Modelli di classificazione per la diagnosi del Diabete

Alessio Splendiani

September 2024

1 Capitolo 1

1.1 Introduzione

Stiamo esaminando il dataset ***Diabetes Data-Set***[1] con l'obiettivo di applicare diversi modelli di classificazione, tra cui la regressione logistica, l'analisi discriminante, il Support Vector Machine (SVM) e il K-Nearest Neighbors (KNN), per analizzare la variabile "Outcome".

I risultati del modello potrebbero essere molto utili per capire in base a quali caratteristiche una persona possa essere classificata come persona con Diabete rispetto ad una senza Diabete. Il DataSet comprende 768 osservazioni per 9 colonne.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0
7	3	78	50	32	88	31.0	0.248	26	1
8	10	115	0	0	0	35.3	0.134	29	0
9	2	197	70	45	543	30.5	0.158	53	1
10	8	125	96	0	0	0.0	0.232	54	1
11	4	110	92	0	0	37.6	0.191	30	0
12	10	168	74	0	0	38.0	0.537	34	1
13	10	139	80	0	0	27.1	1.441	57	0
14	1	189	60	23	846	30.1	0.398	59	1
15	5	166	72	19	175	25.8	0.587	51	1
16	7	100	0	0	0	30.0	0.484	32	1

Figure 1: Diabetes Data

Il nostro obiettivo sarà quello di classificare correttamente i valori della variabile dipendente Outcome sulla base dei valori assunti dalle seguenti variabili indipendenti:

- **Pregnancies**: indica il numero di gravidanze;
- **Glucose**: esprime il livello di glucosio nel sangue;
- **BloodPresure**: esprime la misurazione della pressione sanguigna;
- **SkinThickness**: esprime lo spessore della pelle;
- **Insulina**: indica il livello di insulina nel sangue;
- **BMI**: indica l'indice di massa corporea;
- **DiatabesPedigreeFunction**: calcola la probabilità di diabete in base all'età del soggetto e alla sua storia familiare diabetica.

[1] Puoi accedere al dataset ***Diabetes Data-Set*** tramite il seguente link: `https://www.kaggle.com/datasets/pritsheta/diabetes-dataset`.

La prima cosa che faremo nel modello è andare a verificare la presenza di valori nulli e se presenti li andremo ad eliminare.

```
vis_miss(data)
data <-na.omit(data)
```

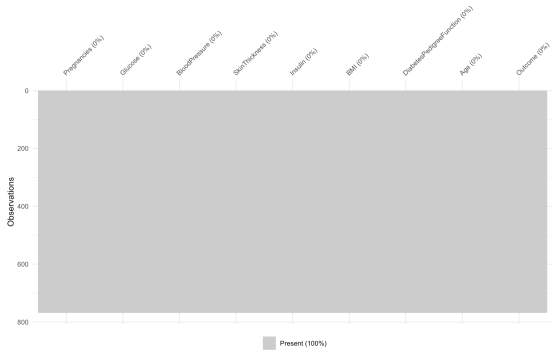


Figure 2: Valori Nulli

Possiamo osservare che nel nostro modello non sono presenti valori nulli.

1.2 Analisi Esplorativa

Successivamente vado a fare un'analisi esplorativa del Dataset, evidenziando informazioni che possono essere interessanti da osservare. Con il comando corrplot vado a visualizzare la presenza di possibili correlazioni tra le variabili.

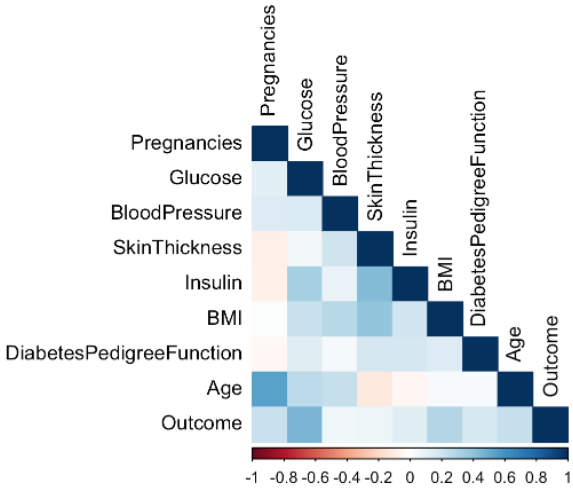


Figure 3: Corrplot

Questa matrice suggerisce che Glucose, BMI, Age e Pregnancies sono le variabili più importanti per prevedere il diabete in questo dataset. Tuttavia, è importante notare che nessuna correlazione è estremamente forte, indicando che il diabete è una condizione complessa influenzata da molteplici fattori.

Sono poi andato a creare un grafico a torta per vedere la percentuale di persone classificate come "**Diabetico**" e quelle classificate con "**Non diabetico**".

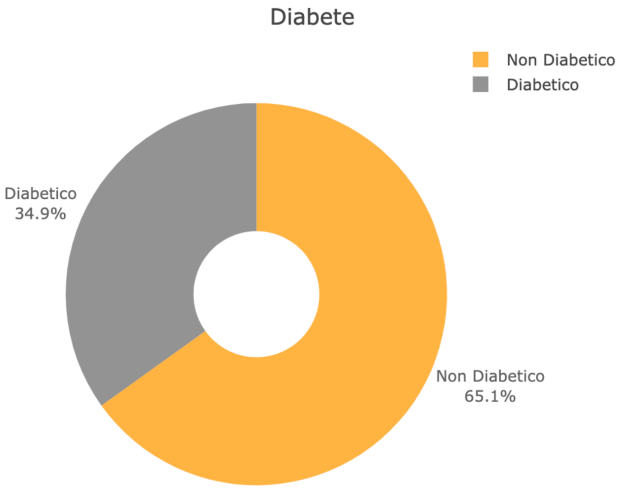


Figure 4: Grafico a torta

Il 65.1 % delle persone del nostro DataSet sono classificate come diabetiche ed il 34.9% come non diabetiche.

Per ogni variabile sono poi andato a mostrare la distribuzione dei dati con il comando **summary** ed il comando **boxplot**.

```
> summary(data$Pregnancies)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   3.000   3.845   6.000  17.000
```

Figure 5: Summary Pregnancies

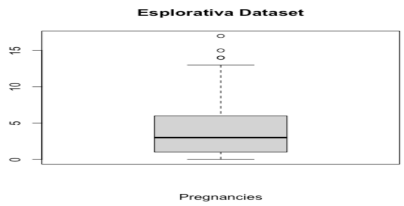


Figure 6: BoxPlot Pregnancies

Il valore di **Pregnancies** va da un minimo di 0 ad un massimo di 17, il valore della media è 3.845 e quello della mediana di 3.0.

```
> summary(data$Glucose)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    99.0   117.0   120.9   140.2   199.0
```

Figure 7: Summary Glucose

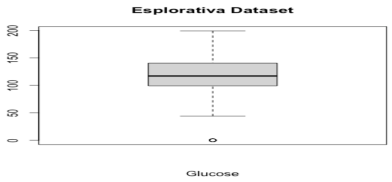


Figure 8: BoxPlot Glucose

Il valore di **Glucose** va da un minimo di 0 ed un massimo di 199, il valore della media è di 120.9 e quello della mediana di 117.

```
> summary(data$BloodPressure)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   62.00   72.00   69.11   80.00  122.00
```

Figure 9: Summary BloodPressure

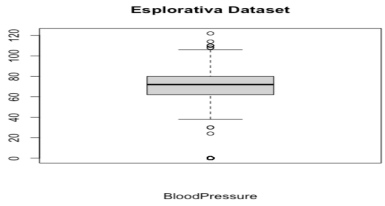


Figure 10: BoxPlot BloodPressure

Il valore di **BloodPressure** va da un minimo di 0 ad un massimo di 122, il valore della media è di 69.11 e quello della mediana di 72.

```
> summary(data$SkinThickness)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00   23.00   20.54   32.00   99.00
```

Figure 11: Summary SkinThickness

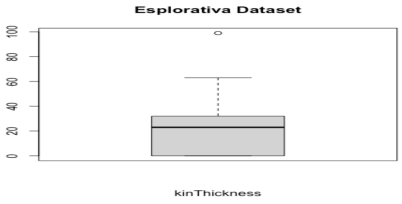


Figure 12: BoxPlot SkinThickness

Il valore di **SkinThickness** va da un minimo di 0 ad un massimo di 99, il valore della media è di 20.54 e quello della mediana di 23.

```
> summary(data$Insulin)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0     0.0    30.5    79.8   127.2   846.0
```

Figure 13: Summary Insulina

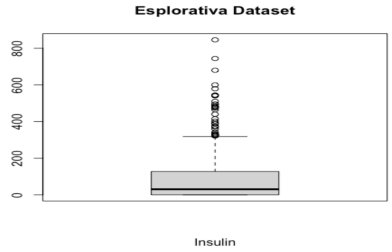


Figure 14: BoxPlot Insulina

Il valore di **Insulin** va da un minimo di 0 ed un massimo di 846, il valore della media è di 79.7 e quello della mediana di 30.5.

```
> summary(data$BMI)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   27.30   32.00   31.99   36.60   67.10
```

Figure 15: Summary BMI

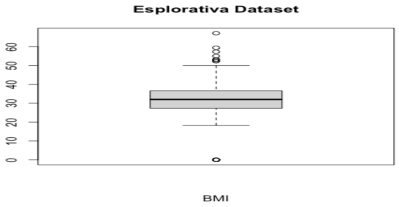


Figure 16: BoxPlot BMI

Il valore di BMI va da un minimo di 0 ad un massimo di 67.10, il valore della media è di 31.99 e quello della mediana di 32.

```
> summary(data$DiabetesPedigreeFunction)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0780  0.2437  0.3725  0.4719  0.6262  2.4200
```

Figure 17: Summary DiabetesPedigreeFunctions

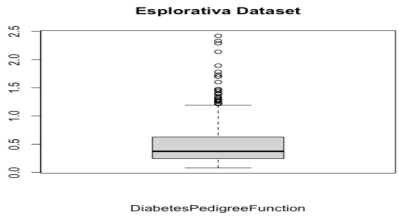


Figure 18: BoxPlot DiabetesPedigreeFunctions

Il valore di **DiabetesPedigreeFunction** va da un minimo di 0 ad un massimo di 2.42, il valore della media è di 0.472 e quello della mediana di 0.37.

```
> summary(data$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  21.00   24.00   29.00   33.24   41.00   81.00
```

Figure 19: Summary Età

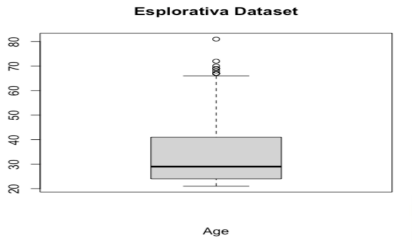


Figure 20: BoxPlot Età

Il valore di **Age** va da un minimo di 21 ad un massimo di 81, il valore della media è di 33.24 e quello della mediana di 29.

2 Modelli di Classificazione

Gli algoritmi di classificazione che abbiamo utilizzato sul nostro dataset sono i seguenti:

- **Regressione logistica**: È un modello statistico per la classificazione binaria che utilizza una funzione per modellare la probabilità di appartenenza a una classe. Stima la probabilità che un'osservazione appartenga a una classe e usa una soglia (tipicamente 0.5) per fare la previsione finale.
- **Analisi discriminante lineare**: È un metodo di classificazione che cerca di trovare una combinazione lineare di caratteristiche che separa al meglio le due classi. Assume che le variabili siano normalmente distribuite e abbia una matrice di covarianza comune per tutte le classi.
- **SVM Lineare**: È una variazione del Support Vector Machine (SVM) che utilizza un iperpiano lineare per separare le classi. Mira a massimizzare il margine tra le classi,
- **SVM Radiale**: È un tipo di Support Vector Machine che utilizza il kernel radiale (RBF) per mappare i dati in uno spazio ad alta dimensione dove le classi possono essere separate non linearmente. Permette la classificazione di dati che non sono linearmente separabili nel loro spazio originale.
- **KNN:** È un metodo di classificazione basato sulla vicinanza tra punti di dati. Classifica un'osservazione in base alla maggioranza dei voti dei suoi k vicini più prossimi.

Andiamo a dividere il dataset in **training set** e **test set**. Nel primo andrò ad addestrare i miei modelli e nel secondo andrò a testare quanto le previsioni siano corrette. Ho deciso di dividere i miei dati in questo modo:

- 75 % training set;
- 25 % test set;

```
#Divido il mio dataset in training e test set
training.samples = createDataPartition(data$Outcome, p = 0.75, list
    = FALSE)
train <- data[training.samples, ]
test <- data[-training.samples, ]

#Metodo di cross validation
control <- trainControl(method="repeatedcv", number=10, repeats=3)
```

Il comando control esegue una cross-validation con 10 fold, ripetuta 3 volte, per valutare la performance del modello su diverse suddivisioni dei dati.

2.1 Training set

Successivamente vado ad addestrare tutti i miei modelli.

```
# ADDESTRO I DIVERSI MODELLI
# Regressione logistica
fit_logistica <- train(Outcome ~ ., data = train, method = "glm",
                       family = "binomial", trControl = control)

# LDA
fit_lda <- train(Outcome ~ ., data = train, method = "lda",
    trControl = control)

# SVM Lineare
fit_svm_linear <- train(Outcome ~ ., data = train, method = "
    svmLinear",
                        trControl = control)

# SVM Radiale
fit_svm_radial <- train(Outcome ~ ., data = train, method = "
    svmRadial",
                        trControl = control)

# KNN
fit_knn <- train(Outcome ~ ., data = train, method = "knn",
                 trControl = control)
```

Per valutare le prestazioni dei modelli di classificazione è importante analizzare non solo i risultati ottenuti sul test set, ma anche le prestazioni sui dati di addestramento. In particolare sono andato ad osservare ***l'accuracy*** (che mi dice la % dei soggetti correttamente classificati)

	Model	Accuracy
1	Logistica	0.7916667
2	LDA	0.7916667
3	SVM Radial	0.8229167
4	SVM Lineare	0.7795139
5	KNN	0.7829861

Figure 21: Accuracy training

Il **SVM Radiale** si distingue come il modello con la migliore accuratezza sul training set, suggerendo una buona capacità di adattamento ai dati di addestramento.

2.2 Test set

Successivamente dopo aver addestrato i miei modelli, sono andato ad applicare gli algoritmi al test set.

```
#Dopo aver addestrato i modelli procedo con la previsione sul test
    set
pred_logistica <- predict(fit_logistica, test)

#lda
pred_lda <- predict(fit_lda,test)

#svm
pred_svm_radial <- predict(fit_svm_radial,test)
pred_svm_linear <- predict(fit_svm_linear,test)

#KNN
pred_knn <- predict(fit_knn, test)
```

Sono poi andato a valutare le prestazioni dei miei modelli sfruttando sempre l'***accuracy*** vista prima, e la ***sensitivity*** (mi dice l'abilità del modello a riconoscere i positivi, cioè i soggetti con label = 1).

Ho prima costruito le matrici di confusione uno strumento di valutazione per modelli di classificazione che mostra il numero di previsioni corrette e incorrette suddivise per ciascuna classe. Consiste in una tabella con le seguenti informazioni:

- **Veri Positivi (VP):** Numero di osservazioni correttamente classificate come positive.

- **Falsi Positivi (FP):** Numero di osservazioni erroneamente classificate come positive.
- **Veri Negativi (VN):** Numero di osservazioni correttamente classificate come negative.
- **Falsi Negativi (FN):** Numero di osservazioni erroneamente classificate come negative.

Successivamente ho ricavato **l'accuracy** e la **sensitivity**.

```
        Model  Accuracy Sensitivity
1   Logistica 0.7447917       0.848
2         LDA 0.7500000       0.840
3  SVM Radial 0.7395833       0.864
4 SVM Lineare 0.7500000       0.848
5         KNN 0.7447917       0.856
```

Figure 22: Accuracy e sensitivity test

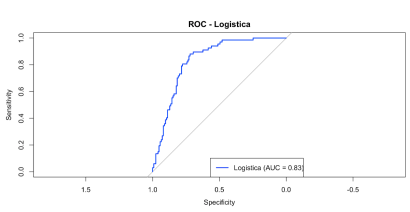
I modelli **LDA** e **SVM** Lineare mostrano l'accuratezza più alta (75.0 %), seguiti da **Logistica** e **KNN** (74.5%), e infine **SVM Radiale** (73.9%).

Il **KNN** ha la sensibilità più alta (85.6%), seguito da **SVM Radiale** (86.4%), **Logistica** e **SVM Lineare** (84.8%), e infine **LDA** (84.0%).

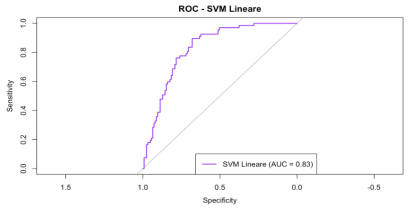
Un altro modo di verificare la bontà della performance del classificatore è quello di passare per la ***receiver operating characteristic*** **(ROC)** ***curve*** o ***curve di*** **ROC**. Queste curve mettono a confronto il **True positive rate (la sensitività**) e il **false positive rate.**

Quello che ci aspettiamo è che il primo sia maggiore del secondo e che quindi la curva stia sulla parte superiore della bisettrice (che è il caso in cui il TPR = FPR). Quello che vorrei è che la curva di ROC sia spinta il più possibile verso l'angolo in alto a sx il che implica che il metodo è buono e che sto abbattendo il FPR.

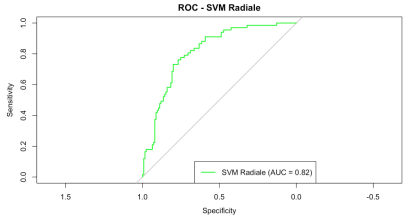
Inoltre abbiamo calcolato l'**AUC**(*Area Under the Curve*) è una misura che quantifica l'abilità di un modello di classificazione binaria nel distinguere tra le classi. È l'area sotto la curva ROC (Receiver Operating Characteristic) e varia tra 0 e 1, dove 1 indica un modello perfetto e 0.5 un modello che classifica casualmente.



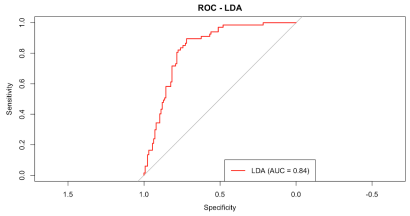
(a) ROC Regressione Logistica



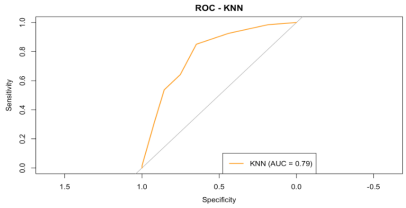
(b) ROC SVM Lineare



(c) ROC SVM Radiale



(d) ROC LDA



(e) ROC KNN

Figure 23: Curve ROC dei Modelli di Classificazione

Il valore più alto di **AUC** è dato dal modello **LDA** con un valore di **0.84**, questo indica che il modello ha una buona capacità di classificare correttamente

le osservazioni. Seguono con **0.83** il modello di regressione logistica e SVM Lineare.

3 Conclusioni

L'analisi condotta ha mostrato l'efficacia dei vari modelli di classificazione per la diagnosi del diabete. I modelli che abbiamo utilizzato sono stati la regressione logistica, l'analisi discriminante lineare (LDA), il Support Vector Machine (SVM)con kernel lineare e radiale, e il K-Nearest Neighbors (KNN).

Dai risultati ottenuti, emerge che i modelli LDA e SVM Lineare offrono la migliore accuratezza sul test set, con una percentuale del 75%. Tuttavia, il modello KNN si distingue per la sensibilità più alta, suggerendo una migliore capacità di rilevare correttamente i casi positivi di diabete. Anche la curva ROC ed in particolare il valore di AUC ci ha mostrato che il modello LDA risulta essere quello che ha una migliore capacità di classificare correttamente le osservazioni.

In conclusione nonostante il modello LDA sembrerebbe essere quello che ha classificato meglio le osservazioni, nessuno dei modelli emerge come nettamente superiore in tutte le metriche, evidenziando la complessità intrinseca del problema di classificazione.