Relazione Apprendimento Computazionale

Dermatology

Ludovico Ferreri

Indice

Comprensione del problema e obiettivi dell'analisi

Il database usato per questa analisi è una collezione di dati clinici e istopatologici relativi a malattie eritemato-desquamative ben note nel campo della Dermatologia, questo perché, a causa delle caratteristiche cliniche condivise con eritema e desquamazione, la loro differenziazione è un problema complesso.

Il dataset si compone di 366 tuple descritte da 34 attributi di particolare interesse per discernere le diverse patologie. Comprende 32 attributi categorici (>2 categorie), un attributo binario e uno continuo. La valutazione clinica coinvolge 12 caratteristiche, come erythema, scaling, definite borders, itching, e mucosal involvement. L'analisi istopatologica, invece, esamina 22 caratteristiche, tra cui melanin incontinence, eosinophils in the infiltrate, fibrosis of the papillary dermis ecc. Queste caratteristiche sono classificate su una scala da 0 a 3, indicando l'assenza o i vari gradi della loro presenza.

Le malattie prese in considerazione durante la costruzione del database sono:

1. Psoriasis
2. Seborrheic dermatitis
3. Lichen planus
4. Pityriasis rosea
5. Chronic dermatitis
6. Pityriasis rubra pilaris

Inoltre, oltre che l'età dei pazienti, il database fornisce approfondimenti sulla loro storia familiare, indicando se una qualsiasi delle malattie è stata osservata all'interno delle loro famiglie.

Il database ha 8 valori mancanti, tutti relativi all'età, indicati da un '?'.

Nella seguente analisi, servendosi di Orange e MATLAB, verrà costruito un modello predittivo tramite l'utilizzo di tecniche di machine learning.

Visualizzazione e Preparazione dei dati

Il primo passo di questa analisi riguarda il pre-processing, quindi "pulire", trasformare e preparare i dati in un formato adatto a un'ulteriore elaborazione.

La fase di pre-processing comprenderà:

1. Esplorazione e Visualizzazione dei dati
2. Outliers
3. Feature Selection e Analisi della Variabilità

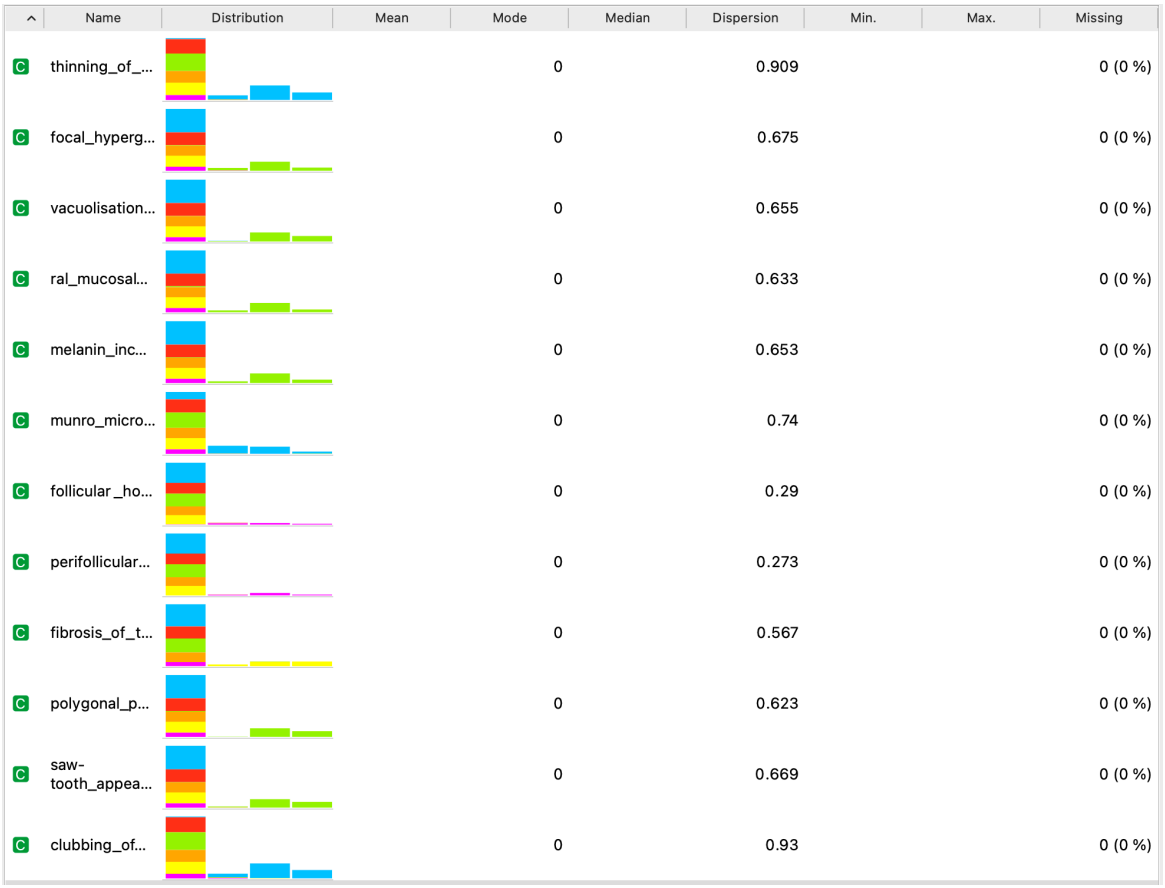
Esplorazione e Visualizzazione dei dati

La visualizzazione e esplorazione dei dati svolgono un ruolo cruciale nella comprensione e acquisizione di informazioni provenienti dal data-set sotto analisi.

Usando Orange leggo il dataset e, per prima cosa, vado a fare imputing degli 8 dati mancanti sull'età sostituendoli con la media.

Dato che la quasi totalità delle feature sono categoriche, scelgo di rappresentare i dati tramite istogrammi. Inoltre, per capire se qualitativamente, alcune feature hanno un qualche tipo di associazione con l'outcome, li vado a plottare contro la variabile target.

Come si evince dalla figura 1, alcune categorie di determinati attributi hanno una forte associazione ad uno dei sei outcome.



Name	Distribution	Mean	Mode	Median	Dispersion	Min.	Max.	Missing
thinning_of_...			0		0.909			0 (0 %)
focal_hyperg...			0		0.675			0 (0 %)
vacuolisation...			0		0.655			0 (0 %)
ral_mucosal...			0		0.633			0 (0 %)
melanin_inc...			0		0.653			0 (0 %)
munro_micro...			0		0.74			0 (0 %)
follicular_ho...			0		0.29			0 (0 %)
perifollicular...			0		0.273			0 (0 %)
fibrosis_of_t...			0		0.567			0 (0 %)
polygonal_p...			0		0.623			0 (0 %)
saw-tooth_appea...			0		0.669			0 (0 %)
clubbing_of...			0		0.93			0 (0 %)

fig. 1 Istogrammi di alcune feature, plottati contro l'outcome (tipo di malattia della pelle)

Dato che le feature sono fortemente sbilanciate e le categorie associate ([1 2 3]) maggiormente ad un singolo outcome hanno un significato simile tra di loro (seppur indicano i diversi gradi di severità del sintomo, ne rappresentano tutti almeno la presenza), decido di **binarizzarle**.

Questa scelta è anche supportata dal fatto che vado a diminuire il peso degli outlier e mi libero della variabilità introdotta dalle categorie rare.

Per far ciò mi sposto su MATLAB.

Lavorando con variabili categoriche, non è saggio utilizzare la correlazione per studiare l'associazione tra gli attributi. Esiste la correlazione di Spearman, ma è stata creata per confrontare variabili continue contro categoriche (non funziona bene per quelle nominali).

Per questo motivo utilizzo il test di indipendenza del χ^2 per vedere quali variabili sono dipendenti tra di loro.

Questo test da solo però non basta, perché mi dice solamente se l'indipendenza è statisticamente rilevante o meno, ma non mi dà nessuna

informazione riguardo la forza del legame. Per questo motivo vado anche a calcolare la V di Cramer.

```
% ----------------------  LETTURA DATI -------------------------
warning('off','all');
% La readtable da un warning perché i nomi degli attributi vengono
% modificati da Matlab. Per questo vado a sopprimerli.
Matrice = readtable("Dati_Nuovo.csv");

Header = Matrice.Properties.VariableNames;
Dati = Matrice.Variables;
[m, n] = size(Dati);

n = n-1;
```

Decido di confrontare il database **prima** della binarizzazione delle variabili e **dopo** per vederne l'effetto. Vado quindi a Binarizzare le variabili selezionate precedentemente e vado ad incorporare una categoria rara di una variabile non presente in quelle mostrate. Questo perché il test del χ^2 funziona male per categorie con frequenze < 5.

```
% ----------------------  LETTURA DATI -------------------------
for i = 1:n
    freq{i} = tabulate(Dati(:,i)); %Calcolo le tab. di frequenza

    if sum(freq{i}(:,2) < 5) ~= 0 %Se ho categorie < 5

        Header{i}       % Nome feature
        find(freq{i}(:,2) < 5) - 1  % Categorie con frequenza < 5

    end
end
```

```
ans = 'vacuolisation_and_damage_of_basal_layer'
  ans = 1

ans = 'erythema'
  ans = 0

ans = 'follicular_horn_plug'
  ans = 3

ans = 'perifollicular_parakeratosis'
  ans = 2×1
       1
       3


ans = 'polygonal_papules'
  ans = 1
```

```
ans = 'band_like_infiltrate'
  ans = 1
```

Tutte queste variabili, tranne ***erythema*** e ***band like infiltrate***, fanno parte delle variabili già precedentemente selezionate su Orange. Per **band like infiltrate** decido di collassare solo la categoria 1 con la 2. **erythema**, invece, rimarrà invariato perché la categoria 0, l'assenza della caratteristica, è l'antitesi di [1 2 3].

```
% Variabili che vado a Binarizzare
%    2 - thinning_of_the_suprapapillary_epidermis
%    3 - focal_hypergranulosis
%    4 - vacuolisation_and_damage_of_basal_layer
%    5 - ral_mucosal_involvement
%    7 - melanin_incontinence
%   20 - munro_microabcess
%   23 - follicular_horn_plug
%   24 - perifollicular_parakeratosis
%   27 - fibrosis_of_the_papillary_dermis
%   29 - polygonal_papules
%   30 - saw_tooth_appearance_of_retes
%   31 - clubbing_of_the_rete_ridges


Dati2 = Dati;
Binarizza = [2 3 4 5 7 20 23 24 27 29 30 31];
for i = Binarizza
    Dati2(Dati2(:,i) ~= 0, i) = 1;
end

Dati2(Dati2(:,33) == 1, 33) = 2;
```

Ora faccio il test del χ^2 e calcolo la V di Cramer.

```
ppre = zeros(n,n); %Matrice con i p-value pre-binarizzazione
Vpre = zeros(n,n); %Matrice con la V di Cramer prebinarizzazione
ppost = zeros(n,n); %Matrice con i p-value post-binarizzazione
Vpost = zeros(n,n); %Matrice con la V di Cramer post-binarizzazione

for i = 1:n
    for j = 1:n
        % CALCOLO PRE-BINARIZZAZIONE
        % Test Chi2
        [tbl, chi2, ppre(i,j)] = crosstab(Dati(:,i), Dati(:,j));

        % Cramer's V
        %                X²
        % V = √ ------------------
        %        m * min(r-1,c-1)
        Vpre(i,j) = sqrt(chi2/(m * (min(size(tbl)-1)) ) );

        % CALCOLO POST-BINARIZZAZIONE
        % Test Chi2
        [tbl, chi2, ppost(i,j)] = crosstab(Dati2(:,i), Dati2(:,j));
        Vpost(i,j) = sqrt(chi2/(m* (min(size(tbl)-1)) ) );
    end
end

% --------------------- PLOTTO DATI P-VALUE ----------
%
    figure(1)
    subplot(1,2,1)
    imagesc(ppre);
    colorbar;
    set(gca, 'XTick', 1:n, 'XTickLabel', Header, 'YTick', 1:n,
'YTickLabel', Header);
    xtickangle(90);
    ax = gca;
    ax.XAxisLocation = 'top';

    subplot(1,2,2)
    imagesc(ppost);
    colorbar;
    set(gca, 'XTick', 1:n, 'XTickLabel', Header, 'YTick', 1:n,
'YTickLabel', Header);
    xtickangle(90);
    ax = gca;
    ax.XAxisLocation = 'top';
    fig = gcf;
    fig.Position(3:4) = [1400 600];
```

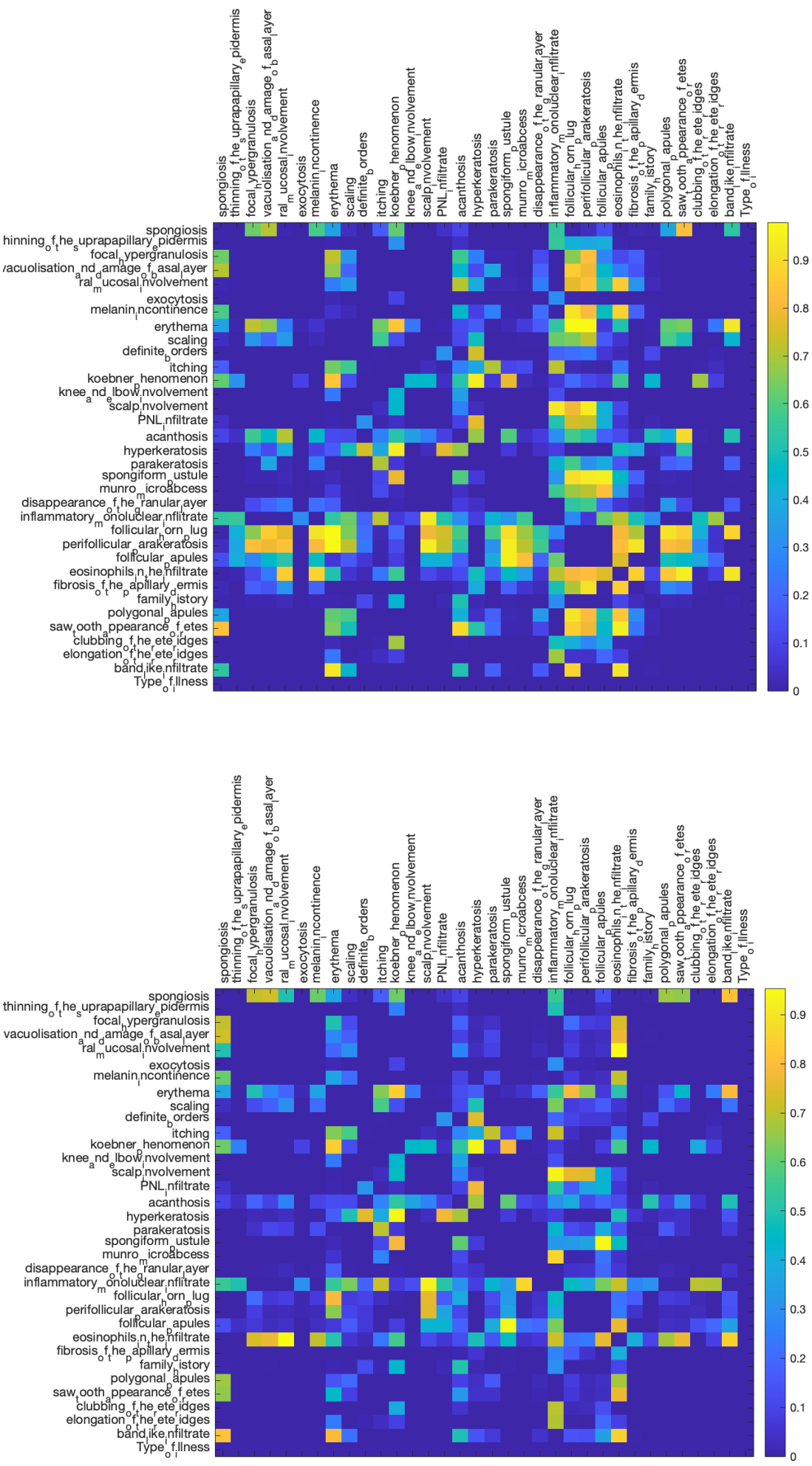


fig. 2 HeatMap dei p-value del test di indipendenza del χ^2, prima e dopo la binarizzazione (ultima riga/colonna è il target)

Nel grafico in **figura 2** vengono plottati i p-value del test del χ^2, più il colore è blu più è basso il p-value, con più sicurezza vado a rifiutare l'ipotesi nulla (H0: Le 2 variabili sono indipendenti)

Come ci si poteva aspettare, andando a diminuire l'informazione (variabili ordinali → nominali), l'indipendenza tra le variabili è, generalmente, peggiorata.

Bisogna però vedere cosa succede alla forza del loro legame, sopratutto con la variabile target.

```
% ---------------------- PLOTTO DATI Cramers'V ----------------------
    figure(1)
    subplot(1,2,1)
    imagesc(ppre);
    colorbar;
    set(gca, 'XTick', 1:n, 'XTickLabel', Header, 'YTick', 1:n,
'YTickLabel', Header);
    xtickangle(90);
    ax = gca;
    ax.XAxisLocation = 'top';

    subplot(1,2,2)
    imagesc(ppost);
    colorbar;
    set(gca, 'XTick', 1:n, 'XTickLabel', Header, 'YTick', 1:n,
'YTickLabel', Header);
    xtickangle(90);
    ax = gca;
    ax.XAxisLocation = 'top';
    fig = gcf;
    fig.Position(3:4) = [1400 600];
```

Nel grafico in **figura 3**, vengono plottate le V di Cramer. In questo caso, più il valore è vicino ad 1 più è forte la relazione tra 2 variabili.

Si nota subito che il legame tra alcuni degli attributi binarizzati e la variabile target, è quasi raddoppiato (Ultima colonna).

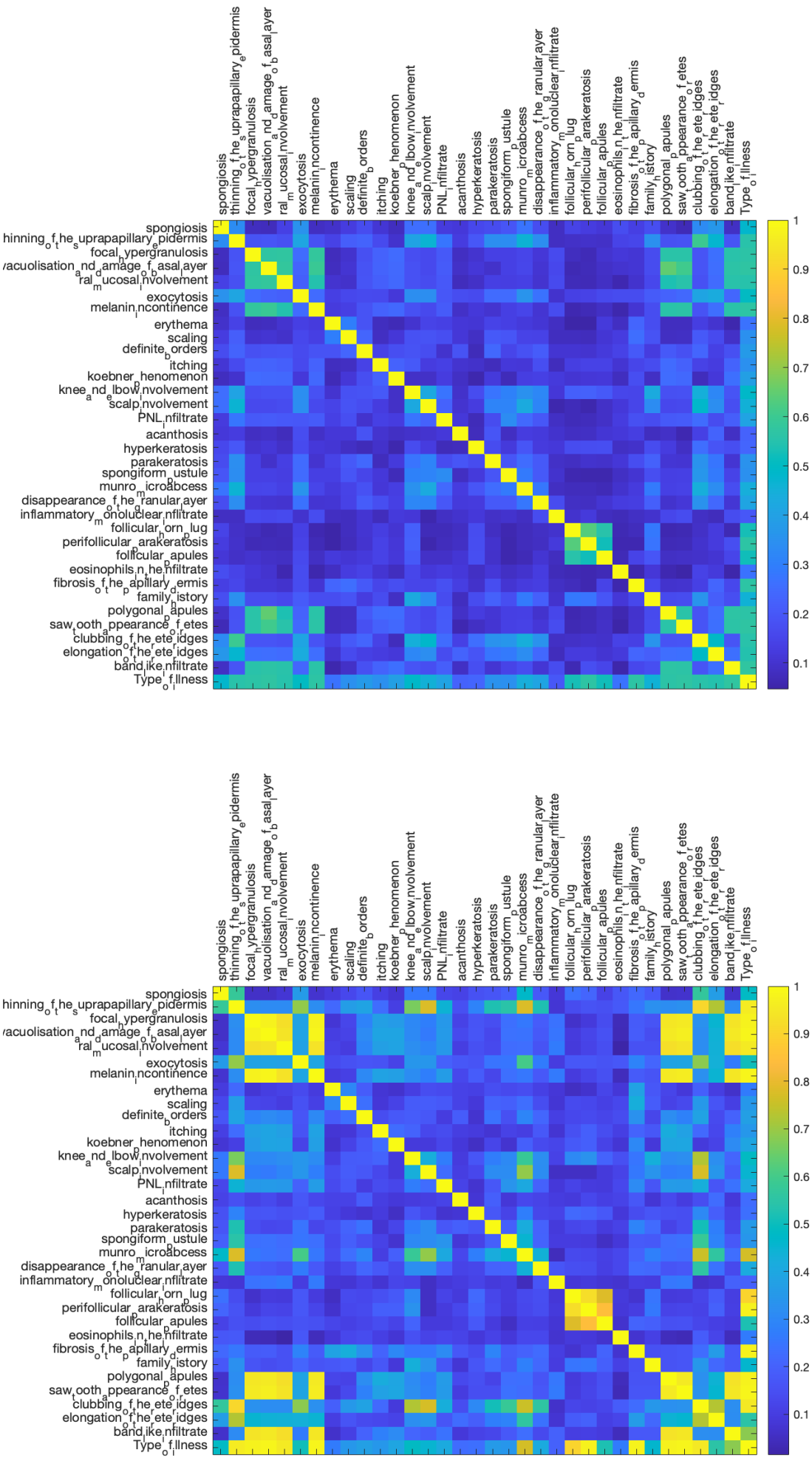


fig. 3 HeatMap della V di Cramer, prima e dopo la binarizzazione (ultima riga/colonna è il target)

Tramite la funzione creaCSV vado ad esportare il database in modo da poter essere utilizzato da Orange.

```
function creaCSV(Data,Header)

nomi = strjoin(Header, ', ');
dominio = "";
for col = 1:(size(Data, 2)-1)
    val = unique(Data(:, col));
    if col == 1
        dominio = [strjoin(string(val), ' ')];
    else
        dominio = [dominio strjoin(string(val), ' ')];
    end
end
dominio = [dominio "continuous"];
dominio = strjoin(dominio, ', ');

virgole = repmat(',', 1, size(Data, 2));

dati = '';
for row = 1:size(Data, 1)
    dati = [dati strjoin(string(Data(row, :)), ',')];
end

filename = 'Database_PP_Matlab.csv';
fid = fopen(filename, 'w');
fprintf(fid, '%s\n', nomi);
fprintf(fid, '%s\n', dominio);
fprintf(fid, '%s', virgole);
fprintf(fid, '%s\n', dati);
fclose(fid);
end
```

Outliers

Rileggo il database e utilizzo la funzione *t-sne (figura 4)* per ridurre la dimensionalità dei dati, così da poter visualizzare i cluster ed eventuali outliers.

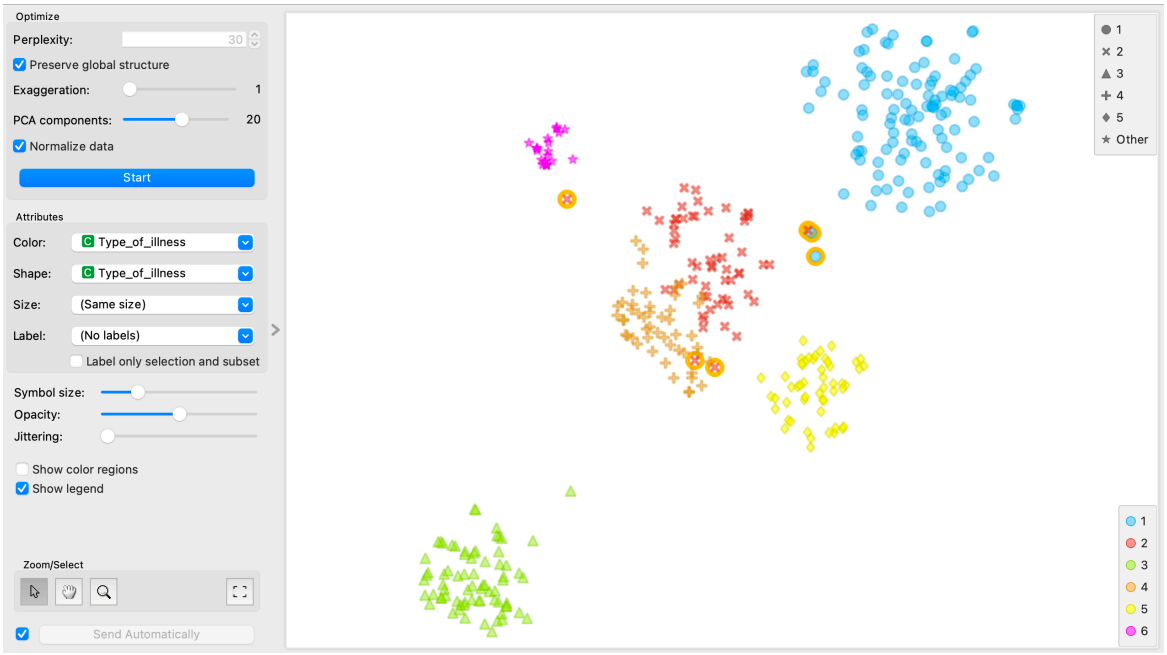


fig. 4 Output della funzione t-sne, guardo i cluster e gli outliers

I dati sono ben clusterizzati, tranne che per le categorie 2 e 4 che si confondono tra di loro. Molto probabilmente la difficoltà della predizione ricadrà nel discernere tra **Seborrheic dermatitis** e **Chronic dermatitis**.

Dato che hanno caratteristiche simili e sono entrambi dermatiti potrebbe avere senso combinarle, però non avendo conoscenze specifiche, ne pareri di esperti nel campo della dermatologia, decido di lasciarle invariate.

Dall'analisi noto che alcune tuple possono essere viste come outliers, le quali possono portare ad overfitting e peggiorare la generalizzazione. Quindi ne seleziono 6 (cerchio arancione in figura 4) e li elimino.

Feature Selection

Per scegliere il subset di feature che comporranno il mio modello, utilizzerò una tecnica di feature selection supervisionato.

Prima però, per evitare overfitting, vado a dividere il dataset in Training e Test (70/30). Continuerò a lavorare sul Training e non toccherò il Test set fino alla fine dell'analisi.

Per la feature selection, mi servirò della funzione *rank* di Orange, per il calcolo di una misura di dispersione per ogni feature. Il problema principale però, è che avendo diversi Outcome, le feature più informative saranno monopolizzate dai gruppi più numerosi (figura 5) e meglio clusterizzati.

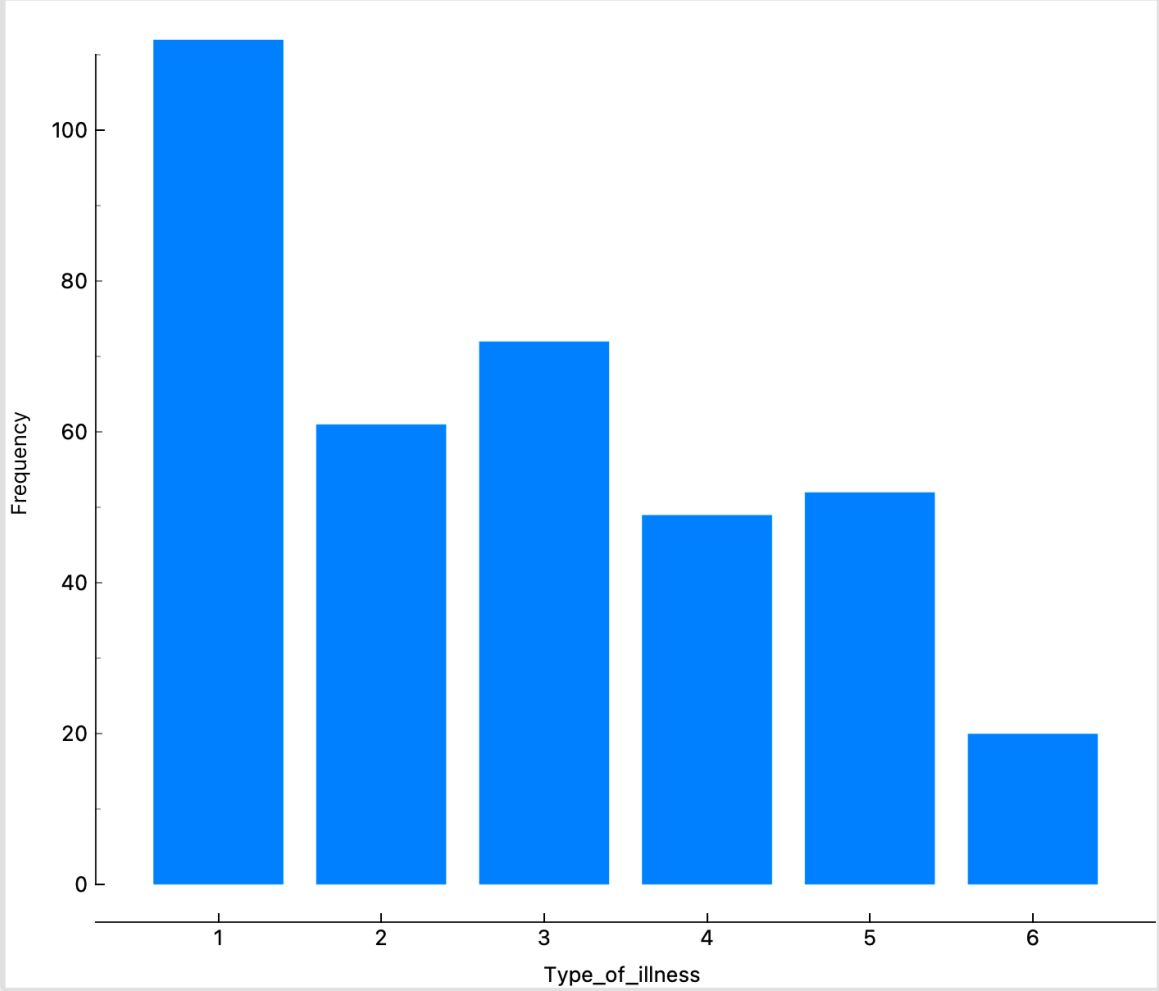


fig. 5 Distribuzione della variabile Target

Per questo motivo vado a creare una dummy variable per ogni singola malattia ([1:6]) della variabile Target. Dopodiché utilizzo la *rank* per calcolare

gli attributi più informativi relativi alle singole malattie. Dato che ho variabili solo categoriche / nominali, utilizzerò l'indice ReliefF.

Per ogni outcome scelgo la feature più informativa. Utilizzando il metodo Relief più volte sullo stesso database, avrò ogni volta un set di feature leggermente diverso. Facendo lo scoring 10 volte ho ottenuto i seguenti risultati:

1. Fibrosis of the papillary dermis - 10 volte
2. Perifollicular parakeratosis - 10 volte
3. Spongiosis - 10 volte
4. Koebner phenomenon - 7 volte
5. Melanin incontinence - 6 volte
6. Clubbing of the rete ridges - 5 volte
7. Thinning of the suprapapillary epidermis - 3 volte
8. Focal hypergranulosis - 1 volte
9. Saw-tooth appearance of retes - 1 volte
10. Band-like infiltrate - 1 volte

Sceglierò quindi le 6 feature più numerose (una per ogni outcome, non in ordine):

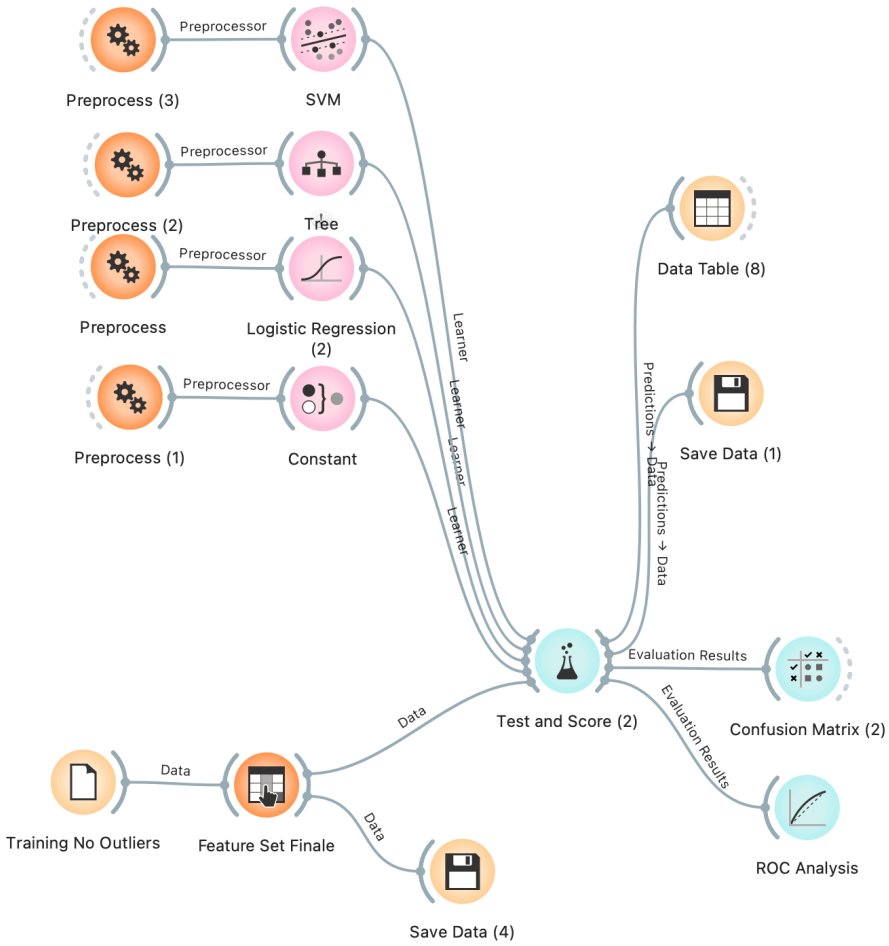
1. fibrosis of the papillary dermis
2. perifollicular parakeratosis
3. Spongiosis
4. Koebner phenomenon
5. Melanin incontinence
6. Clubbing of the rete ridges

4 delle quali sono state precedentemente binarizzate.

Modellizzazione, Validazione e Scelta del modello

Scelte le feature, vado ad utilizzare diversi modelli e vedo quale funziona meglio. Questa analisi verrà fatta sempre sul training set, utilizzando una 10-fold cross validazione.

Gli algoritmi scelti sono Regressione Logistica, Tree e SVM.



Scelgo questi 3 perché sono gli algoritmi migliori, tra quelli disponibili, a gestire variabili categoriche e nominali, ma anche a causa della spiegabilità dei modelli.

Dai risultati della modellizzazione, vedo che la regressione logistica è la migliore, ma gli altri modelli sono comunque validi.

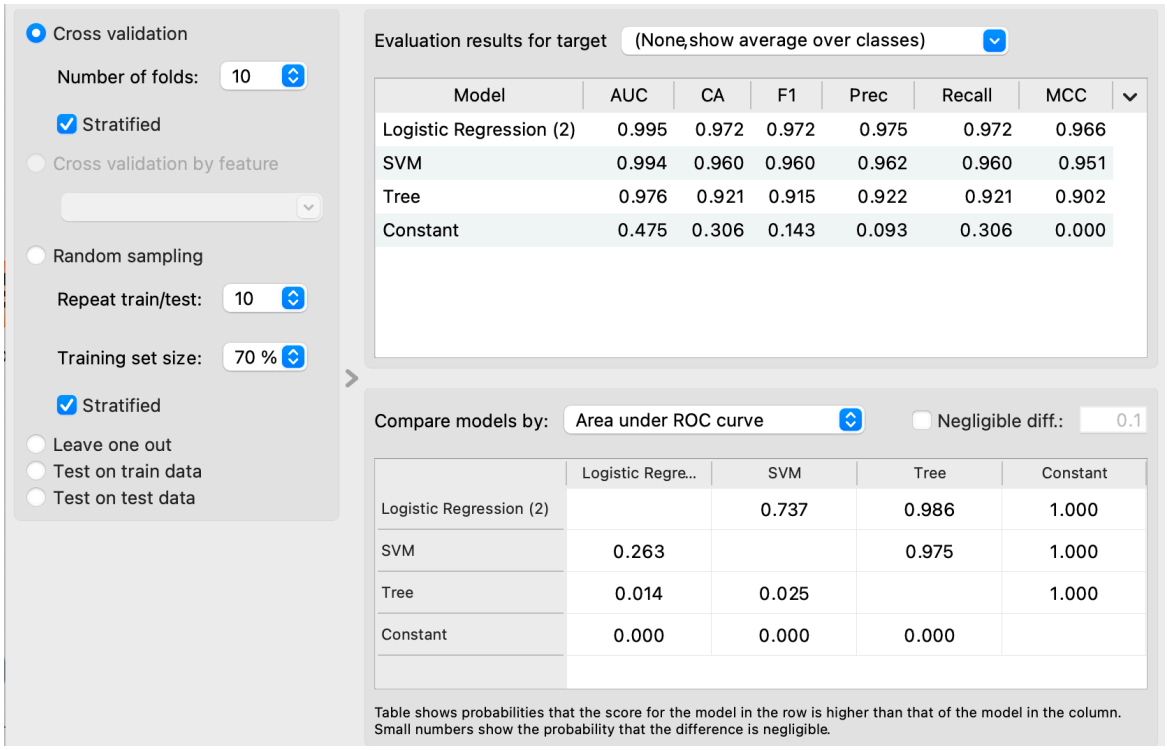


fig. 6 Risultato del test and score, utilizzando la 10-fold cross validation

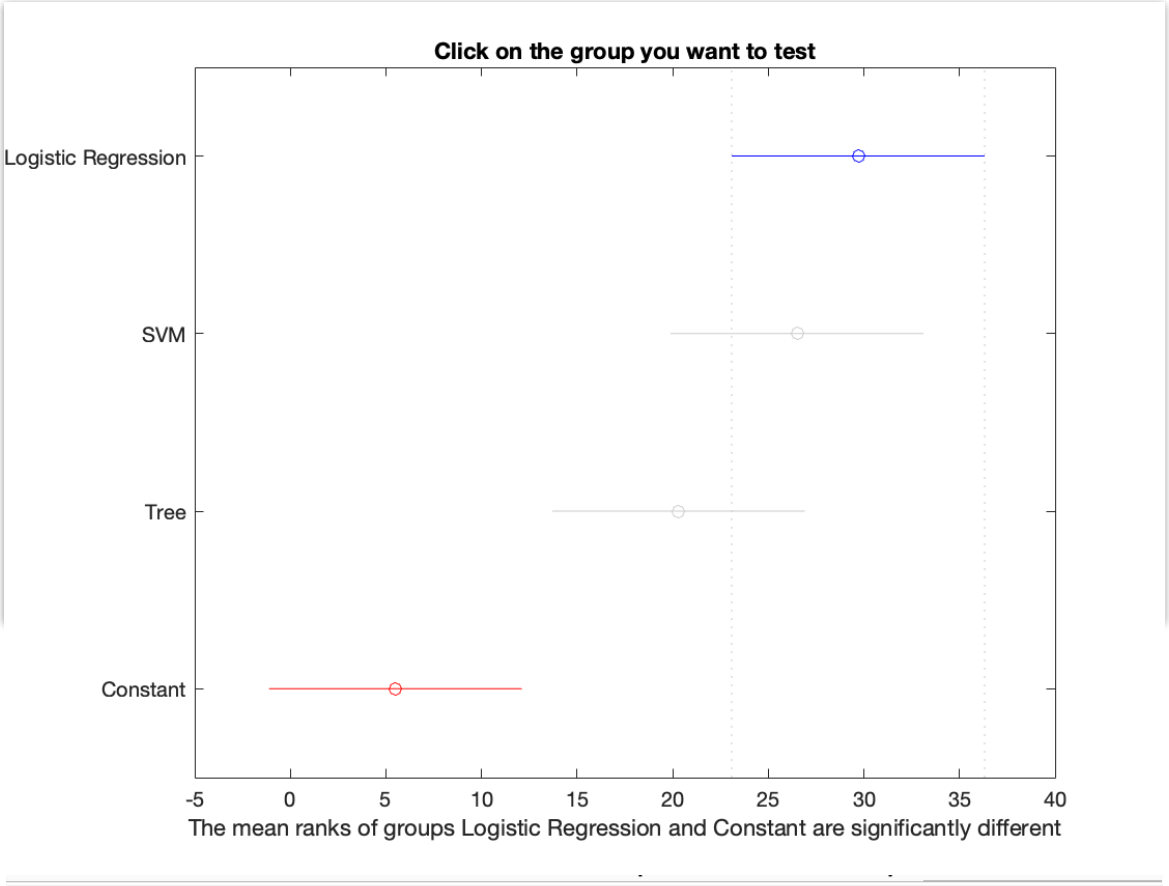
Una volta fatto ciò, esporto i risultati della cross validazione e vado a calcolare Accuratezza e Intervalli di Confidenza dei 3 Classificatori.

```
Matrice = readtable("Classificatori.csv");
Header = Matrice.Properties.VariableNames;
Dati = Matrice.Variables;

Dati = Dati(2:end,[1 2 3 4 5 30]);
[m, n] = size(Dati);
k = 10;
Fold = 1:k;
Accuratezza = zeros(4,k);
% colonne 2:5 contengono le predizioni dei 4 modelli
for j = 2:5
    for i = Fold  % Per ogni fold, prendi dati e calcola accuratezze
        indFold = Dati(:,end) == i;
        k = sum(indFold);
        Accuratezza(j-1,i) = sum(Dati(indFold, j) == Dati(indFold,
1))/k;

    end

    Header{j}
    StdAcc = std(Accuratezza(j-1,:));
       AccMedia = mean(Accuratezza(j-1,:))
    [AccMedia - tcdf(k-1,1-0.05/2)*StdAcc/sqrt(k)  AccMedia +
tcdf(k-1,1-0.05/2)*StdAcc/sqrt(k)]

% IC = [Am ± t(k-1,1-α/2)*s/√k] IC del 95%

end
```

```
ans =
    'LogisticRegression'
AccMedia =
    0.9722
IC =
    0.9668    0.9775

ans =
    'SVM'
AccMedia =
    0.9602
ans =
    0.9549    0.9654



ans =
    'Tree'
AccMedia =
    0.9206
ans =
    0.9108    0.9304

ans =
    'Constant'
AccMedia =
    0.3055
ans =
    0.3019    0.3091
```

Sono tutti abbastanza simili tra di loro, la regressione logistica sembra ancora una volta la migliore, per sicurezza vado comunque a fare un test statistico.

Non potendo fare ipotesi sulla distribuzione statistica delle accuratezze, vado ad utilizzare il test non parametrico di kruskall-wallis, andando poi a comparare i diversi gruppi.



Kruskal-Wallis ANOVA Table

Source	SS	df	MS	Chi-sq	Prob>Chi-sq
Groups	3456.8	3	1152.27	26.09	9.12545e-06
Error	1710.2	36	47.51		
Total	5167	39			

fig. 7 Comparazione delle accuratezze tra i modelli scelti

Come ci si poteva aspettare, la regressione logistica, SVM e Tree sono simili tra di loro. Vado a fare un ulteriore confronto sulle predizioni, andando a vedere la matrice di confusione per ogni modello.

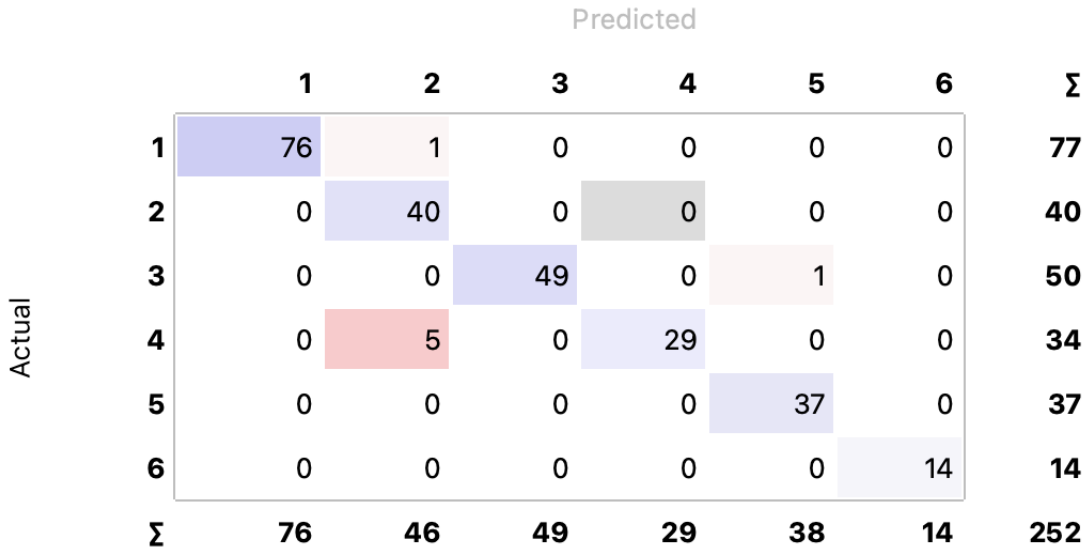


fig. 8 Matrice di confusione su una fold della Regressione Logistica

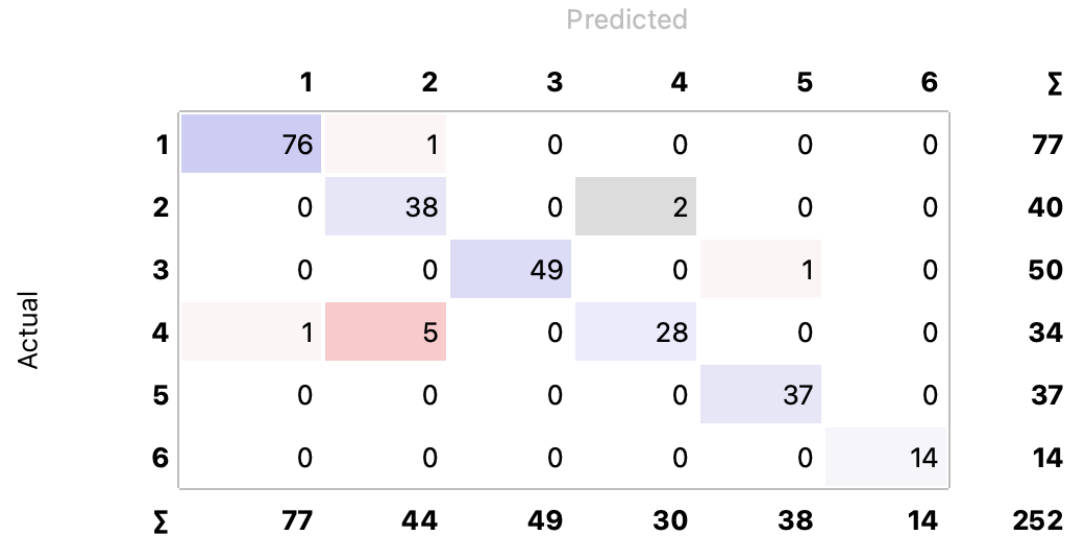


fig. 9 Matrice di confusione su una fold della Support Vector Machine

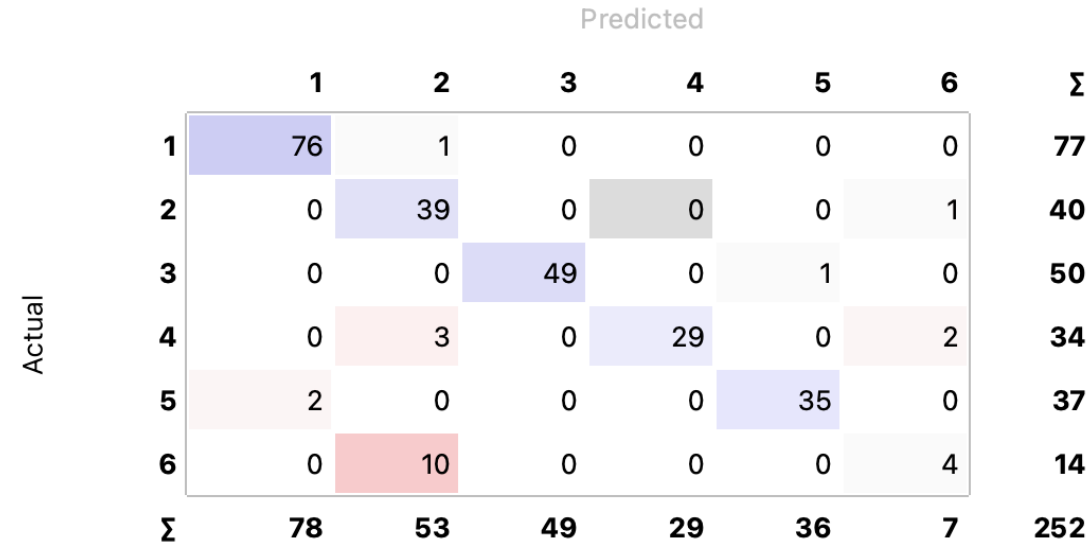


fig. 10 Matrice di confusione su una fold dell'Albero Decisionale

Guardando la matrice di confusione, calcolata su una fold, si nota che SVM e Regressione Logistica sono molto simili tra di loro, ma l'albero decisionale sbaglia quasi tutte le predizione della sesta malattia.

Andando a guardare le Precision e Recall medie, in figura 6, si può evincere che per le altre fold la situazione non cambia. La scelta sarà quindi tra Regressione Logistica e SVM.

Andrò a scegliere la regressione logistica perché, guardando l'accuratezza media e gli intervalli di confidenza, funziona leggermente meglio. Il motivo principale però, riguarda la spiegabilità del modello. Infatti con la Regressione Logistica sono in grado di esportare i coefficienti β per ogni feature, andando a vedere il peso che ognuno di essi ha sulla scelta dell'outcome.

Testing del modello

Una volta deciso il modello, lo vado a testare sul data set di test.

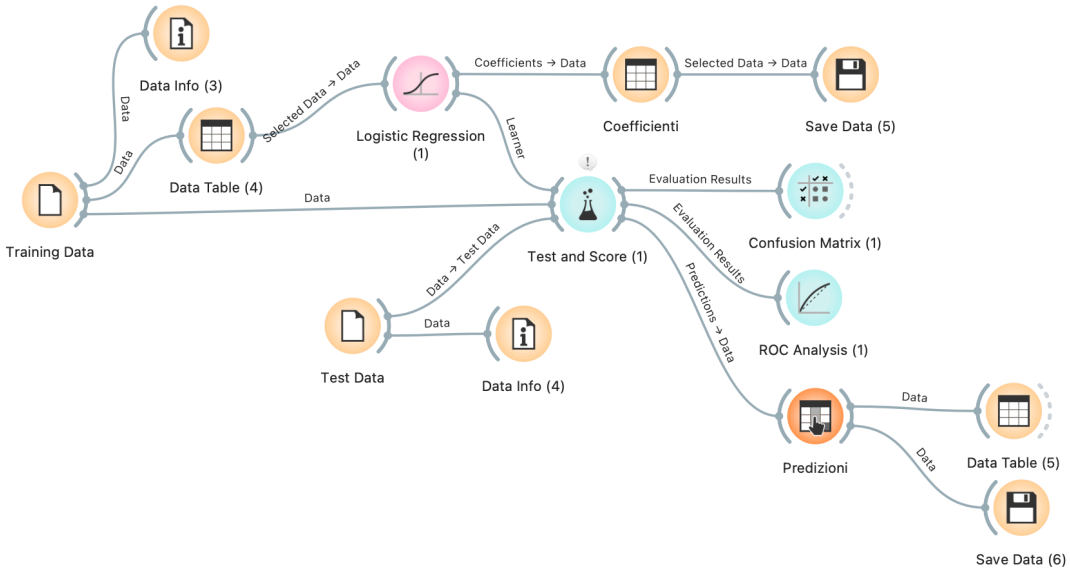


fig. 11 WorkFlow Orange per la fase di testing

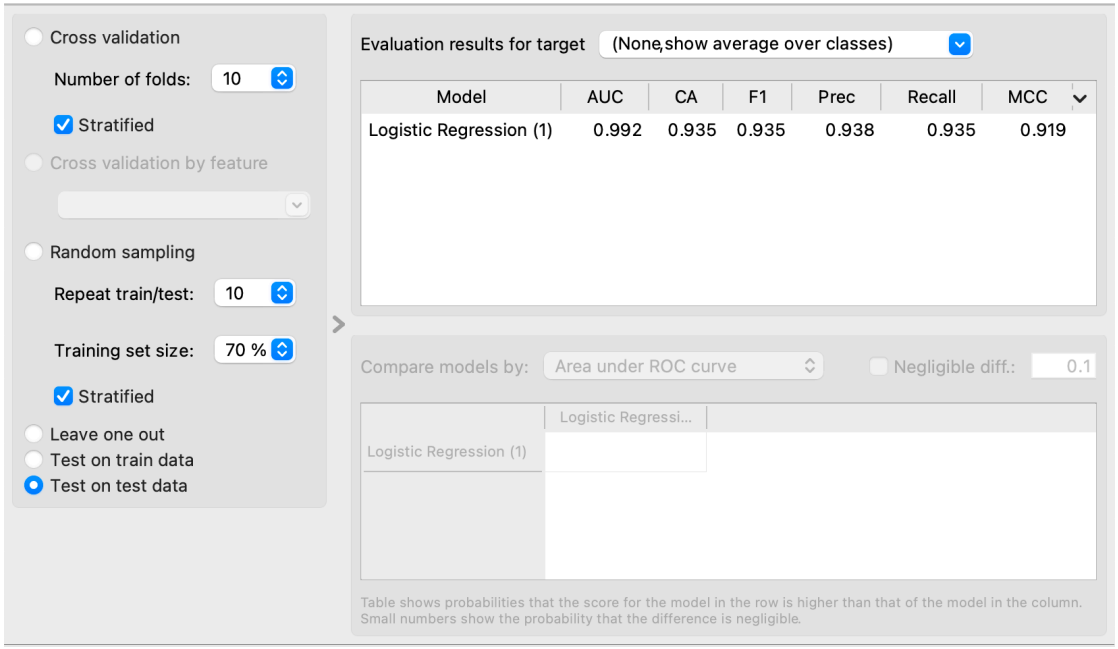


fig. 12 Output del Test and Score, con testing sul test-set

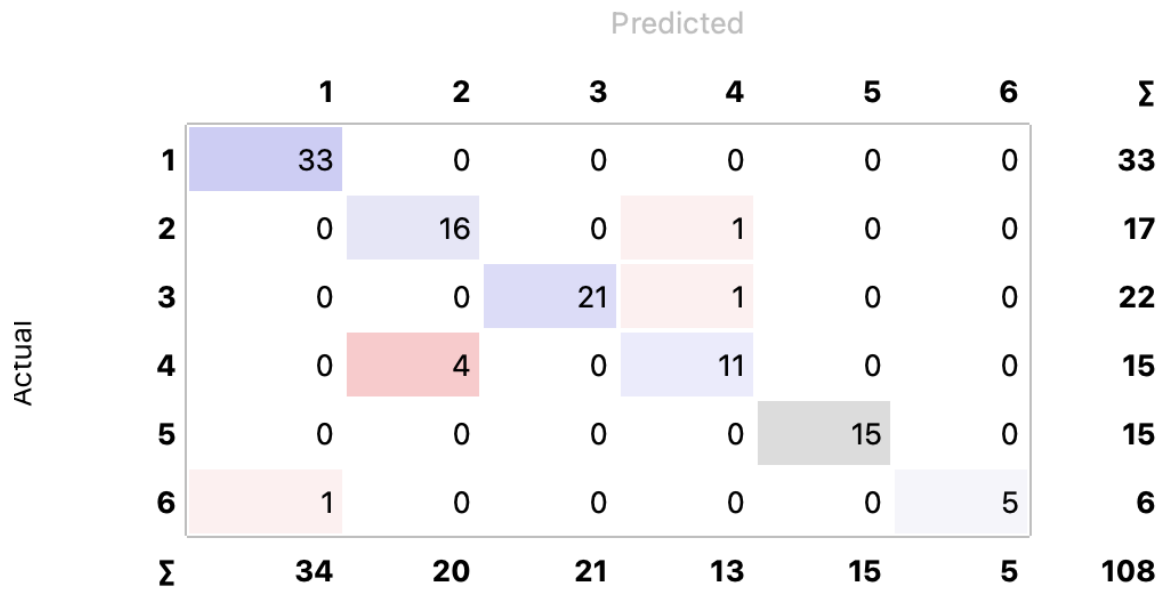


fig. 13 Matrice di confusione sul test-set

Come ci si poteva aspettare dall'analisi t-sne, il problema principale riguarda gli outcome 2 e 4.

Ora calcolo l'accuratezza generale.

```
Matrice = readtable("Predizioni.csv");
Dati = Matrice.Variables;

Accuratezza = sum(Dati(:,1) == Dati(:,2))/length(Dati)
```

```
Accuratezza =

    0.9352
```

L'accuratezza Generale è di **0.9352**

Spiegabilità del modello

Ogni coefficiente della Regressione Logistica (β) rappresenta un aumento nel log-odds in corrispondenza dell'aumento di un'unità del valore della feature, mentre tutte le altre rimangono costanti.

Guardando il modulo di ogni coefficiente posso andare a vedere quanto influisce sulla decisione, e quindi riesco a vedere quanto sia importante ogni categoria di ogni feature.

	name	1	2	3	4	5	6
1	intercept	0.207931	-2.49273	2.05237	-0.351619	0.346858	0.237189
2	fibrosis_of_the_papillary_dermis=0	0.589332	0.88268	-0.29613	0.688865	-2.09949	0.234744
3	fibrosis_of_the_papillary_dermis=1	-0.589399	-0.88293	0.296211	-0.688809	2.09974	-0.234814
4	perifollicular _parakeratosis=0	0.275168	0.880647	0.157455	0.442387	0.199236	-1.95489
5	perifollicular _parakeratosis=1	-0.275235	-0.880897	-0.157374	-0.442331	-0.198987	1.95482
6	spongiosis=0	1.34387	-0.676243	-0.0821763	-0.940441	0.372506	-0.0175192
7	spongiosis=1	-0.258871	-0.55434	-0.0600722	0.57483	0.158531	0.139922
8	spongiosis=2	-0.672414	0.503689	0.147764	0.452604	-0.39926	-0.032384
9	spongiosis=3	-0.412656	0.726643	-0.00543424	-0.086937	-0.131528	-0.0900888
10	clubbing_of_the_rete_ridges=0	-1.90894	0.69042	0.301274	0.685404	0.0294507	0.202388
11	clubbing_of_the_rete_ridges=1	1.90887	-0.69067	-0.301192	-0.685348	-0.0292015	-0.202457
12	melanin_incontinence=0	0.359431	0.772166	-2.3509	0.87873	0.186002	0.154572
13	melanin_incontinence=1	-0.359498	-0.772416	2.35098	-0.878674	-0.185753	-0.154642
14	koebner _phenomenon=0	-0.203197	1.6046	-0.725491	-1.72464	0.688117	0.360604
15	koebner _phenomenon=1	0.158373	-0.786265	0.100256	0.855234	-0.14957	-0.178028
16	koebner _phenomenon=2	0.0254417	-0.559062	0.572763	0.582869	-0.49292	-0.129091
17	koebner _phenomenon=3	0.0193158	-0.259526	0.0525547	0.286589	-0.0453786	-0.0535546

fig. 14 Coefficienti della Regressione Logistica

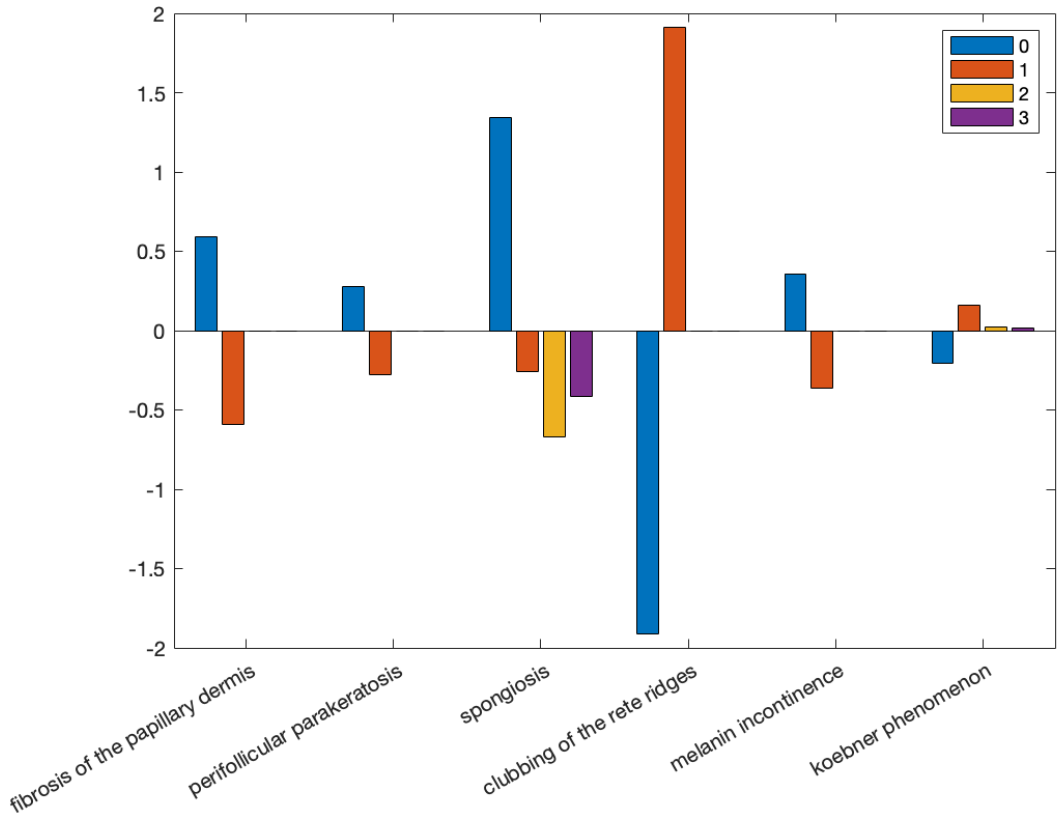


fig. 15 Diagramma a barre dei Beta relativi all'Outcome 1

Prendendo il primo outcome come esempio, graficato in figura 15, si nota che la decisione relativa per questo specifico outcome sarà determinata per lo più da "*clubbing of the rete ridges",* in particolare, se il sintomo è presente allora aumenterà la probabilità che questa sia la predizione da scegliere, e viceversa (Anche *"spongiosis" = 0, ne aumenterà significativamente la probabilità)*

Utilizzo del modello

La vera potenza di questo modello predittivo si potrà capire solo se usato in clinica, in aiuto, e non in sostituzione, ai dermatologi. Questo per minimizzare le classificazioni errate che potrebbero essere commesse sia dal modello che dal dottore.

Una criticità riguardo al modello, una volta che verrà implementato su casi reali, riguarda la variabilità della diagnosi dei sintomi dei diversi dermatologi.

Infatti, andando a rivedere le feature scelte:

1. fibrosis of the papillary dermis [0 1]
2. perifollicular parakeratosis [0 1]
3. Spongiosis [0 1 2 3]
4. Koebner phenomenon [0 1 2 3]
5. Melanin incontinence [0 1]
6. Clubbing of the rete ridges [0 1]

4 di queste feature indicano la presenza/assenza del sintomo, ma 2 di queste ne indicano l'assenza o i vari gradi della loro presenza. Quindi, a meno che non esista una standard ben preciso, è possibile che alcuni dermatologi classifichino il sintomo di uno stesso paziente in maniera diversa.

Se, per esempio, avessi un paziente con Spongiosi, un dermatologo la classifica come presenza **lieve [1]** (β_2= -0.55434) del sintomo, un altro invece come **moderata [2]** (β_2= +0.5036), la classificazione potrebbe cambiare drasticamente (facendo sempre riferimento ai coefficienti nella tabella in figura 14).

Se, una volta utilizzato il modello in clinica, si evince che l'accuratezza è peggiorata drasticamente, si potrebbe pensare di usare solo la presenza/assenza di un sintomo per costruire il modello e valutarne poi l'accuratezza.