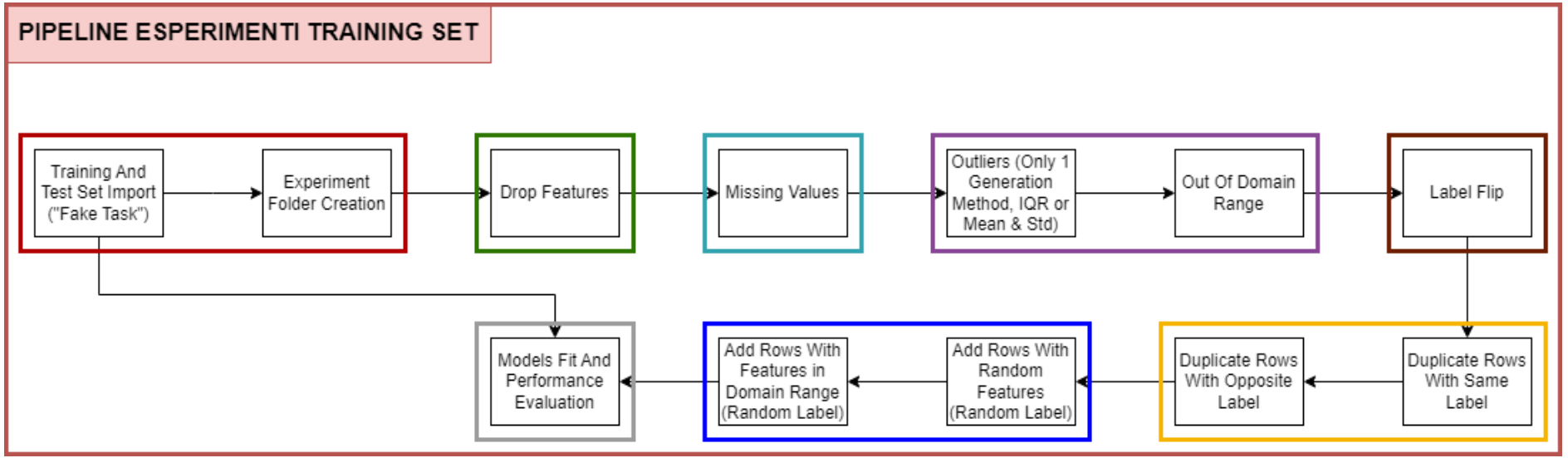


- Viene mostrato come ogni valore della PCA influenza lo SHAP value.

Data Explainability (6)



Pipeline Esperimenti



Colori: suddivisione concettuale.
Struttura interamente sequenziale con **output intermedi**.

Gli Esperimenti

659 Esperimenti effettuati sul Dataset, sporcando con:

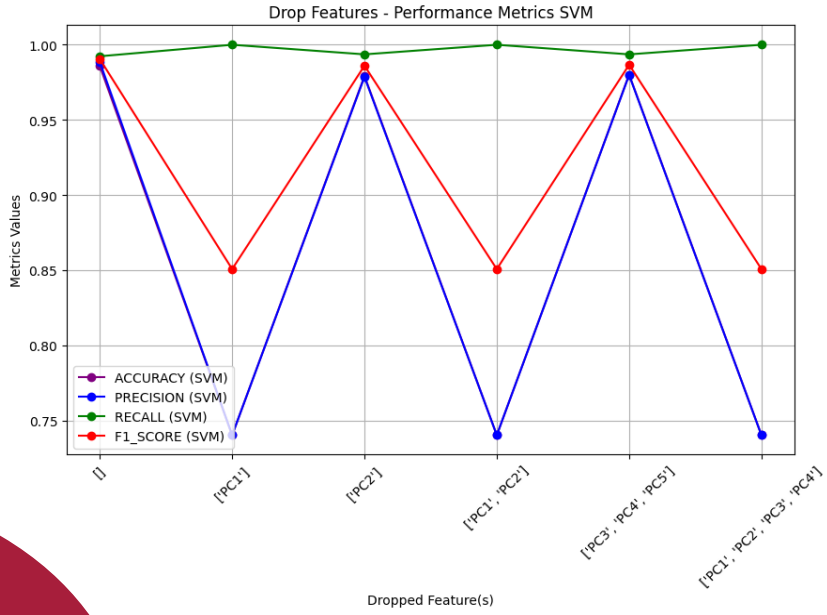
- **Drop Features**
- **Missing Values**
- **Outliers**
- **Out of Domain Values**
- **Flip Labels**
- **Duplicate Rows**
- **Add Rows**

L'analisi degli esperimenti in questa presentazione tratterà solo una parte significativa

I parametri utilizzati saranno elencati dopo le conclusioni in una slide riassuntiva

Drop Features

Vengono droppate le colonne corrispondenti alle componenti PCA



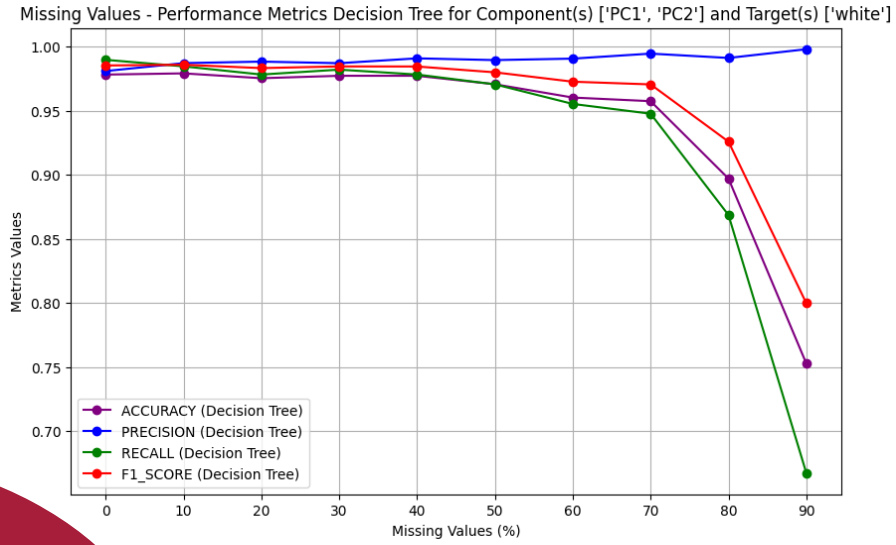
$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

- PC1 e PC2 più significative, abbiamo un impatto sulle performance se mancanti, soprattutto PC1.
- Recall molto alta
  e Precision segue l'opposto, in relazione a TP, FP e FN
- ~0.75 di minimo per Precision e Accuracy

Missing Values

Vengono introdotti valori mancanti in %, in base al target e componenti



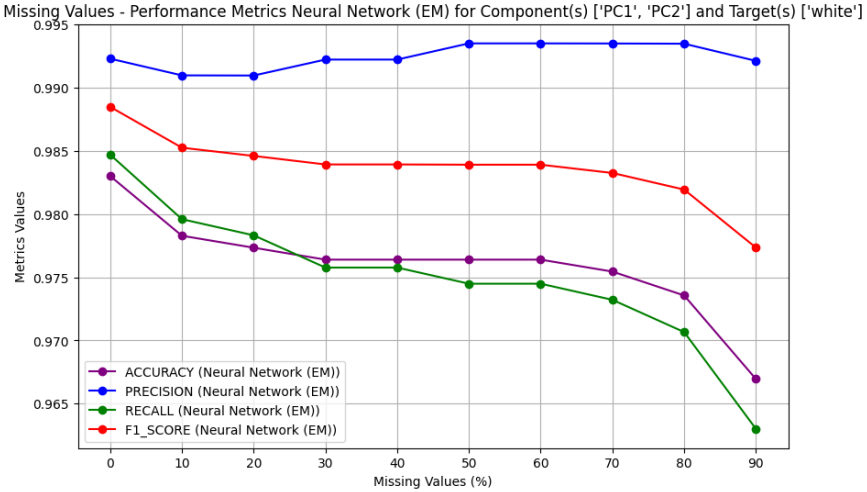
- Per Rete Neurale e SVM incompatibili:
  - **Media**:
    + Imputazione Semplice
  - **Expectation Maximization:**
    + Imputazione Iterativa

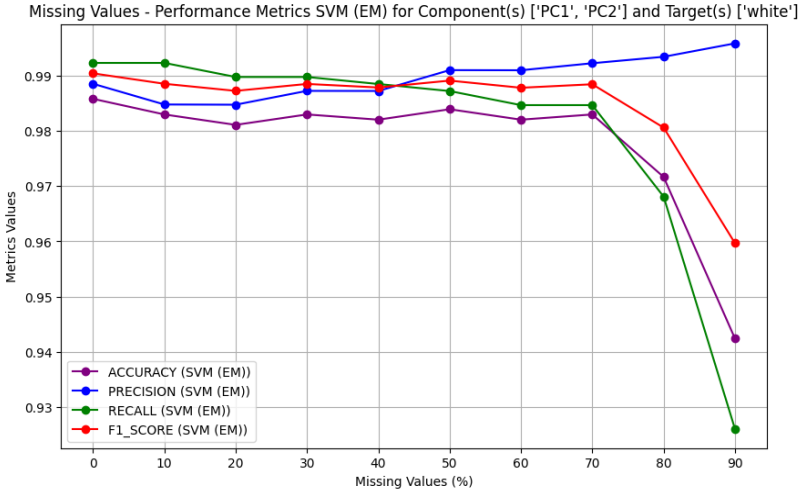
  Il DTC supporta i missing values

- **Target Considerato: White**

- ~0.70 di minimo per DTC su Recall

Missing Values

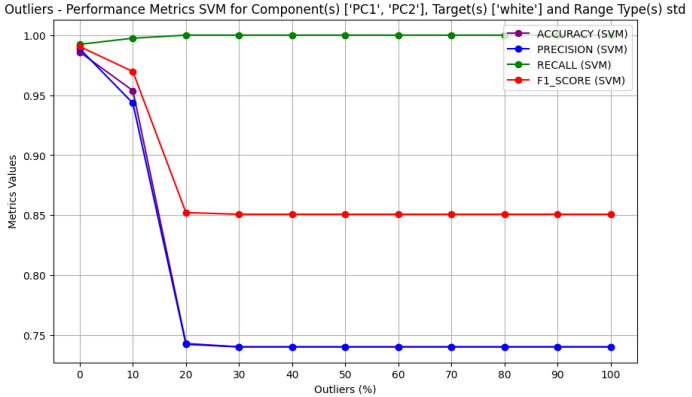


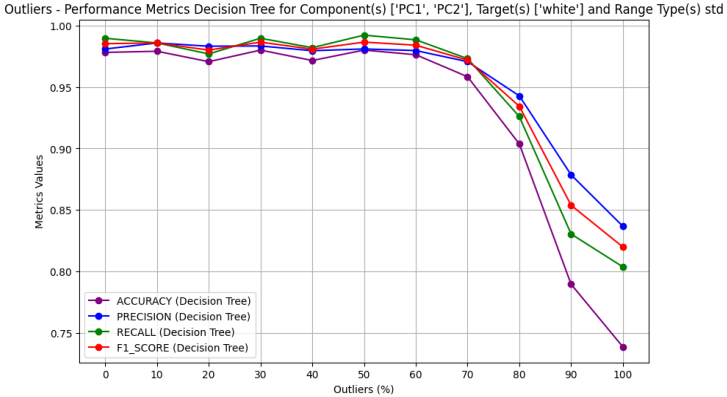


- NN e SVM mantengono metriche sopra il 0.90 per entrambi i target
- Riportati solo i risultati con Expectation Maximization in quanto migliori metriche

Outliers

Vengono introdotti valori nel dominio ma fuori dalla distribuzione (ranges)





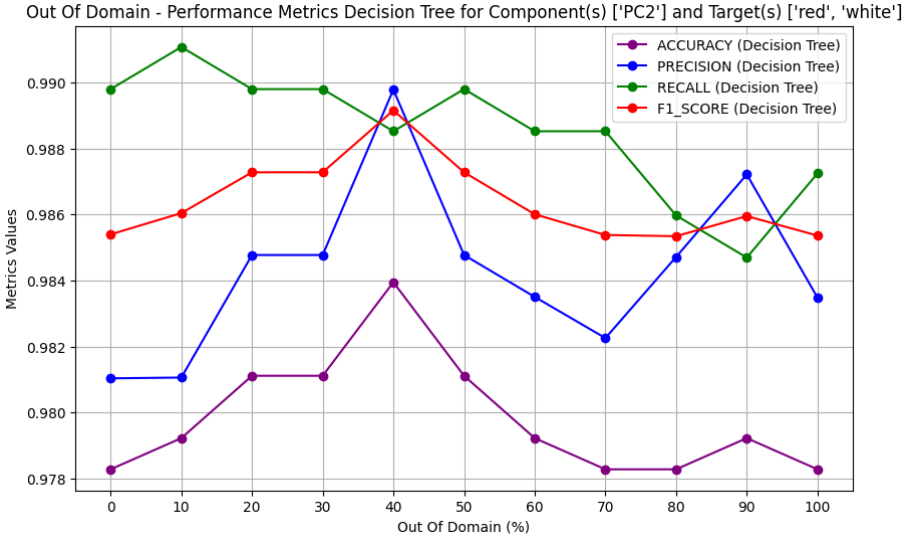
- Due Approcci:
  - Media ± 3 Deviazione Std
  - Interquartile Range:

  (Q1 - 2 * IQR, Q3 + 2 * IQR)

- DTC scende in modo più graduale in base alla percentuale rispetto a NN e SVM
- Tutti modelli toccano un minimo di ~0.75
- Comportamento simile ai precedenti per Precision e Recall

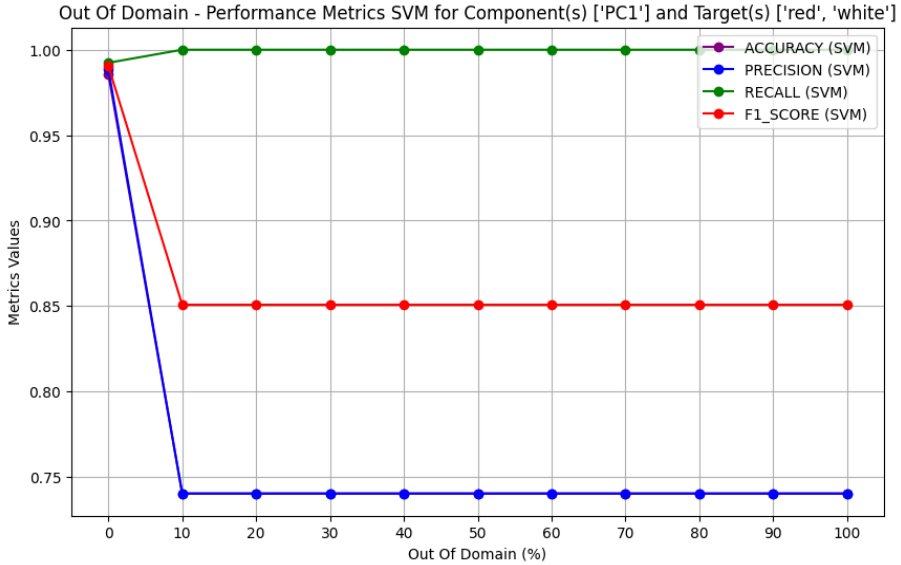
Out of Domain Values

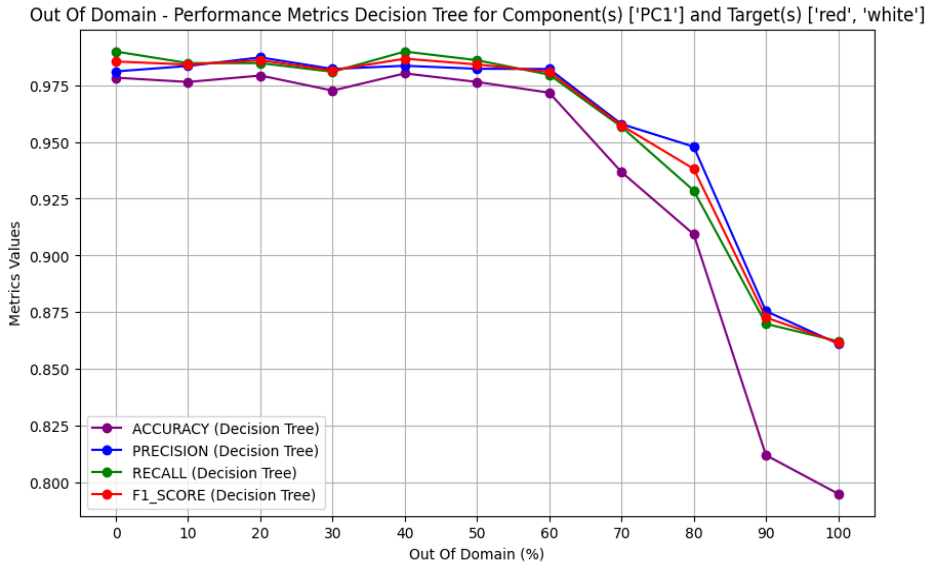
Vengono introdotte percentuali di valori largamente fuori "dominio" della PCA



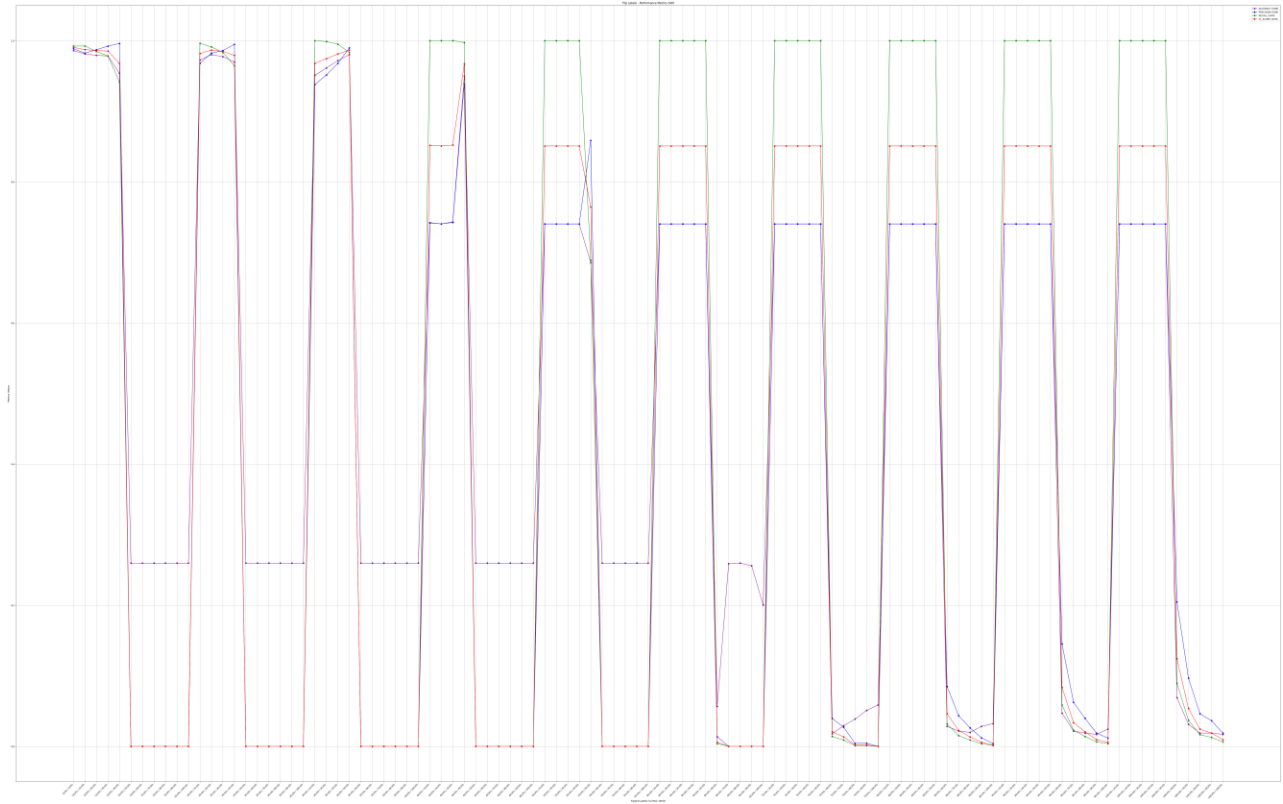
- Definizione "Fuori Dominio": **Media ± 10 Deviazione Std**
- Le performance tendono a subire un leggero aumento al crescere della percentuale con un picco al 40% tranne che per la Recall

Out of Domain Values

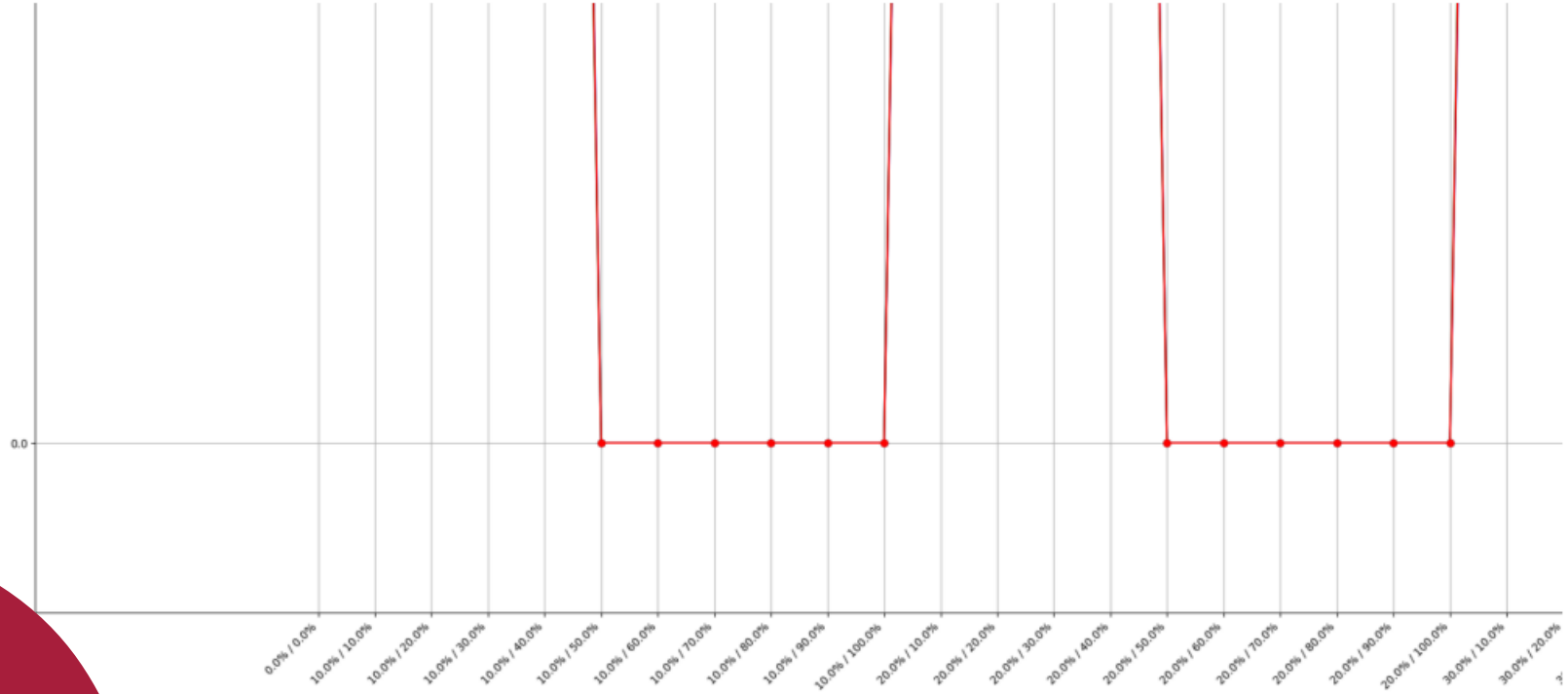




Flip Labels

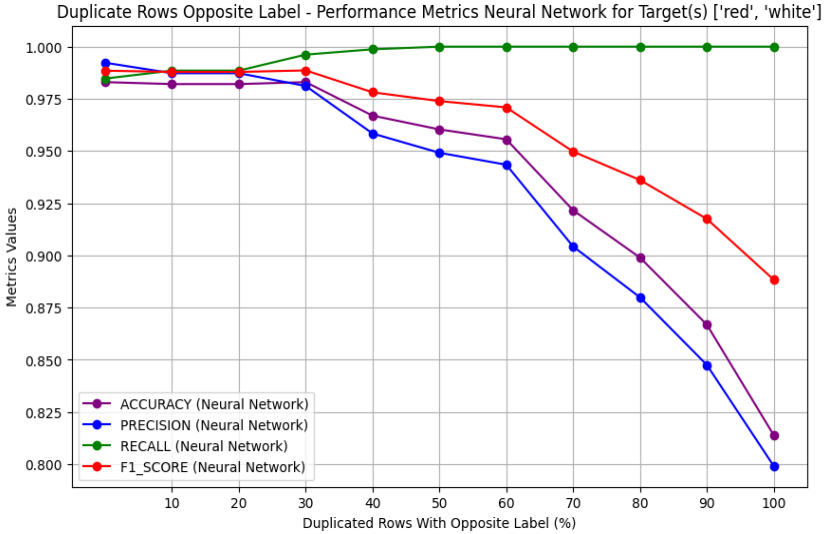


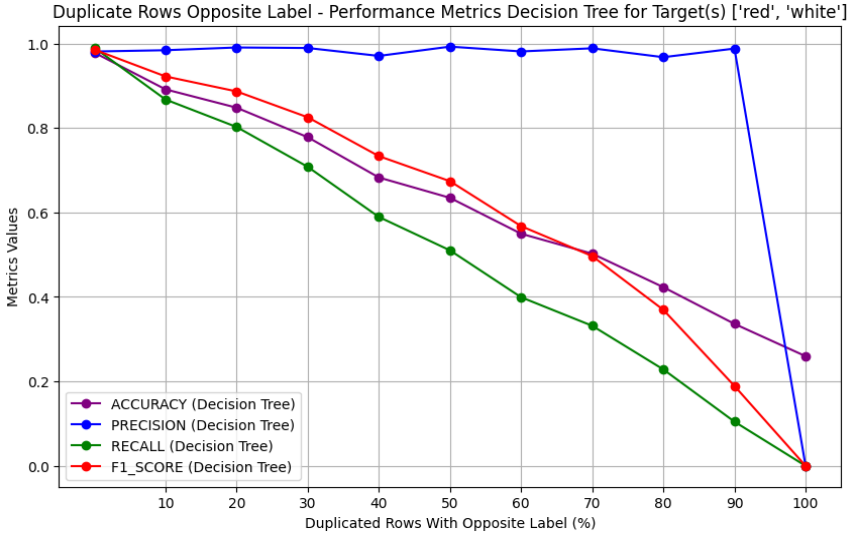
Flip Labels


0.0
0.0% / 0.0%
10.0% / 10.0%
10.0% / 20.0%
10.0% / 30.0%
10.0% / 40.0%
10.0% / 50.0%
10.0% / 60.0%
10.0% / 70.0%
10.0% / 80.0%
10.0% / 90.0%
10.0% / 100.0%
20.0% / 10.0%
20.0% / 20.0%
20.0% / 30.0%
20.0% / 40.0%
20.0% / 50.0%
20.0% / 60.0%
20.0% / 70.0%
20.0% / 80.0%
20.0% / 90.0%
20.0% / 100.0%
30.0% / 10.0%
30.0% / 20.0%

Duplicate Rows Opposite Label

Duplichiamo una certa percentuale di istanze del dataset invertendo la label

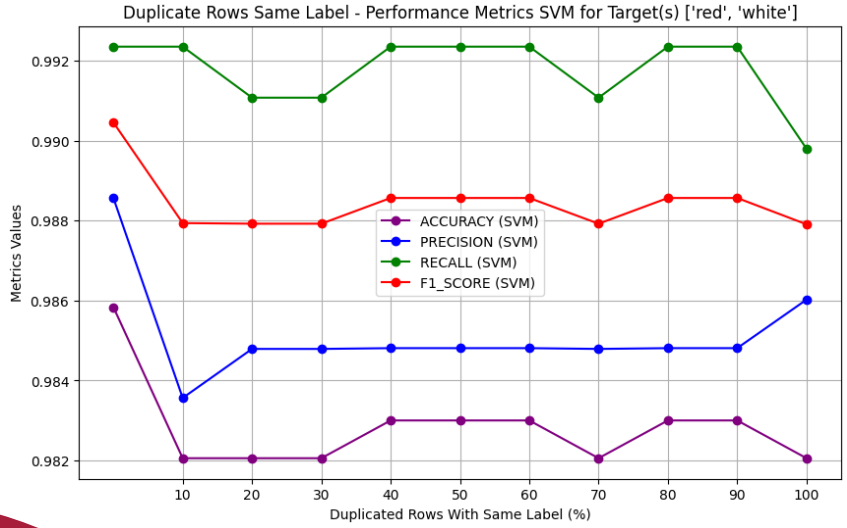


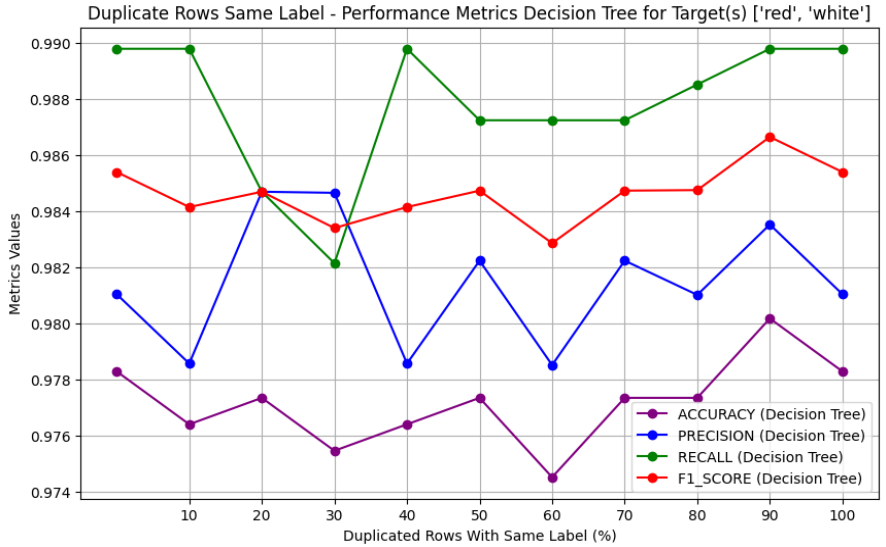


La NN diminuisce nelle prestazione di un 20% circa,
il DTC subisce invece un decremento costante fino ad arrivare allo 0%

Duplicate Rows Same Label

Duplichiamo una certa percentuale di istanze del dataset mantenendo la stessa label

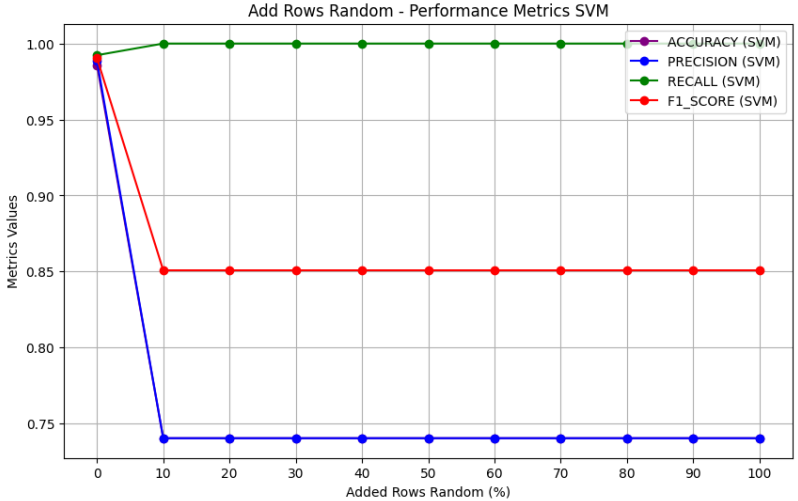


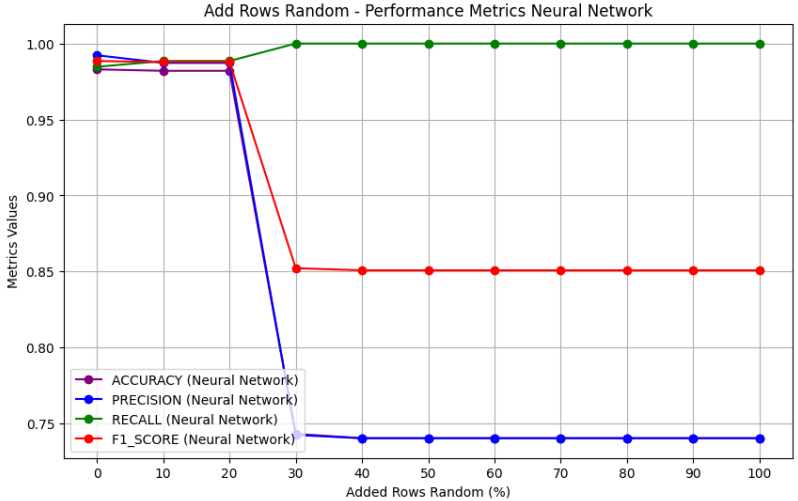


Osserviamo un leggerissimo decremento delle performance.
Nel DTC le variazioni sono più imprevedibili.

Add Rows Random

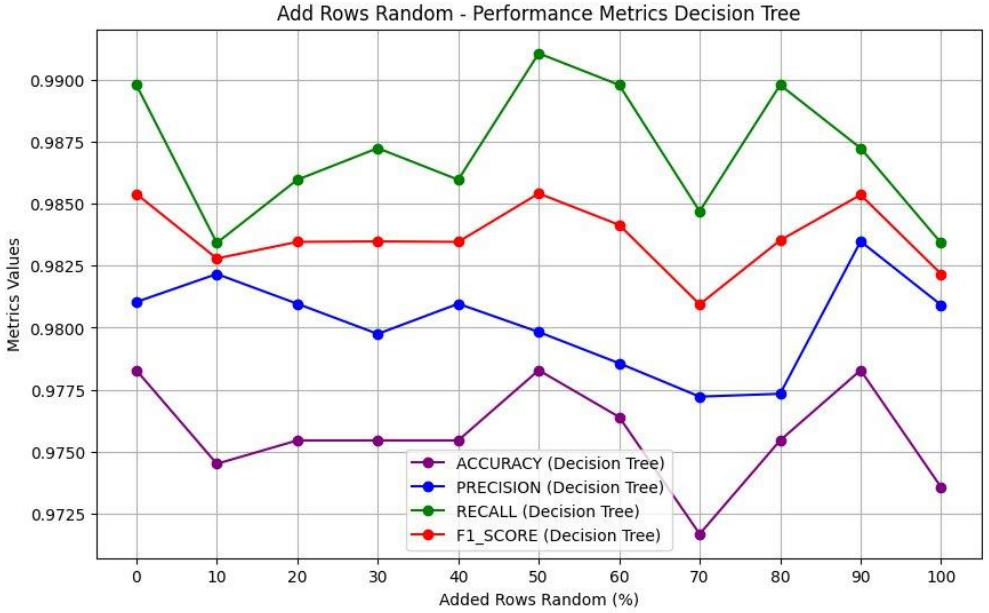
Vengono aggiunte righe con features random in (-100, 100) e target random





- **SVM** è **simile a NN** ma la sua discesa si ha al 10% al posto del 30%
- Recall = 1 → non ci sono vini bianchi classificati come rossi erroneamente

Add Rows Random

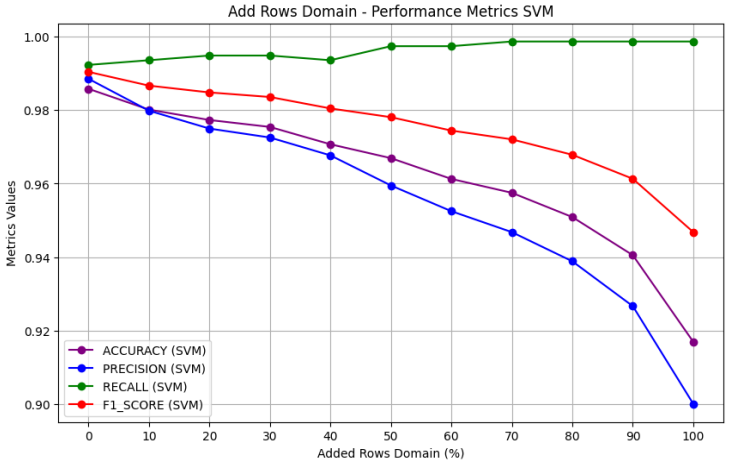


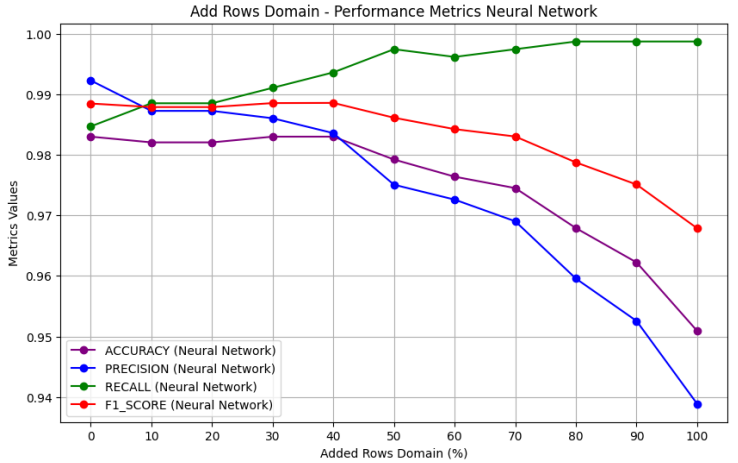
- **DTC** risulta **più robusto**, con piccole variazioni dovute agli split diversi

- Al variare delle percentuali, Precision rimane minore di Recall, dunque FP > FN (**più vini** rossi classificati **erroneamente** come **bianchi**)

Add Rows Domain

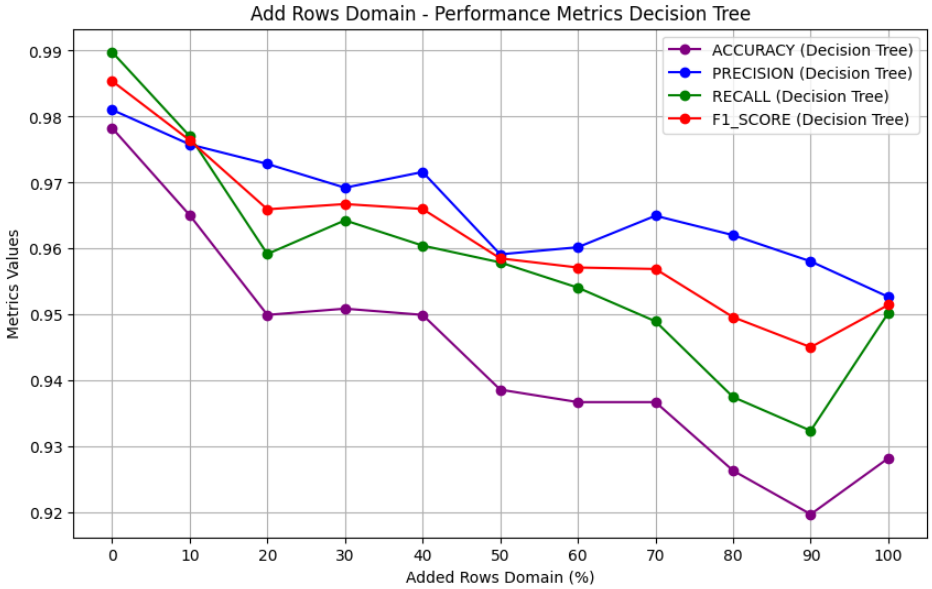
Vengono aggiunte righe con features random in Mean ± 3 * Std e target random





- **NN** presenta metriche **migliori di SVM**
- Al 100% di righe aggiunte: Accuracy NN di circa 0.95 vs circa 0.92 di SVM
- Recall tende a 1 → pochi vini bianchi classificati come rossi erroneamente

Add Rows Domain

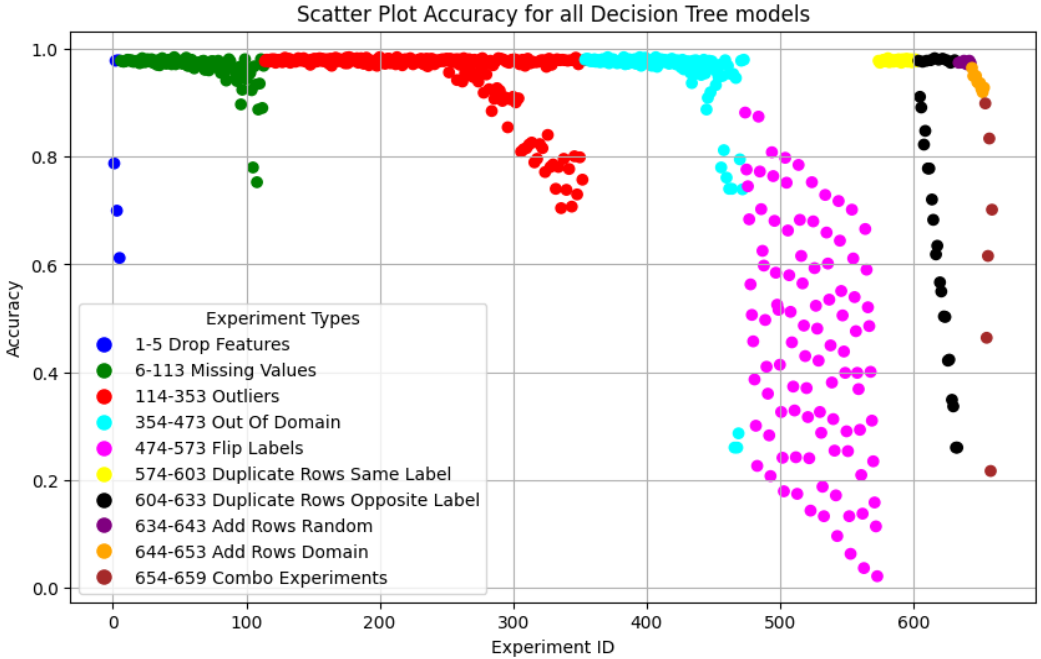


- Confrontando solo l'Accuracy, DTC questa volta non è il più robusto (Accuracy 0.95 già al 20%)
- Tuttavia, al 100% Precision e Recall si avvicinano
  → modello **efficace**

$$Precision = \frac{TP}{TP + FP}$$

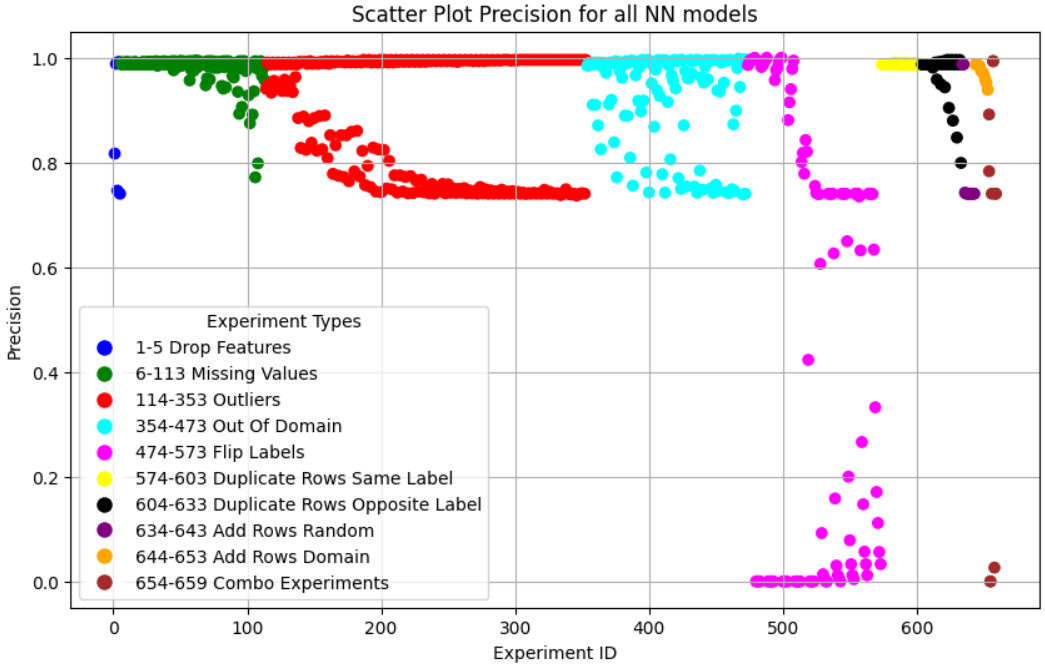
$$Recall = \frac{TP}{TP + FN}$$

Considerazioni e Conclusioni (1)



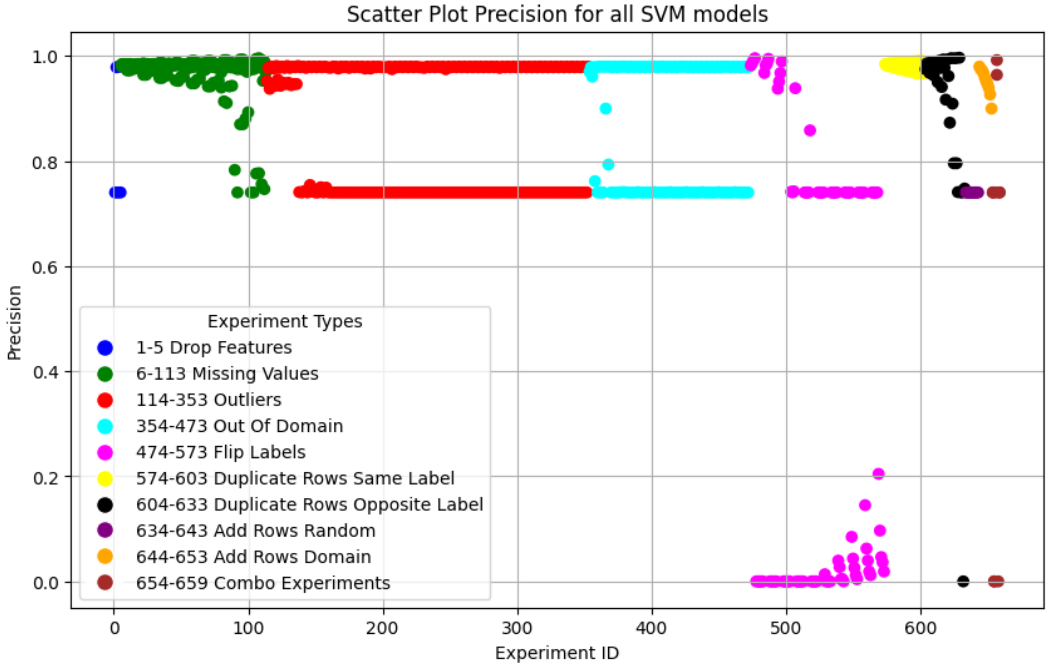
- Scatter plot che riassume l'andamento dell'**Accuracy di DTC**
- Effetto "cascata"

Considerazioni e Conclusioni (2)



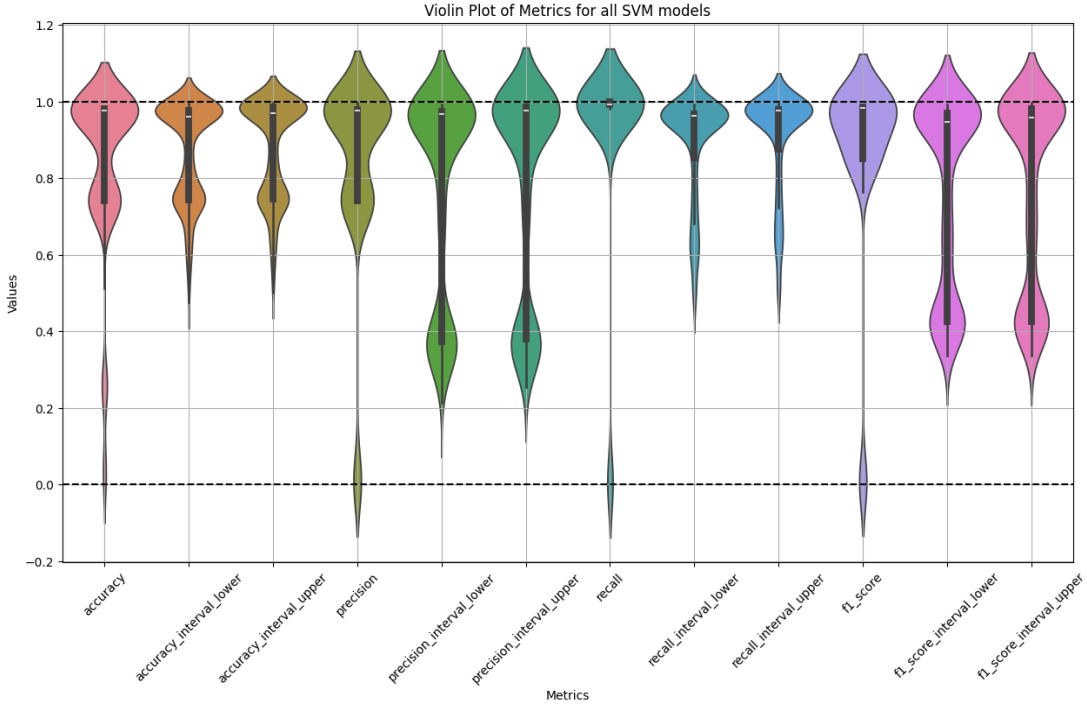
- Scatter plot che riassume l'andamento dell'**Accuracy di NN**
- **NN** ha **discese graduali, ma in anticipo** rispetto a DTC

Considerazioni e Conclusioni (3)



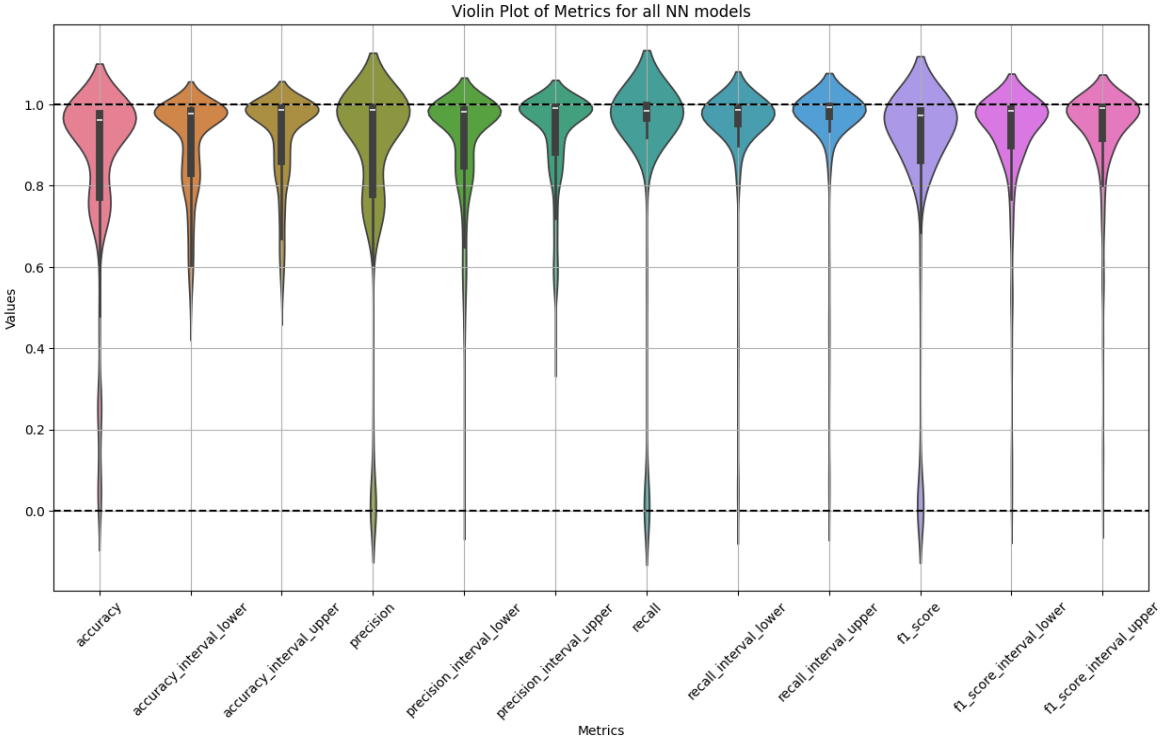
- Scatter plot che riassume l'andamento dell'**Accuracy di SVM**
- **SVM** si stabilizza **a livelli** in maniera "brusca"

Considerazioni e Conclusioni (4)



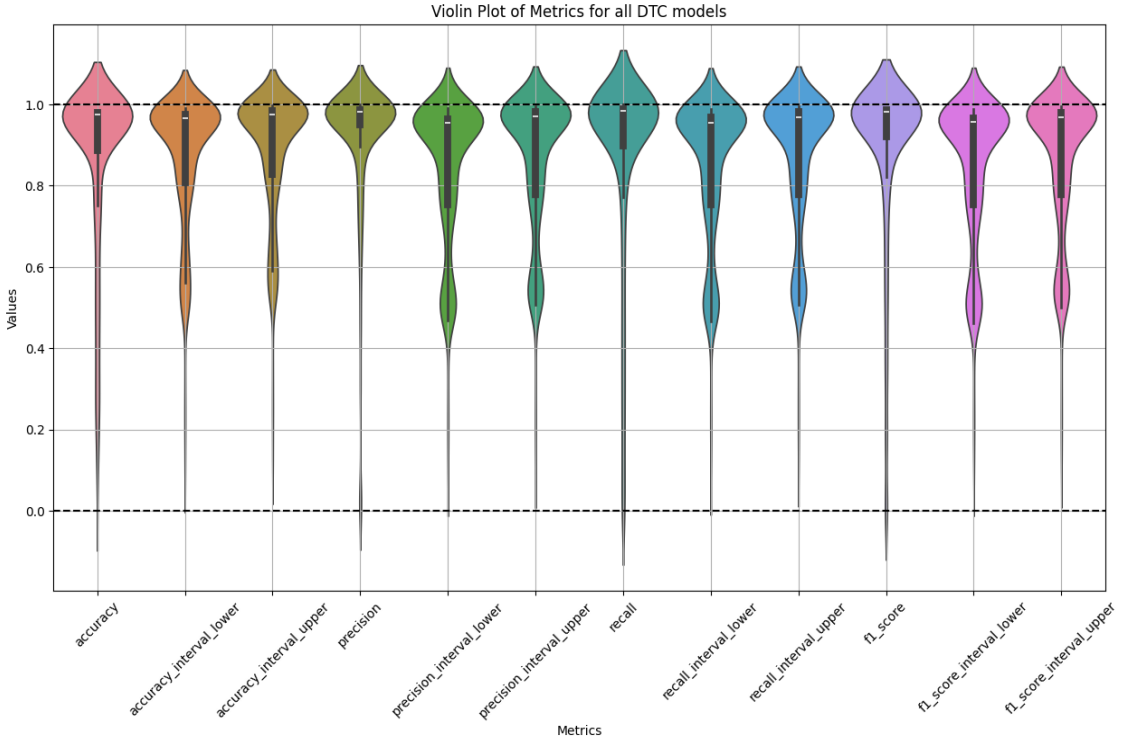
- Metriche **SVM**
- **Precision < Recall**: le classificazioni errate riguardano maggiormente la classe white
- **Molto sensibile**

Considerazioni e Conclusioni (5)



- Metriche **NN**
- **Meno sensibile** di SVM in Cross Validation

Considerazioni e Conclusioni (6)



- Metriche **DTC**
- L'Accuracy più alta degli altri modelli suggerisce **maggior robustezza**

Considerazioni e Conclusioni (7)

1. **DTC** è **il modello più robusto**: presenta meno classificazioni errate in generale (Accuracy), in particolare rispetto agli altri si hanno meno errori per la classe white (Precision). Tuttavia, è anche molto variabile.

2. **NN** è un buon compromesso tra SVM e DTC.

3. **SVM** è **il più sensibile** alla variazione della qualità, ma l'Accuracy si stabilizza su 0.75.

Bonus: Esperimento in real-time

Le percentuali sono tra 0 e 1, le liste di features possono contenere solo "PC1", "PC2", "PC3", "PC4", "PC5", le liste di classi di vino possono contenere solo "red", "white", il tipo di range è una stringa tra "std" e "iqr". Di seguito tutti i possibili parametri con i valori di default:

- **experiment-name**: stringa obbligatoria.

- **features-to-drop**: [].

- **features-to-dirty-mv**: [].
- **missing-values-percentage**: 0.0.
- **wine-types-to-consider-missing-values**: ["red, "white"].

- **features-to-dirty-outliers**: [].
- **outliers-percentage**: 0.0.
- **wine-types-to-consider-outliers**: ["red", "white"].
- **range-type**: "std".

- **features-to-dirty-oodv**: [].
- **oodv-percentage**: 0.0.
- **wine-types-to-consider-oodv**: ["red", "white"].

- **flip-percentage-red**: 0.0.
- **flip-percentage-white**: 0.0.

- **wine-types-to-consider-same-label**: ["red", "white"].
- **duplicate-rows-same-label-percentage**: 0.0.

- **wine-types-to-consider-opposite-label**: ["red", "white"].
- **duplicate-rows-opposite-label-percentage**: 0.0.

- **add-rows-random-percentage**: 0.0.

- **add-rows-domain-percentage**: 0.0.