HEART DISEASE PREDICTION

Progetto di Ingegneria della Conoscenza

Palermo Sandro, Matricola: 738749
Email: s.palermo8@studenti.uniba.it

Link repository Github:
https://github.com/Sapalermo/Progetto-Icon-Heart-Disease

Sommario

Introduzione

L'obiettivo di questo progetto consiste nel predire, attraverso l'analisi delle informazioni contenute nel dataset e i valori delle feature di input, la presenza di una patologia cardiaca in un individuo.
Inoltre, all'interno di questo progetto ci si occupa dell'inferenza probabilistica, ovvero la creazione di una Belief Network che consente al sistema di rispondere query probabilistiche definite dall'utente.

Requisiti funzionali

Questo progetto è stato realizzato in Python poiché esso risulta più funzionale per l'analisi e il trattamento dei dati. L'ambiente di lavoro utilizzato è Colab.

Librerie utilizzate

Le librerie utilizzate all'interno del progetto sono:

- **Pandas**: usata per l'importazione del Dataset in formato csv
- **Numpy**: usata per la visualizzazione dei grafici presenti nel progetto
- **Scikit-learn**: usata per applicare i concetti del Machine Learning
- **Pgmpy**: usata per la creazione della Belief Network
- **Matplotlib**: usata per la visualizzazione dei grafici presenti nel progetto

Dataset

Il dataset utilizzato, "heart.csv", contiene dati su circa 400 soggetti riguardo la presenza o meno di possibili patologie cardiache.

	age	gender	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	\
0	52	1	0	125	212	0	1	168	0	1.0	
1	53	1	0	140	203	1	0	155	1	3.1	
2	70	1	0	145	174	0	1	125	1	2.6	
3	61	1	0	148	203	0	1	161	0	0.0	
4	62	0	0	138	294	1	1	106	0	1.9	

	slope	ca	thal	target
0	2	2	3	0
1	0	0	3	0
2	0	0	3	0
3	2	1	3	0
4	1	3	2	0

Preprocessing dataset

Nella fase di preprocessing, il dataset viene modificato in modo da poterlo utilizzare correttamente. Il dataset non presenta problemi di mancanza dei dati.

Feature dicotomiche

Una feature dicotomica è una feature che presenta soltanto due valori come {si, no}, oppure {vero, falso}. Nel nostro caso, le features dicotomiche sono:

- Gender: sesso del soggetto (0 = donna, 1= uomo)
- Fbs: Glicemia a digiuno > 120 mg/dl (0 = no, 1 = si)
- Exang: angina indotta da esercizio (0 = no, 1 = si)

Feature presenti nel dataset

Le feature presenti nel dataset sono:

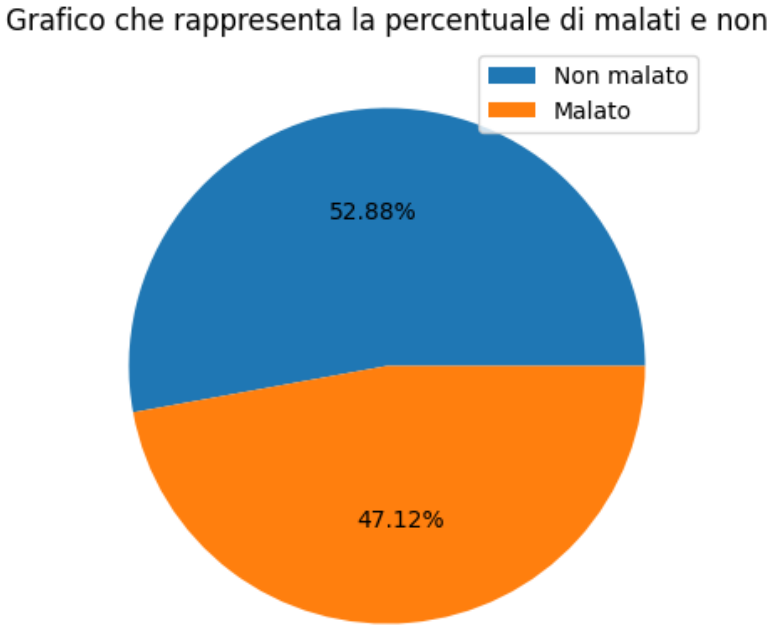
1. Age: età del soggetto
2. Gender: sesso del soggetto(0 = donna, 1 = uomo)
3. Cp: tipologia di dolore al petto (valori 0,1,2,3)
4. trestbps: Pressione del sangue a riposo
5. chol: Colesterolo totale sierico in mg/dl
6. fbs: Glicemia a digiuno > 120 mg/dl (0 = no, 1 = si)
7. restecg: risultati elettrocardiografici a riposo (valori 0,1,2)
8. thalach: frequenza cardiaca massima raggiunta
9. exang: angina indotta da esercizio (0 = no, 1 = si)
10. oldpeak: depressione del tratto ST indotta dall'esercizio rispetto al riposo
11. slope: la pendenza del segmento ST di picco dell'esercizio
12. ca: numero di vasi principali colorati da fluoroscopia(valori 0,1,2,3)
13. thal: 0 = normale; 1 = difetto fisso; 2 = difetto reversibile

Come feature target:

14. target: presenza di malattie cardiache nel paziente (0 = nessuna malattia, 1 = malattia)

Bilanciamento delle classi

Verifichiamo, attraverso un grafico, se il dataset è ben bilanciato (in funzione di 'target') in modo da riuscire ad avere ottimi risultati durante l'apprendimento.



Dal grafico possiamo notare che il dataset è abbastanza bilanciato, quindi in questo caso non c'è bisogno di effettuare un bilanciamento delle classi.

Apprendimento supervisionato

Scelta del modello
Sono stati utilizzati vari modelli per l'apprendimento:

- **KNN**: algoritmo utilizzato nel riconoscimento di pattern per la classificazione di oggetti basandosi sulle caratteristiche degli oggetti vicini a quello considerato.
- **Decision Tree**: tecnica utile per l'apprendimento della classificazione supervisionata. Un albero di decisione è un albero in cui ogni nodo interno (non foglia) è etichettato con una condizione, una funzione booleana sui valori delle feature degli esempi. Ogni nodo interno ha due figli, uno etichettato con true e l'altro con false. Ogni foglia dell'albero è etichettata con una stima puntuale sulla classe.
- **Random Forest**: classificatore d'insieme ottenuto dall'aggregazione tramite bagging di alberi di decisione. Esso si pone come soluzione che minimizza l'overfitting del training set rispetto agli alberi di decisione.
- **SVM**: modello utilizzato per task di classificazione che utilizza funzioni di dati di input originali come input delle funzioni lineari.

Le metriche di performance utilizzate nella valutazione sono:

- precision
- recall
- accuracy
- F1-score

Le prestazioni di questi classificatori vengono poi confrontate mediante una K-fold Cross Validation (con k fissato a 10), ossia un algoritmo che usa dati classificati come esempi di training per valutare il modello appreso prima dell'utilizzo degli esempi di test.

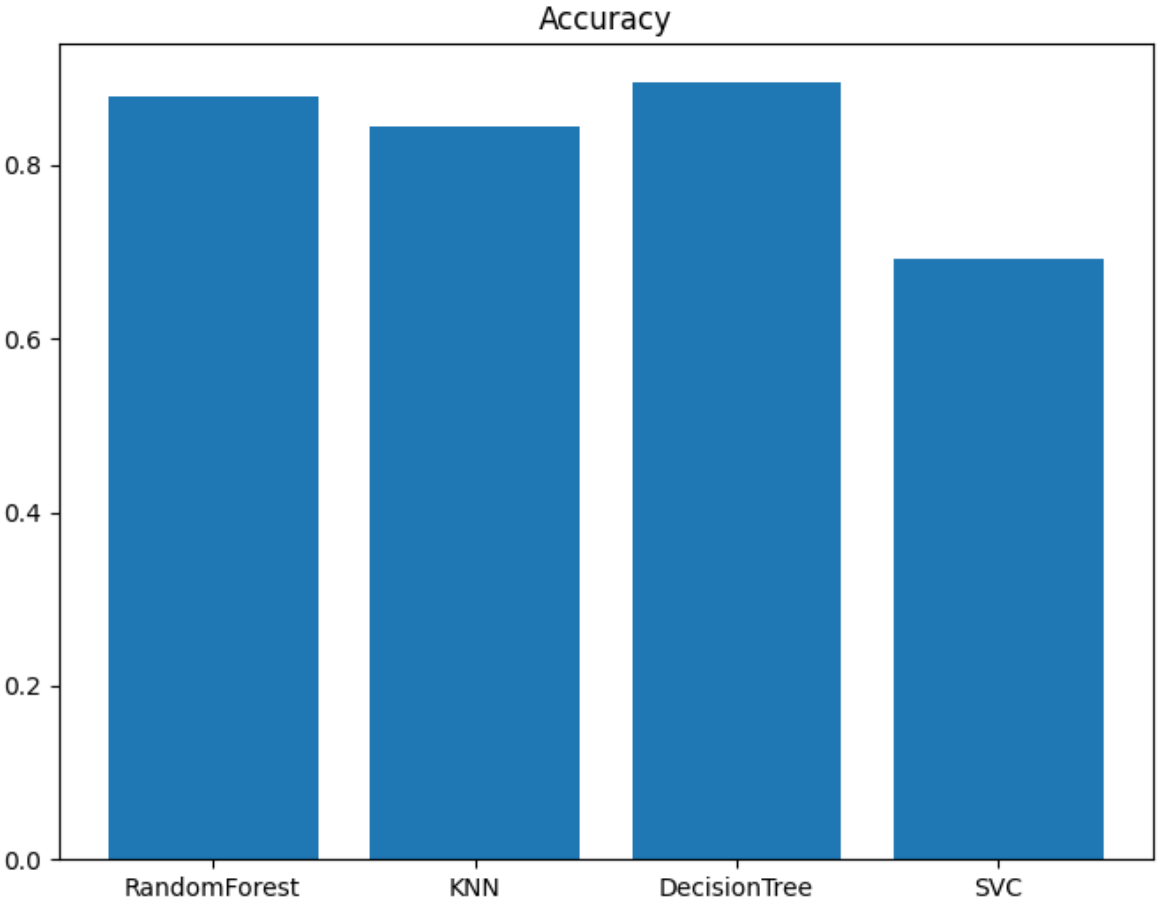
Ho fissato random_state=0 ove possibile perché se esso non viene fissato, il modello, anche con gli stessi dati di input, può produrre risultati diversi ogni volta che viene eseguito. Impostandolo a 0, si garantisce che il modello utilizzi gli stessi numeri casuali in ogni esecuzione, rendendo i risultati riproducibili e consentendo una comparazione accurata tra i diversi modelli e parametri.
Gli iperparametri utilizzati dai vari classificatori, scelti attraverso diverse prove, si è optato per inserire:

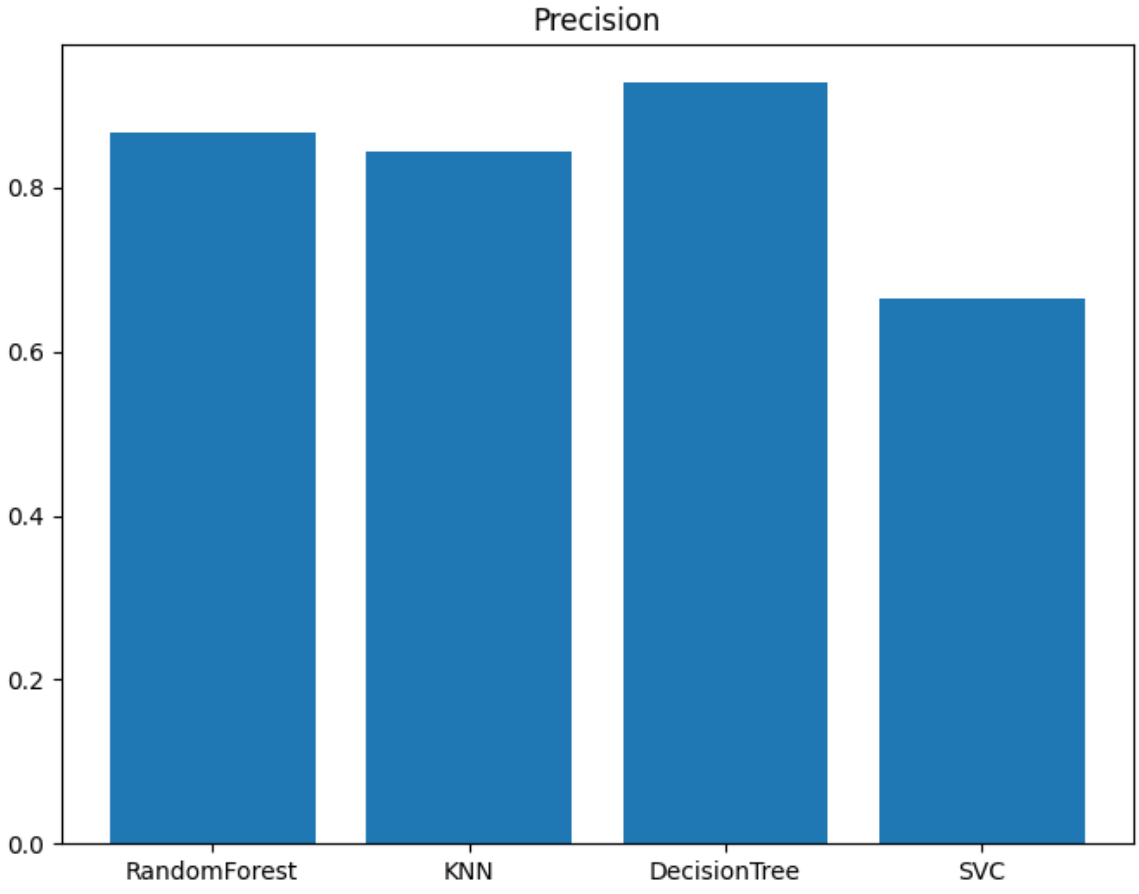
- Per KNN si è optato per dare un peso ai k vicini in base alla distanza, aggiungendo weights="distance". Il metodo per il calcolo della distanza è quello di default. Aggiungendo i pesi alle distanze, il valore delle metriche ha subito un discreto incremento.
- Per RandomForest è stata scelta una profondità non troppo elevata(max_depth=3)in modo da evitare l'overfitting.

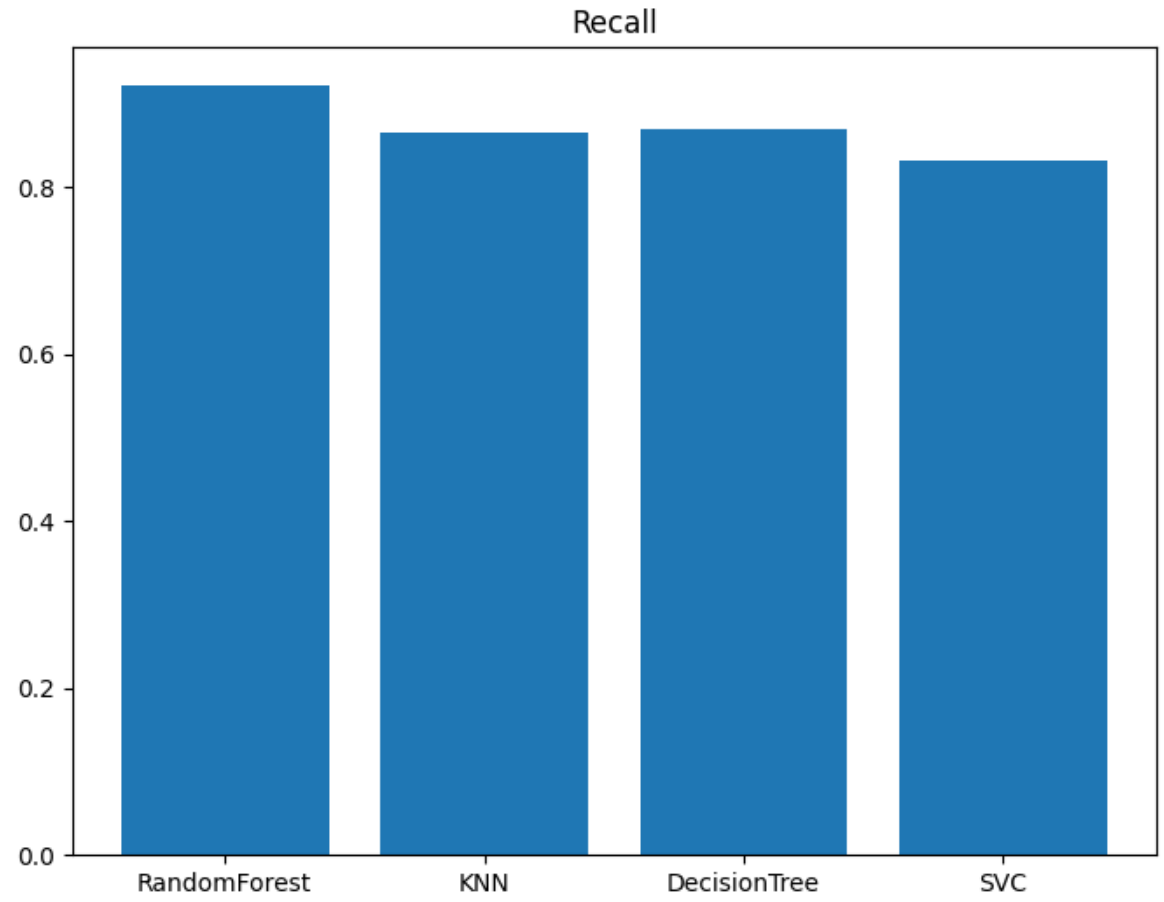
I risultati medi ottenuti dalla valutazione sono i seguenti:

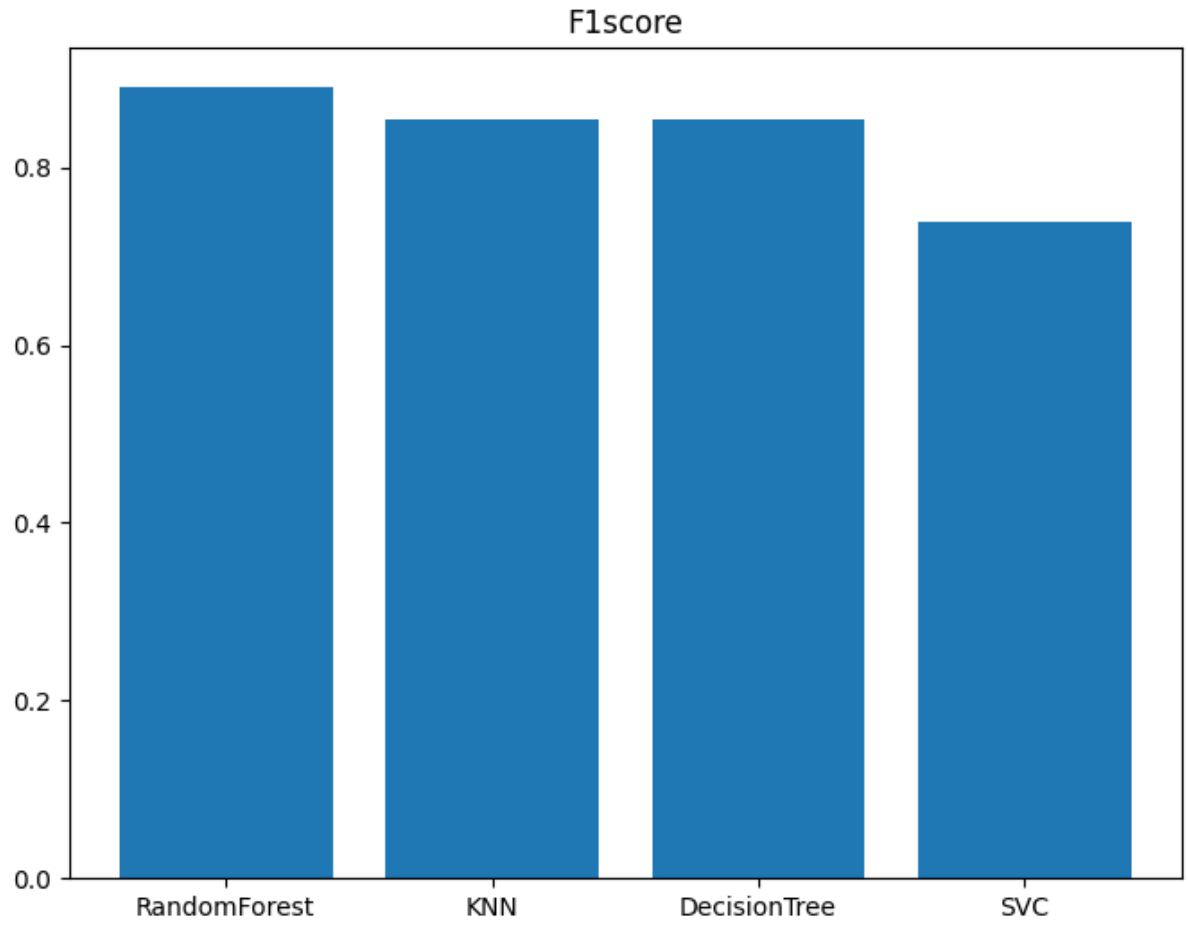
	model	accuracy	precision	recall	f1score
0	RandomForest	0.879679	0.866316	0.921916	0.889814
1	KNN	0.844423	0.844803	0.866548	0.853110
2	DecisionTree	0.894744	0.927836	0.870190	0.853110
3	SVC	0.691603	0.665664	0.832072	0.737851

Entrando nello specifico abbiamo:

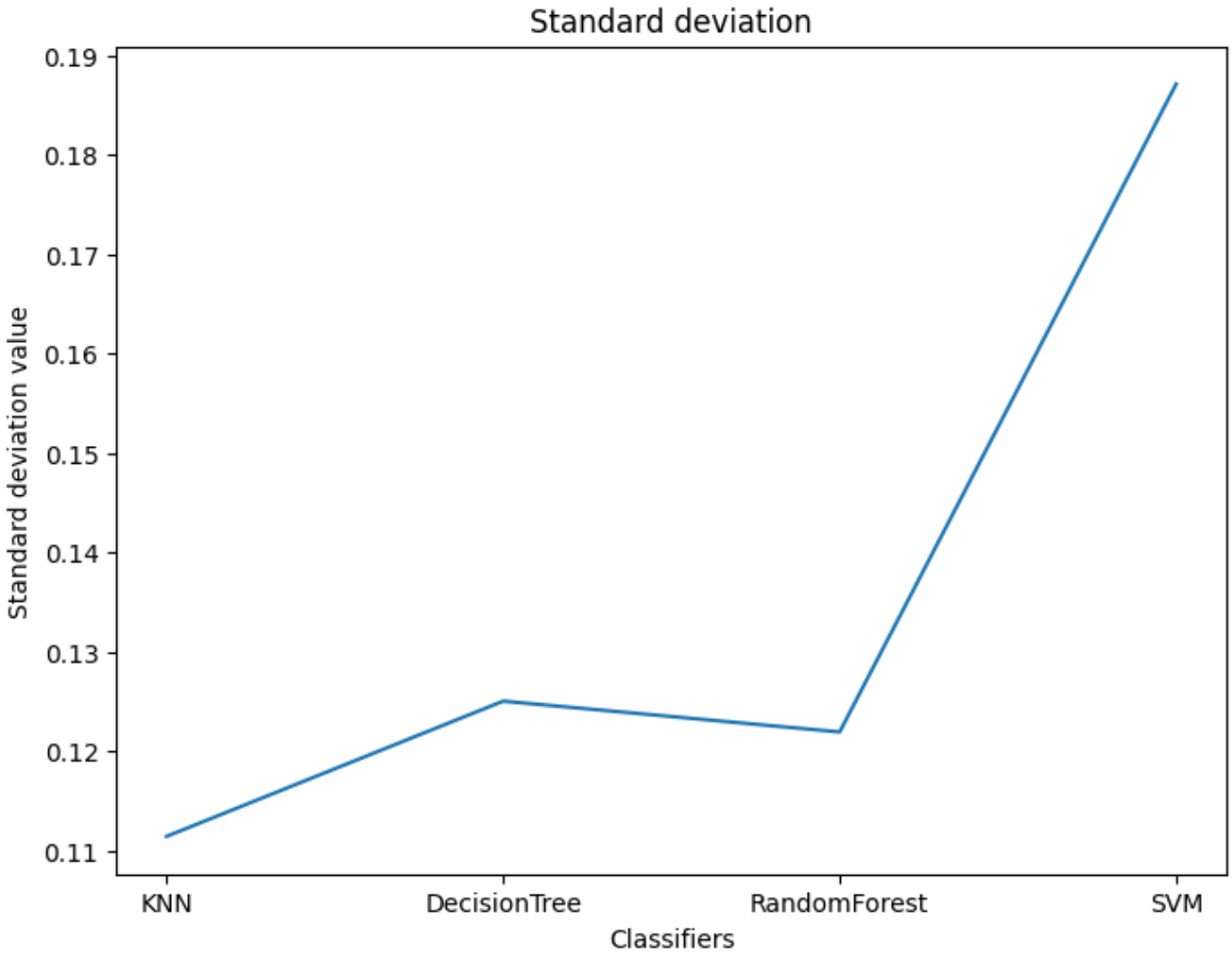





Recall
0.0
0.2
0.4
0.6
0.8
RandomForest
KNN
DecisionTree
SVC


F1score
0.0
0.2
0.4
0.6
0.8
RandomForest
KNN
DecisionTree
SVC

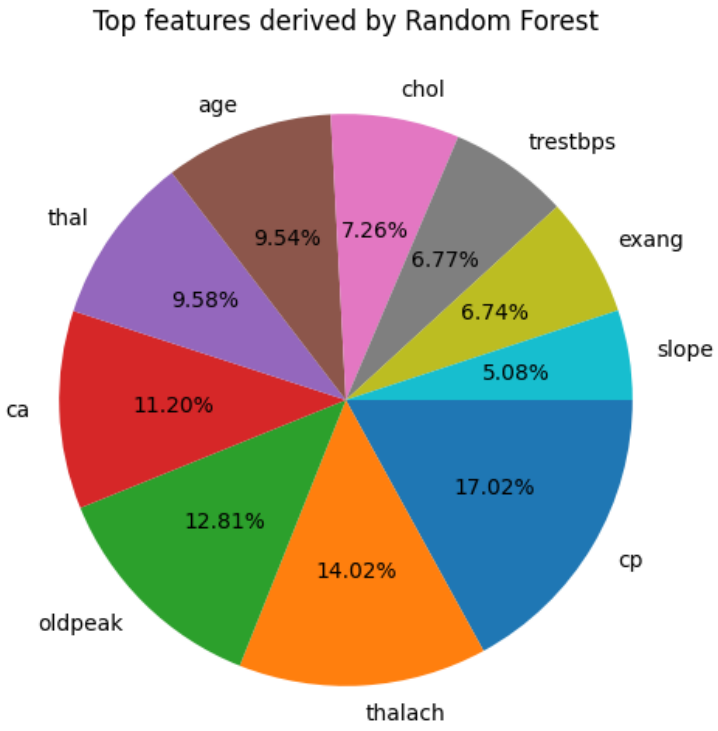
```
Standard deviation for Knn: 0.1114606181023374

Standard deviation for DecisionTree: 0.12505101000008284

Standard deviation for RandomForest: 0.12195265056402768

Standard deviation for SVM: 0.18708286933869706
```



Prendendo in considerazione le performance riportate, soprattutto quella dell'F1-score, ho riscontrato che il classificatore migliore tra quelli riportati è il Random Forest.

Verifica importanza features

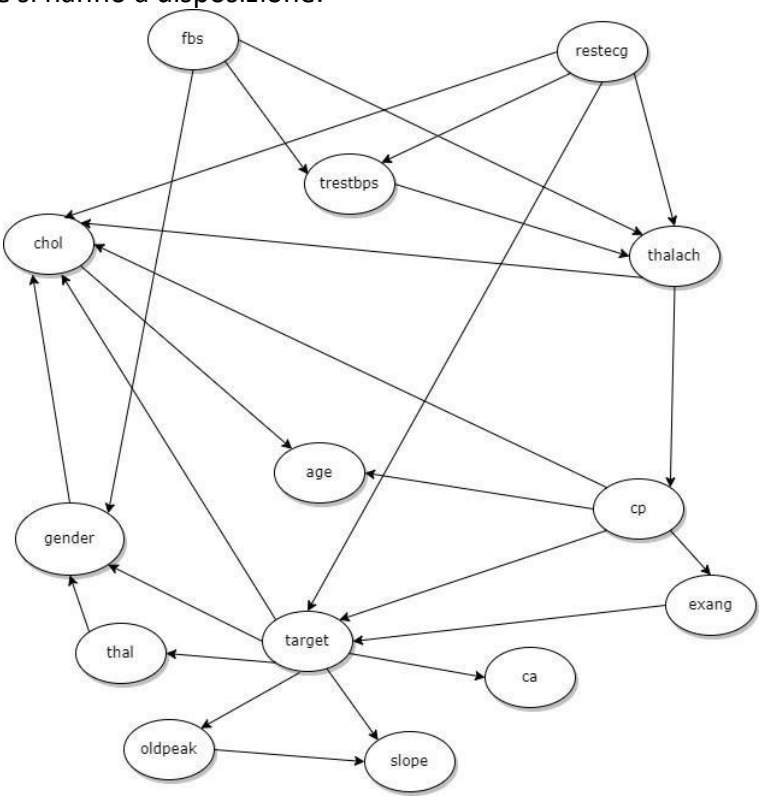
A seguito dell'analisi effettuata precedentemente, ho generato un grafico che estrae le features più importanti derivanti proprio dal Random Forest:



Dal grafico possiamo osservare che il cp(tipologia di dolore al petto) e il thalach(frequenza cardiaca massima raggiunta) sono le caratteristiche mediche predittive più importanti per diagnosticare un individuo potenzialmente malato o meno.

Rete bayesiana

Ho implementato una rete bayesiana per effettuare delle interrogazioni, utilizzando come metodo di scoring il K2score, per verificare le probabilità delle features. Quindi la rete viene creata e modellata sui dati che si hanno a disposizione:



Calcolo probabilità

Calcoliamo le probabilità per le features cp e target grazie alla rete bayesiana precedentemente creata.

Cp: Questa feature influenza diverse features, ragion per cui osserviamo con delle interrogazioni tali influenze. Data una variabile X, solo alcune variabili influenzano direttamente il suo valore. Le variabili che influenzano localmente sono dette Markov Blanket.

Le Markov blanket di cp sono:

```
Markov blanket for "cp"
['restecg', 'gender', 'age', 'chol', 'target', 'exang', 'thalach']
```

Successivamente, grazie alle Markov Blanket, ho effettuato delle query riguardanti la rete con un individuo non malato femminile e maschile.

```
print("Test su soggetto di sesso femminile non malato")
donna = data.query(show_progress=False, variables=['cp'], evidence={'restecg': 0, 'chol': 200,
                                                                    'exang': 0, 'thalach': 170, 'age': 50,
                                                                    'target': 0, 'gender': 0
                                                                    })
print(donna, '\n')
print("Test su soggetto di sesso maschile non malato")
uomo = data.query(show_progress=False, variables=['cp'], evidence={'restecg': 1, 'chol': 300,
                                                                   'exang': 1, 'thalach': 150, 'age': 40,
                                                                   'target': 0, 'gender': 1})
print(uomo, '\n')
```

I risultati ottenuti sono i seguenti:

```
Test su soggetto di sesso femminile non malato
+-------+-----------+
| cp    |   phi(cp) |
+=======+===========+
| cp(0) |    0.8040 |
+-------+-----------+
| cp(1) |    0.1388 |
+-------+-----------+
| cp(2) |    0.0573 |
+-------+-----------+
| cp(3) |    0.0000 |
+-------+-----------+

Test su soggetto di sesso maschile non malato
+-------+-----------+
| cp    |   phi(cp) |
+=======+===========+
| cp(0) |    0.0000 |
+-------+-----------+
| cp(1) |    0.0000 |
+-------+-----------+
| cp(2) |    0.7453 |
+-------+-----------+
| cp(3) |    0.2547 |
+-------+-----------+
```

Lo stesso processo è stato effettuato con un individuo malato femminile e maschile.

```
print("Test su soggetto di sesso femminile malato")
donna = data.query(show_progress=False, variables=['cp'], evidence={'restecg': 0, 'chol': 200,
                                                                    'exang': 0, 'thalach': 170, 'age': 50,
                                                                    'target': 1, 'gender': 0
                                                                    })
print(donna, '\n')
print("Test su soggetto di sesso maschile malato")
uomo = data.query(show_progress=False, variables=['cp'], evidence={'restecg': 1, 'chol': 300,
                                                                   'exang': 1, 'thalach': 150, 'age': 40,
                                                                   'target': 1, 'gender': 1})
print(uomo, '\n')
```

I risultati sono i seguenti:

```
Test su soggetto di sesso femminile malato
+-------+-----------+
| cp    |   phi(cp) |
+=======+===========+
| cp(0) |    0.5586 |
+-------+-----------+
| cp(1) |    0.2254 |
+-------+-----------+
| cp(2) |    0.2160 |
+-------+-----------+
| cp(3) |    0.0000 |
+-------+-----------+

Test su soggetto di sesso maschile malato
+-------+-----------+
| cp    |   phi(cp) |
+=======+===========+
| cp(0) |    0.0000 |
+-------+-----------+
| cp(1) |    0.0000 |
+-------+-----------+
| cp(2) |    0.0000 |
+-------+-----------+
| cp(3) |    1.0000 |
+-------+-----------+
```

Target: Le markov blanket di target sono:

```
Markov blanket for "target"
['restecg', 'thalach', 'ca', 'slope', 'gender', 'cp', 'chol', 'fbs', 'thal', 'oldpeak', 'exang']
```

Successivamente, grazie alle Markov Blanket, ho effettuato delle query riguardanti la rete con un individuo non malato e malato e calcola la probabilità di esserlo o no. Queste query sono state effettuate su individui maschili.

```
nonMalati = data.query(show_progress=False, variables=['target'],
                       evidence={'restecg': 1, 'fbs': 0, 'oldpeak': 3, 'chol': 250, 'exang': 0 , 'thal': 2, 'ca': 3, 'thalach': 150, 'cp': 0, 'slope': 0, 'gender':1})

print('\nProbabilità per un potenziale non malato:')
print(nonMalati, '\n')

malato = data.query(show_progress=False, variables=['target'],
                    evidence={'restecg': 1, 'fbs': 1, 'oldpeak': 1, 'chol': 300, 'exang': 1 , 'thal': 1, 'ca': 0, 'thalach': 150, 'cp': 0, 'slope': 2, 'gender':1})
print('\nProbabilità per un potenziale malato:')
print(malato)
```

I risultati sono stati i seguenti:

```
Probabilità per un potenziale non malato:
+----------+--------------+
| target   |   phi(target) |
+==========+==============+
| target(0) |        0.9468 |
+----------+--------------+
| target(1) |        0.0532 |
+----------+--------------+


Probabilità per un potenziale malato:
+----------+--------------+
| target   |   phi(target) |
+==========+==============+
| target(0) |        0.6345 |
+----------+--------------+
| target(1) |        0.3655 |
+----------+--------------+
```

Per verificare l'effettiva rilevanza delle features all'interno della rete bayesiana, ho effettuato dei test sugli individui presi in considerazione.

```
testNonMalato = data.query(show_progress=False, variables=['target'],
                           evidence={'restecg': 1, 'fbs': 0, 'oldpeak': 1, 'chol': 200, 'exang': 1 , 'thal': 2, 'ca': 0, 'thalach': 150, 'cp': 0, 'slope': 2, 'gender':1})

print('\nTest su soggetto potenzialmente non malato:')
print(testNonMalato, '\n')

testMalato = data.query(show_progress=False, variables=['target'],
                        evidence={'restecg': 1, 'fbs': 1, 'oldpeak': 3, 'chol': 200, 'exang': 1 , 'thal': 1, 'ca': 1, 'thalach': 150, 'cp': 3, 'slope': 0, 'gender':1})

print('\nTest su soggetto potenzialmente malato:')
print(testMalato, '\n')
```

I risultati sono stati i seguenti:

```
Test su soggetto potenzialmente non malato:
+----------+--------------+
| target   |   phi(target) |
+==========+==============+
| target(0) |        0.3530 |
+----------+--------------+
| target(1) |        0.6470 |
+----------+--------------+


Test su soggetto potenzialmente malato:
+----------+--------------+
| target   |   phi(target) |
+==========+==============+
| target(0) |        0.8983 |
+----------+--------------+
| target(1) |        0.1017 |
+----------+--------------+
```

Interazione con l'utente

La Knowledge Base (KB) viene impiegata al fine di rispondere a interrogazioni probabilistiche poste dall'utente e basate su evidenze. L'utente seleziona una caratteristica dal dataset da considerare come ipotesi e, successivamente, specifica l'evidenza mediante la descrizione delle variabili e dei loro corrispondenti valori.

```
Vorresti effettuare una predizione?(si/no)
 > si
Seleziona una o più variabili (in caso si vogliano scrivere più di una variabile, separale con uno spazio):
 | age |
 ---------
 | gender |
 ---------
 | cp |
 ---------
 | trestbps |
 ---------
 | chol |
 ---------
 | fbs |
 ---------
 | restecg |
 ---------
 | thalach |
 ---------
 | exang |
 ---------
 | oldpeak |
 ---------
 | slope |
 ---------
 | ca |
 ---------
 | thal |
 ---------
 | target |
 ---------
```

Un esempio di interrogazione è questa:
P(target | gender:1 cp:2 thalach:155 thal:0), le probabilità che un individuo con queste caratteristiche abbia una patologia cardiaca sono:

```
 > gender:1 cp:2 thalach:155 thal:0
+-----------+---------------+
| target    |   phi(target) |
+===========+===============+
| target(0) |        0.5043 |
+-----------+---------------+
| target(1) |        0.4957 |
+-----------+---------------+
```