Università degli Studi di Milano - Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di laurea Magistrale in Informatica

Relazione Progetto: Machine Learning

Relazione del progetto di:
Stoianov Oleg (829519)
Gusmara Andrea (831141)
Villani Alessio (830075)

Anno Accademico 2020-2021

INDICE

CAPITOLO 1
DESCRIZIONE DEL DOMINIO

Il progetto del corso di Machine Learning si pone come obiettivo quello di condurre un'esperimento di apprendimento automatico basato sulle tecniche approfondite durante le lezioni. Lo scopo di questa relazione è di analizzare una serie di dati mediche di alcune pazienti donne di un ospedale indiano. Questo insieme di variabili mediche possono essere usate per predire la presenza della malattia del Diabete.

1.1 Analisi DataSet

Il Dataset che andremo ad utilizzare per questa ricerca è il Pima Indians Diabetes Database pubblicato su Kaggle dall'National Institute of Diabetes and Digestive and Kidney Diseases . Il dataset consiste in diverse variabili raccolte da studi medici . Le variabili in questione sono :

- Pregnancies : Numero di gravidanze del soggetto
- Gloucose : Concentrazione di glucosio nel plasma del soggetto a due ore dal test orale di tolleranza al glucosio
- BloodPressure: Pressione diastolica del sangue (mm/Hg)
- SkinThickness : Spessore della pelle del tricipite (mm)

- Insulin : Valore dell'insulina nel sangue (muU/ml) (micro unità di insulina per ml di sangue)
- BMI : Indice di massa corporea (peso in $kg/(heightinm)^2$)
- DiabetesPedigreeFunction : valore associato all'eridatarietà dei geitori
- Age : Età (anni)
- Outcome : Variabile di classificazione (0 o 1): 0 classificato negativo 1 altrimenti .

1.1.1 Analasi delle covariate

Considerando il dataset originale è inoltre possibile effettuare descrizione statistica di ogni singola feature rappresentando i valori dei quantili e dei loro minimi e massimi.

1.1.2 PCA come analisi esplorativa del dataset

La Principal component analisis è una procedura in grado di diminuire le dimensioni del nostro dataset , per dataset molto grandi mantendo comunque un numero di informazione elevato. La PCA è inoltre utilizzata per effettuare un'analisi del dataset .

1.1.3 Pulizia del Dataset

Il Dataset presenta però valori mancanti in alcuni soggetti , dovuti probabilmente ad analisi non complete , che in caso utilizzati nella predizione porterebbero a ottenere dei risultati non ottimali . Dopo diverse ricerca sulla letteratura disponibile online abbiamo deciso di seguire procedura di eliminazione dei record appartenenti dei soggetti a cui risultava mancante almeno uno dei valori proposti dallo studio , a meno della variabile Pregnancies. Con questa selezione dei record siamo però passati da avere una totalità di 768 (di cui 268 positivi e gli altri 500 negativi) a 392 (di cui 130 positivi e gli altri 262 negativi) , riducendo cosi la dimensione del dataset ,e mantenendo una proporzione tra le classi due classi.

Cercare di intabellare le varie immagini ottenute

Questo grafico indica la percentuale di positività per soggetti che condividono lo stesso valore glucosio nel sangue.

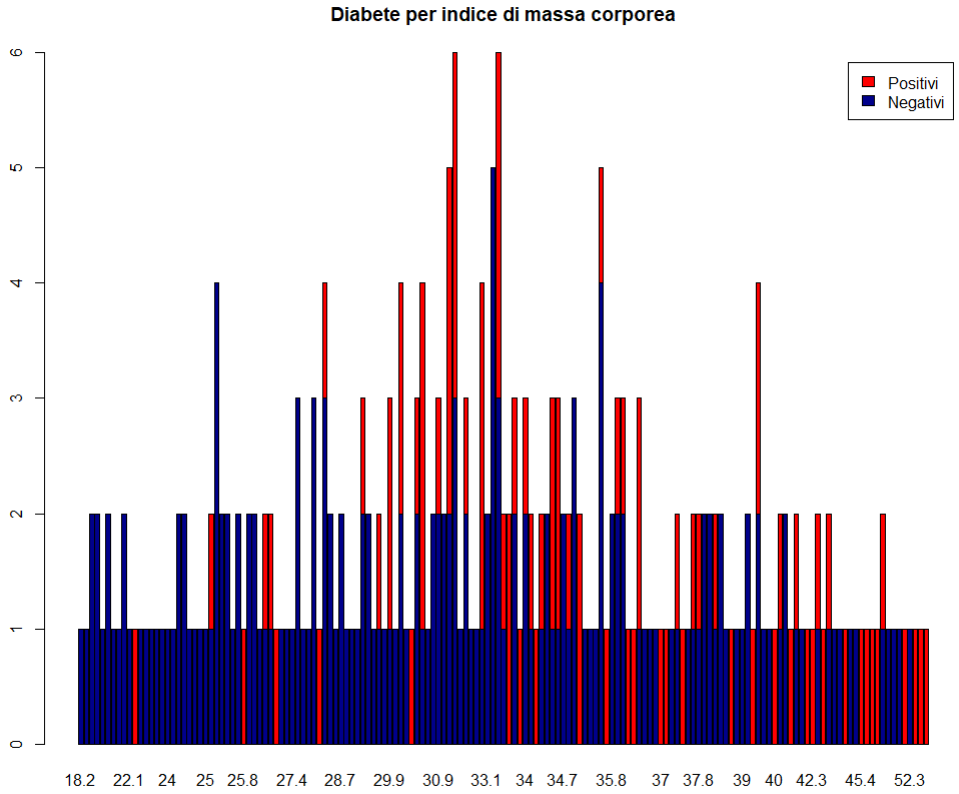

Figura 1.1:

1.2 Prima partizione Dataset

Dopo avere effettuato una pulizia iniziale del dataset andiamo ad effettuare una prima divisione su di esso. Sostanzialmente andiamo a suddividerlo in testset e in trainset con una proporzione di 70/30. Questo procedimento con la funzione sample che crea degli indici casuali con la percentuale 70 30, con cui andremo ad assegnare con il record del nostro dataset.

DIVISIONE INIZIALE DEL DATASET E DEL TRAINING SET set.seed(1000) ind = sample(2, nrow(dataset), replace = TRUE, prob=c(0.7, 0.3)) trainset = dataset[ind == 1,]

Questa divisione permetterà di utilizzare il trainset insieme alla 10-folds cross validation per andare ad allenare i nostri modelli , e invece il testset

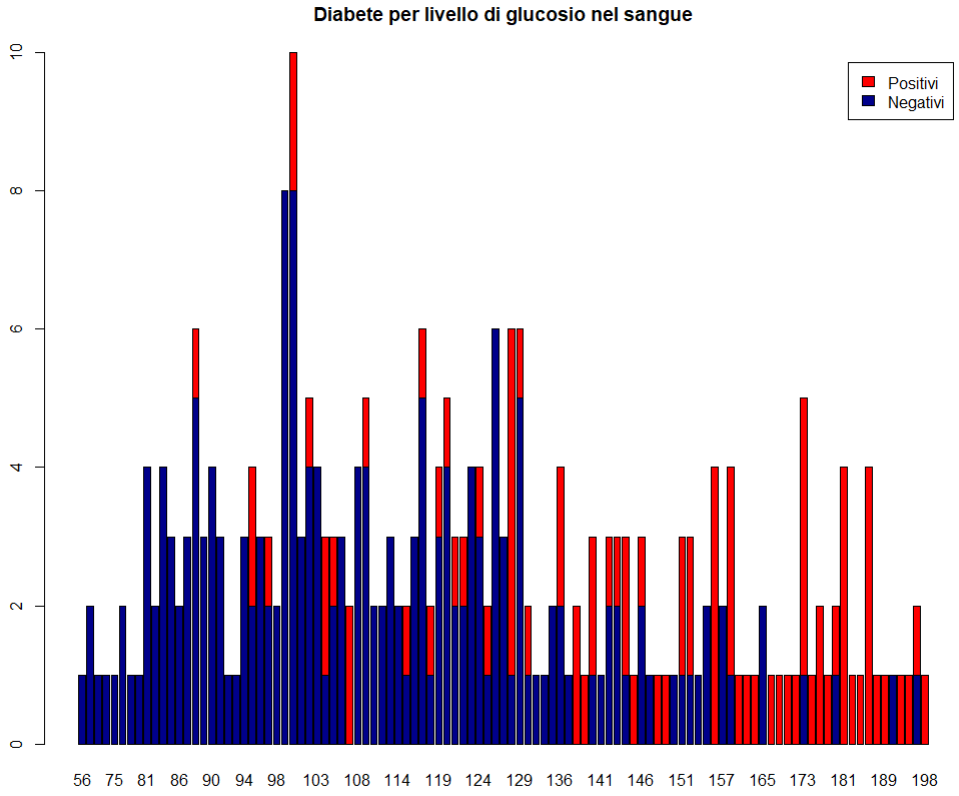

Figura 1.2:

sarà usato appunto come metodo di validazione dei nostri modelli , in modo tale che il modello al momento della sua valutazione abbia visto per la prima volta quell'insieme de dati .

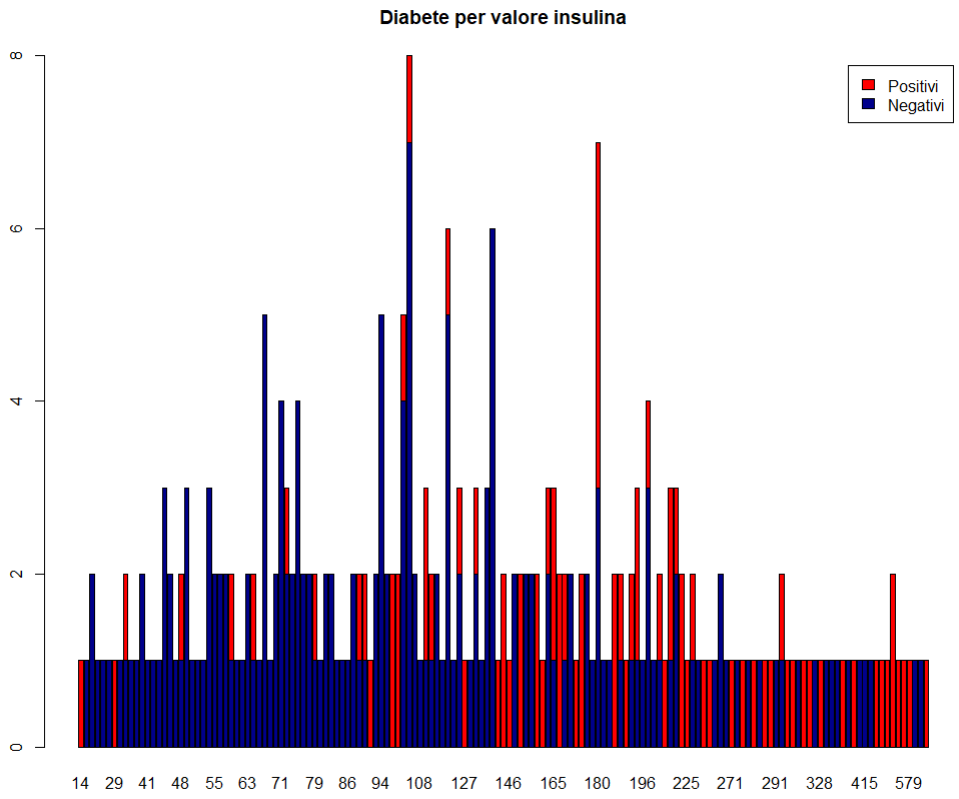
Diabete per valore insulina
Positivi
Negativi
0
2
4
6
8
14
29
41
48
55
63
71
79
86
94
108
127
146
165
180
196
225
271
291
328
415
579

Figura 1.3:

CAPITOLO 2

ESPERIMENTI

2.1 Cross-Validation

Per tutte la fasi di allenamento di ogni modello andremo ad applicare la tecnica di 10 folds cross validation . Questa tecnica cosiste nel andare suddividere il nostro dataset (che rappresenta il trainset della prima divisione) in 10 parti uguali chiamati folds e associare uno di questi subset al testset e gli altri restanti al trainset .Questa procedura viene ripetuta per l'appunto 10 volte andando a cmaniare ogni volta il subset associato al test in modo tale da considerarli tutti. Il metodo che crea un controllo della cross-validation lo vediamo a seguito TRAINING create the traincontrol parameter control = trainControl(method = "repeatedcv", number = 10,repeats = 10, classProbs = TRUE, summaryFunction = twoClassSummary)

2.2 Support Vector Machine

Come prima tecnica utilizziamo le support vector machine . Andiamo a valutare le dverse funzioni kernel per allenare la nostre support vector.

SVM con kernel radial : svm.model
SVM con kernel lineare : svm.linear.model
SVM con kernel polinomiale : svm.poly.model

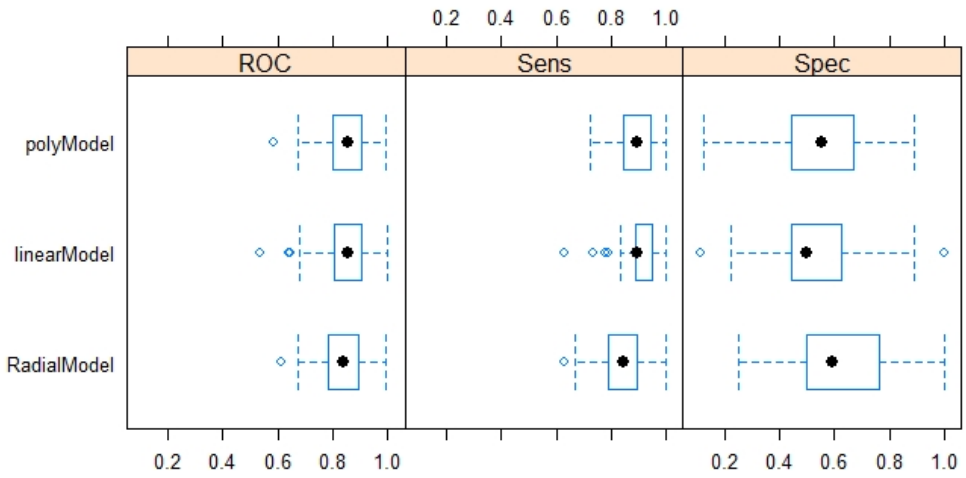

Figura 2.1:

Per poi confrontarle tra di loro

Commentare questo grafico e spigare qualcosa sul boxplot e sul l'altro plot sotto.

2.3 Decision tree

Come seconda tecnica utilizziamo l'albero di decisione . L'albero viene allenato con la funzione train sul trainset .

Di seguito vediamo la rappresentazione dell'albero.

Qua invece i risultati delle sue matrici di confusione.

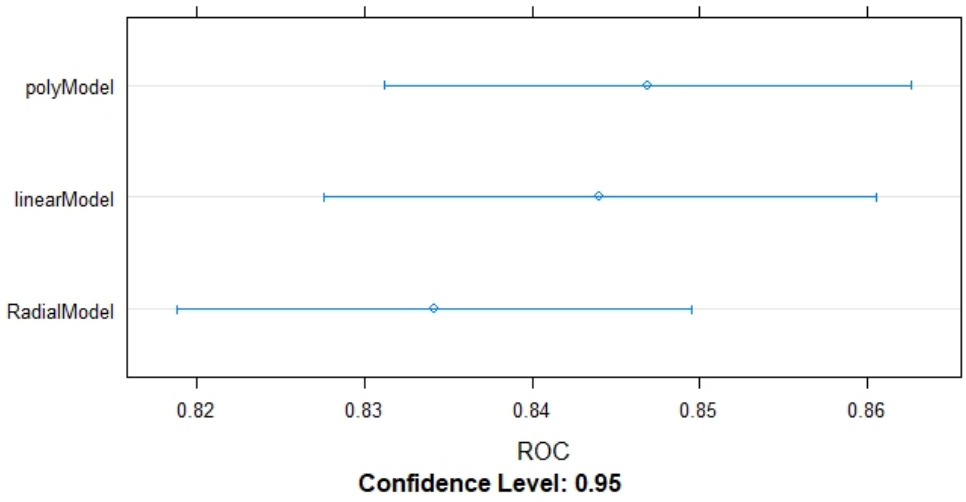

Figura 2.2:

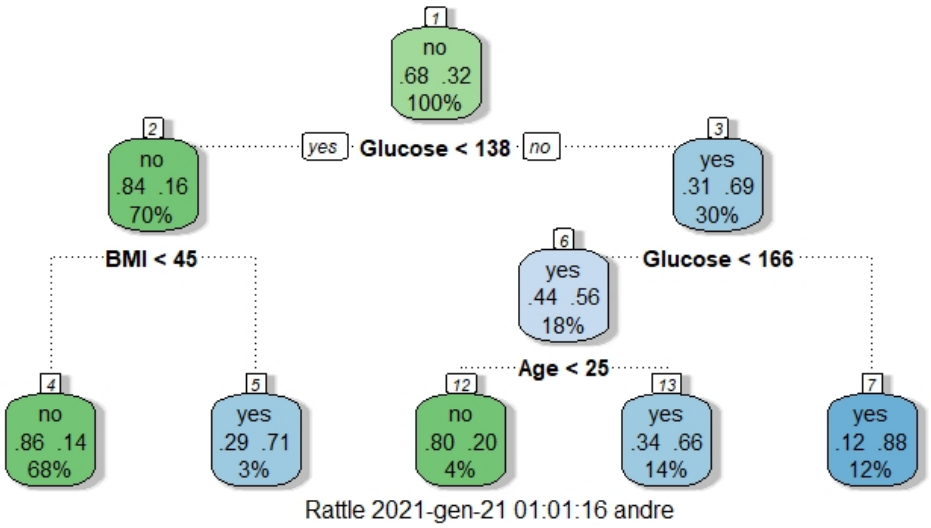

Figura 2.3:

```
> result.rpart.class1
Confusion Matrix and Statistics

          Reference
Prediction no yes
       no  66  22
       yes 12  21

               Accuracy : 0.719
                 95% CI : (0.6301, 0.7969)
    No Information Rate : 0.6446
    P-Value [Acc > NIR] : 0.05135

                  Kappa : 0.3529

 Mcnemar's Test P-Value : 0.12271

            Sensitivity : 0.4884
            Specificity : 0.8462
         Pos Pred Value : 0.6364
         Neg Pred Value : 0.7500
              Precision : 0.6364
                 Recall : 0.4884
                     F1 : 0.5526
             Prevalence : 0.3554
         Detection Rate : 0.1736
   Detection Prevalence : 0.2727
      Balanced Accuracy : 0.6673

       'Positive' Class : yes

> |
```

Figura 2.4:

```
> result.rpart.class2
Confusion Matrix and Statistics

          Reference
Prediction no yes
       no  66  22
       yes 12  21

               Accuracy : 0.719
                 95% CI : (0.6301, 0.7969)
    No Information Rate : 0.6446
    P-Value [Acc > NIR] : 0.05135

                  Kappa : 0.3529

 Mcnemar's Test P-Value : 0.12271

            Sensitivity : 0.8462
            Specificity : 0.4884
         Pos Pred Value : 0.7500
         Neg Pred Value : 0.6364
              Precision : 0.7500
                 Recall : 0.8462
                     F1 : 0.7952
             Prevalence : 0.6446
         Detection Rate : 0.5455
   Detection Prevalence : 0.7273
      Balanced Accuracy : 0.6673

       'Positive' Class : no
```

Figura 2.5:

CAPITOLO 3

RISULTATI

3.1 Valutazione SVM

3.2 Valutazione Decision Tree

3.3 Confronto tra i due modelli