UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

SISTEMA INTELLIGENTE PER LA CLASSIFICAZIONE DELLA QUALITÀ DEL VINO BIANCO

Realizzato da:

Claudio Valenziano

Matr: 706467

GitHub Repository:

https://github.com/claudiovalen/progetto-icon.git

IDE UTILIZZATO:

- PYCHARM

LIBRERIE IMPORTATE:

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Scikit-learn

INTRODUZIONE:

Il caso di studio mira ad aiutare le filiere produttive di vino bianco nella determinazione della qualità del vino prodotto. In Italia, ad esempio, ci distinguiamo per essere i primi produttori al mondo di vino, ma non siamo ancora del tutto abituati all'utilizzo degli strumenti tecnologici a nostra disposizione. In particolare, questo sistema può fungere da supporto alla figura dell'enologo nella valutazione del vino, aiutandolo nell'effettuare controlli che richiedono tempo e meticolosità.

Quindi, attraverso un ampio set di dati contenente le proprietà chimiche del vino, il sistema è stato addestrato a riconoscerne la buona o cattiva qualità.

DATASET:

Il dataset importato, presente nel repository GitHub, contiene le proprietà chimiche del vino bianco che ne permettono la classificazione di qualità:

1 - acidità fissa: la maggior parte degli acidi coinvolti nel vino o fissa o non volatile (non evapora facilmente)

2 - acidità volatile: la quantità di acido acetico nel vino, che a livelli troppo alti può portare ad uno sgradevole sapore di aceto

3 - acido citrico: presente in piccole quantità, l'acido citrico può aggiungere 'freschezza' e sapore ai vini

4 - zucchero residuo: la quantità di zucchero che rimane dopo l'arresto della fermentazione, è raro trovare vini con meno di 1 grammo/litro e vini con più di 45 grammi/litro sono considerati dolci

5 - cloruri: la quantità di sale nel vino

6 - anidride solforosa libera: la forma libera di SO2 esiste in equilibrio tra SO2 molecolare (come gas disciolto) e ione bisolfito; previene la crescita microbica e l'ossidazione del vino

7 - anidride solforosa totale: quantità di forme libere e legate di S02; a basse concentrazioni, l'SO2 è per lo più non rilevabile nel vino, ma a concentrazioni di SO2 libera superiori a 50 ppm, l'SO2 diventa evidente nel naso e nel gusto del vino

8 - densità: la densità del vino è vicina a quella dell'acqua a seconda della percentuale di alcol e zucchero

9 - pH: descrive quanto è acido o basico un vino su una scala da 0 (molto acido) a 14 (molto basico); la maggior parte dei vini sono tra 3-4 sulla scala del pH

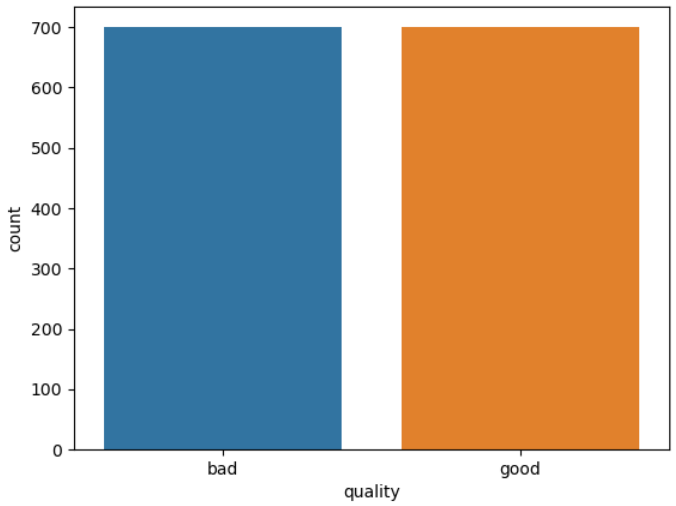
10 - solfati: additivo del vino che può contribuire ai livelli di anidride solforosa gassosa (S02), che agisce come antimicrobico e antiossidante

11 - alcol: la gradazione alcolica del vino

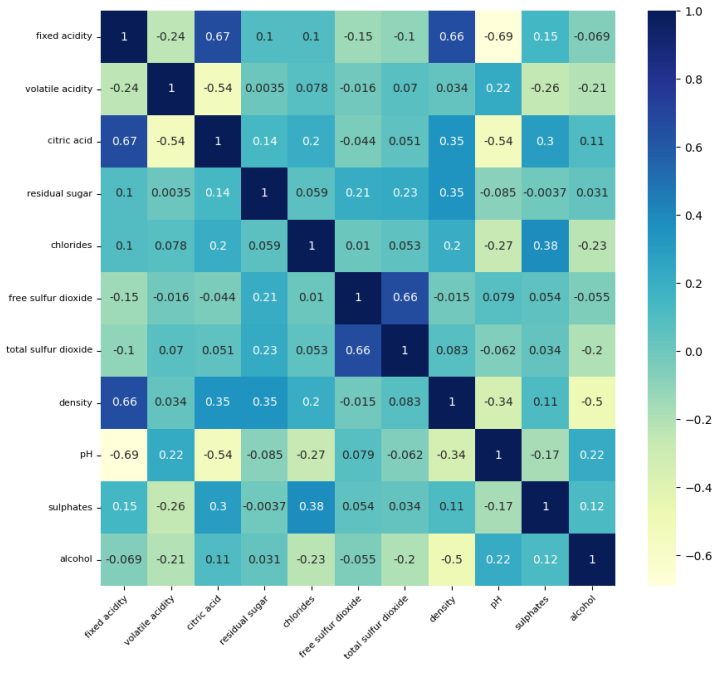
12 – quality: buona o cattiva qualità del vino

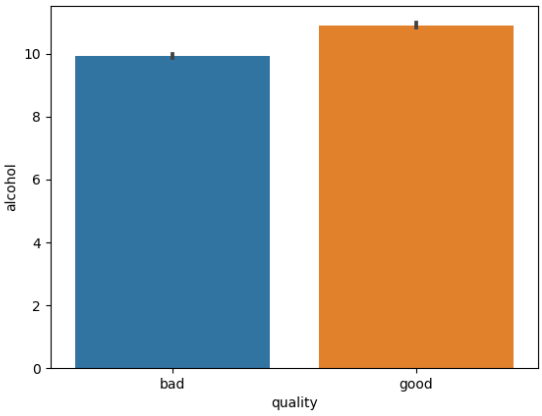
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
1	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad
2	7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968	3.2	0.68	9.8	bad
3	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8	bad
4	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad
5	7.4	0.66	0.0	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	bad
6	7.9	0.6	0.06	1.6	0.069	15.0	59.0	0.9964	3.3	0.46	9.4	bad
7	7.3	0.65	0.0	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10.0	good
8	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9.5	good
9	5.6	0.615	0.0	1.6	0.08900000000000001	16.0	59.0	0.9943	3.58	0.52	9.9	bad
10												

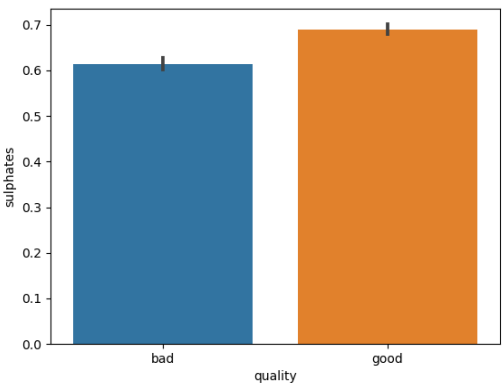
Come si può vedere, la distribuzione delle classi di qualità all'interno del dataset è perfettamente bilanciata:

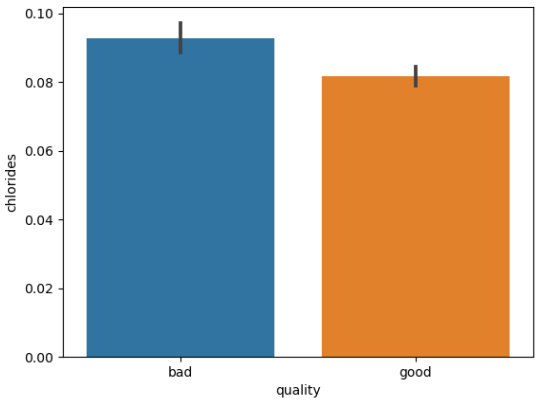


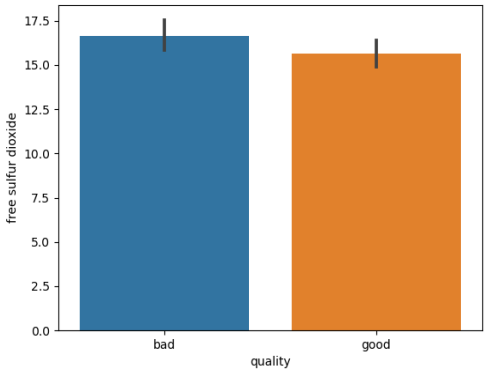
CORRELAZIONE DATI E INFERENZA:

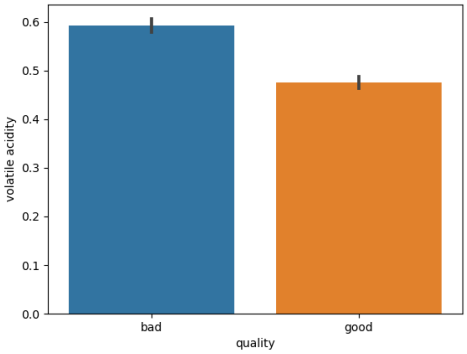


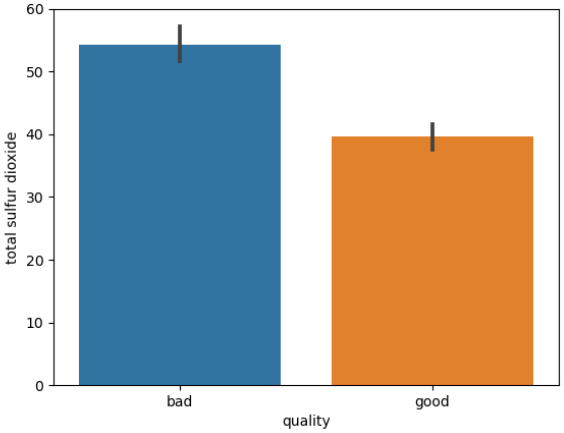


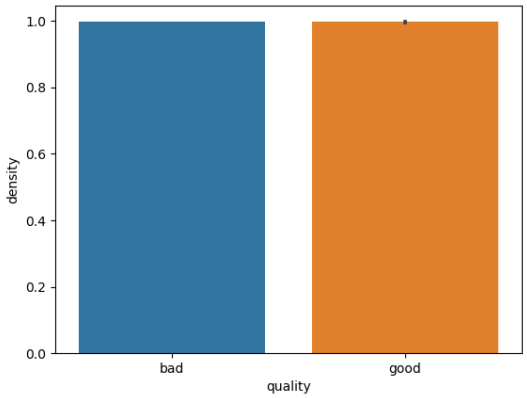


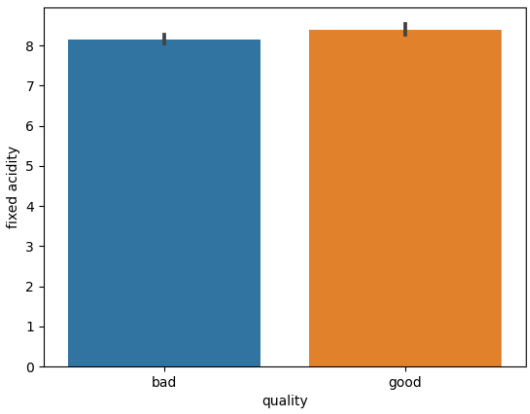

chlorides
0.10
0.08
0.06
0.04
0.02
0.00
bad
good
quality

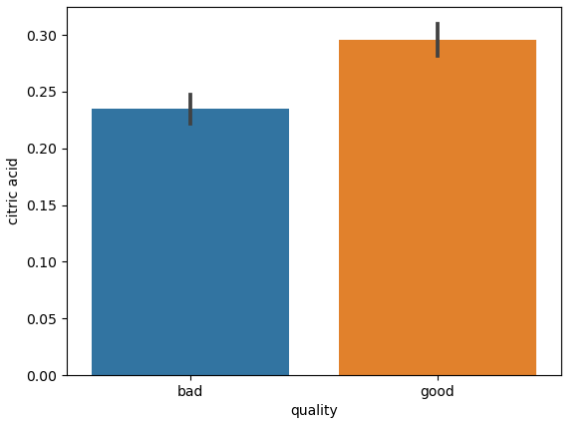

free sulfur dioxide
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
bad
good
quality

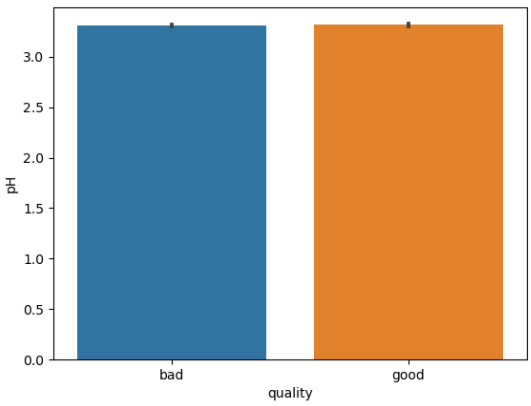

volatile acidity
0.6
0.5
0.4
0.3
0.2
0.1
0.0
bad
good
quality

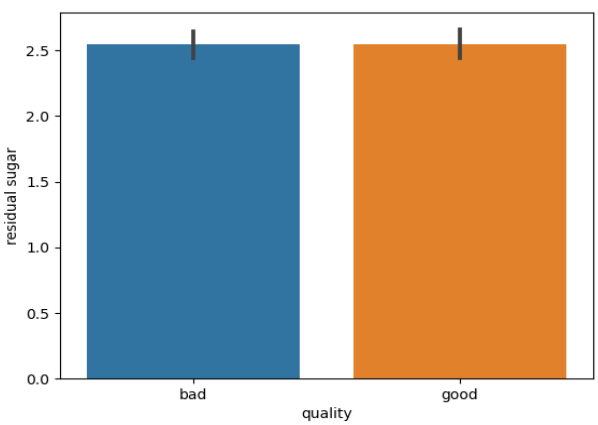

total sulfur dioxide
60
50
40
30
20
10
0
bad
good
quality


density
1.0
0.8
0.6
0.4
0.2
0.0
bad
good
quality


fixed acidity
8
7
6
5
4
3
2
1
0
bad
good
quality


citric acid
0.30
0.25
0.20
0.15
0.10
0.05
0.00
bad
good
quality


pH
3.0
2.5
2.0
1.5
1.0
0.5
0.0
bad
good
quality


residual sugar
2.5
2.0
1.5
1.0
0.5
0.0
bad
good
quality

```
plt.figure(figsize=(10,9))
map =sb.heatmap(df.corr(), annot=True, cmap="YlGnBu",xticklabels=True, yticklabels=True)
map.set_yticklabels(map.get_yticklabels(), rotation=0, fontsize=8)
map.set_xticklabels(map.get_xticklabels(), rotation=45, fontsize=8, rotation_mode='anchor', ha='right')
plt.show()

for i, col in enumerate(df.drop('quality',axis=1)):
    plt.figure(i)
    sb.barplot(x='quality', y =col, data=df)
    plt.show()
```

Emerge dunque che:

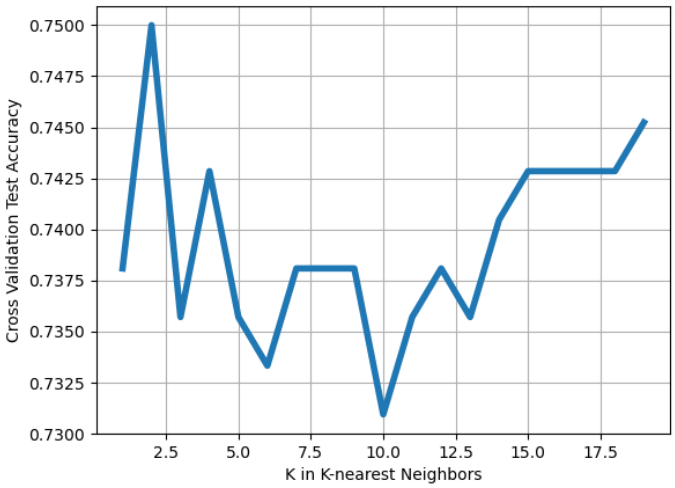
- Ci sono molti dati che non manifestano differenze nei valori a seconda della qualità, come ad esempio più di tutti il pH, lo zucchero residuo e la densità, che posso quindi non considerare nella classificazione, riducendo così il numero elevato di features.
- L'anidride solforosa totale, l'acidità volatile e l'acido citrico sembrano avere una correlazione più elevata con la qualità.
- Risulta che più alcool, maggior presenza di solfati e acido citrico comportano una qualità superiore, mentre maggior anidride solforosa totale e più acidità volatile portano ad una qualità peggiore.
- Non vi è una feature che ricopre un ruolo dominante nella determinazione della classe, anzi, i valori risultano in alcuni casi molto vicini.

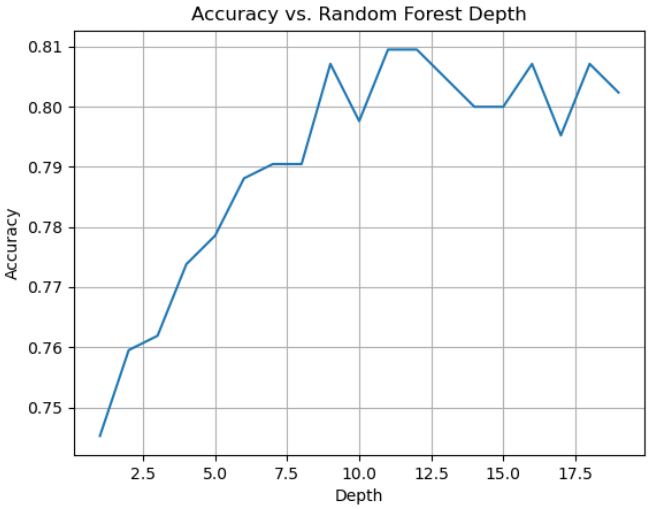
CLASSIFICAZIONE QUALITÀ:

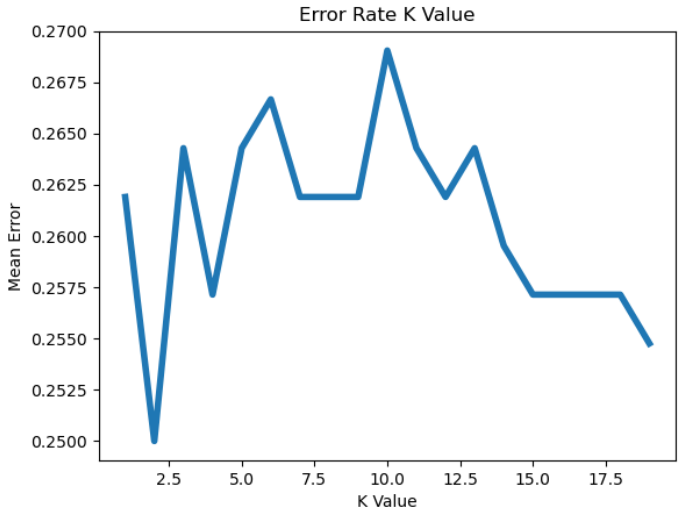
```
# Accuracy VS KNN
scores = []
error = []
for k in interval:
    knn = KNeighborsClassifier(p=k)
    knn.fit(X_train, Y_train)
    predictions = knn.predict(X_test)
    scores.append(accuracy_score(Y_test,predictions))
    error.append(np.mean(predictions!= Y_test))

plt.plot(interval, scores, linewidth=4, markersize=10)
plt.grid()
plt.title('Accuracy vs. KNN')
plt.xlabel("K in K-nearest Neighbors")
plt.ylabel("Cross Validation Test Accuracy")
plt.show()
plt.plot(interval,error,linewidth=4, markersize=10)
plt.title('Error Rate K Value')
plt.xlabel('K Value')
plt.ylabel('Mean Error')
plt.show()
```

```
# Accuracy VS Depth Random forest Plot
accuracy = []
interval = np.arange(1, 20)
for i in interval:
    clf = RandomForestClassifier(max_depth=i, random_state=10)
    clf.fit(X_train, Y_train)
    predictions = clf.predict(X_test)
    accuracy.append(accuracy_score(Y_test, predictions))

plt.plot(interval, accuracy, linewidth=4, markersize=10)
plt.grid()
plt.title('Accuracy vs. Random Forest Depth')
plt.xlabel('Depth')
plt.ylabel('Accuracy')
plt.show()
```

Per la classificazione della qualità del vino, dopo aver standardizzato le variabili numeriche ed aver rimosso le features superflue, ho utilizzato i seguenti algoritmi: il KNN e il Random Forest. Innanzitutto, ho verificato quale fosse la profondità ideale del Random Forest e il valore più appropriato per l'iperparametro K del KNN che garantissero il miglior risultato. Come si può notare dai risultati ottenuti, il Random Forest ha garantito una precisione più elevata:







Dall'analisi delle prestazioni degli algoritmi emerge che l'algoritmo Random Forest ottiene risultati migliori con una profondità compresa tra 11 e 12, mentre il KNN con un valore K tra 1 e 2. Utilizzando questi valori ottengo quindi le seguenti prestazioni e le seguenti matrici di confusione, che mi confermano la maggior efficienza dell'algoritmo Random Forest:

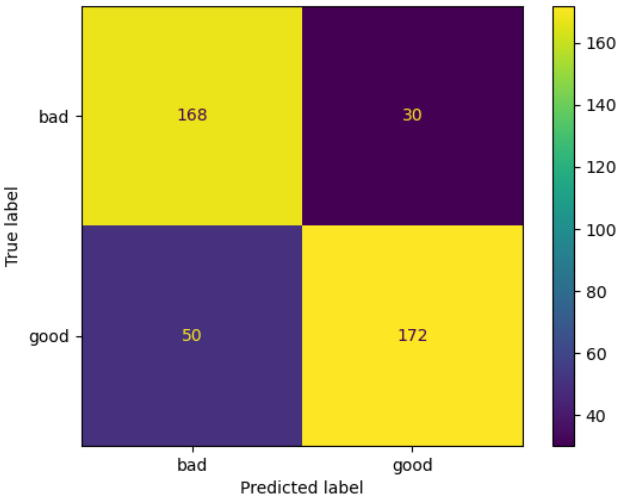
```
#Create a function within many Machine Learning Models
def models(X_train,Y_train,X_test,Y_test):

    #Using the Nearest Neighbor algorithm
    knn = KNeighborsClassifier(p=1)
    knn.fit(X_train, Y_train)
    knn_pred = knn.predict(X_test)
    knn_classification = classification_report(Y_test,knn_pred)

    #Using the Random Forest Classification algorithm
    forest = RandomForestClassifier(max_depth=11,random_state=10)
    forest.fit(X_train, Y_train)
    forest_pred = forest.predict(X_test)
    forest_classification = classification_report(Y_test,forest_pred)
```

```
model= models(X_train,Y_train,X_test,Y_test)
for i in range(len(model)):
    cm = plot_confusion_matrix(model[i],X_test, Y_test)
    plt.show()
```

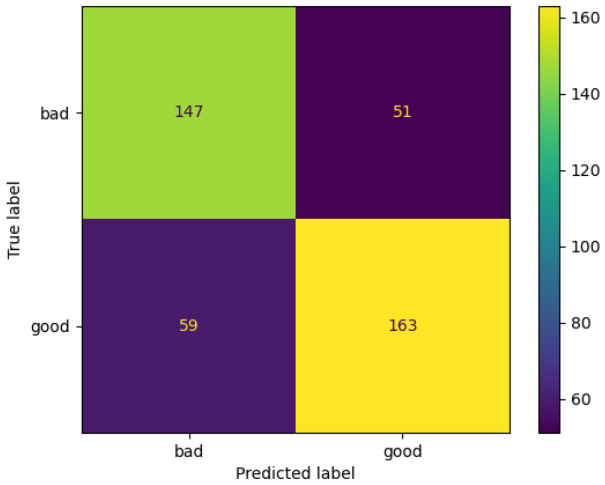
```
[6]Random Forest Classifier Test Accuracy:
              precision    recall  f1-score   support

         bad       0.77      0.85      0.81       198
        good       0.85      0.77      0.81       222

    accuracy                           0.81       420
   macro avg       0.81      0.81      0.81       420
weighted avg       0.81      0.81      0.81       420
```

```
[1]K Nearest Neighbor Test Accuracy:
              precision    recall  f1-score   support

         bad       0.71      0.74      0.73       198
        good       0.76      0.73      0.75       222

    accuracy                           0.74       420
   macro avg       0.74      0.74      0.74       420
weighted avg       0.74      0.74      0.74       420
```

RANDOM FOREST:



KNN:



CONCLUSIONE:

Si può pensare ad una futura implementazione del progetto anche in relazione alla nostra stessa regione Puglia, grande produttrice di vini in Italia, ma sicuramente meno al passo di altre regioni sul fronte tecnologico. Oltre ad un ausilio nella produzione del vino, identificando quali attributi portano a valutazioni di qualità più alte o più basse, i risultati potrebbero essere utilizzati per aiutare i ristoranti a prevedere più facilmente quali vini apprezzeranno i clienti. Inoltre, i risultati potrebbero essere integrati in un sito Web o in un'app per aiutare le persone a trovare i vini entro il loro budget, la varietà preferita di vino, ecc. Ciò aiuterebbe sia le persone a determinare quali vini hanno maggiori probabilità di gustare prima dell'acquisto, sia per aiutare a trovare nuovi vini basati su vini simili già precedentemente graditi. Si può inoltre pensare ad una implementazione di questo sistema che analizzi anche i dati che invece influenzano la qualità dei vini rossi.