**Reporte Proyecto de Métodos de Inteligencia Artificial (Predictor)**

**Members:** Tania Bell Nicholls, Andrea Flores Fernández

**Objective:** Find the best possible neural model that will allow us to classify our bean data, surpassing the following metrics:

|  | Accuracy | Precision | Recall |
| --- | --- | --- | --- |
| **Training** | 0.879 | 0.86 | 0.88 |
| **Test** | 0.877 | 0.87 | 0.88 |

**Tratamiento de datos**

**a)** **Se eliminaron los datos:**

All variables were used

**b)** **Se rellenaron los datos:**

No variables were filled.

**c)** **Se modificaron datos:**

The variable that was modified was Class. It was changed to numeric variables based on the following classification:

* SEKER=0
* SIRA=1
* HOROZ=2
* CALI=3
* BARBUNYA=4
* BOMBAY=5
* DERMASON=6

**Variables usadas en el modelo:**

* Area
* Perimeter
* MajorAxisLength
* MinorAxisLength
* AspectRation
* Eccentricity
* ConvexArea
* EquivDiameter
* Extent
* Solidity
* Roundness
* Compactness
* ShapeFactor1
* ShapeFactor2
* ShapeFactor3
* ShapeFactor4
* Class

**Se crearon nuevas variables (Sí/No)**

No

**Cómo fueron creadas:**

Doesn’t apply

**Se Escalaron las variables (Sí/No):**

No

**Tipo de Escalamiento usado:**

Doesn’t apply

**Características del modelo**

**a)** **Tipo de Modelo (Perceptrón Simple o Perceptrón Multicapa)**

**b)** **Grado del polinomio usado o forma de la función o neuronas usadas en cada capa**

**Medidas de Desempeño del modelo en el entrenamiento**

| **Exactitud** | **Precisión** | **Recall** |
| --- | --- | --- |
| **0.9738** | **0.9298** | **0.9117** |

**Desempeño del modelo en la Predicción**

| **Exactitud** | **Precisión** | **Recall** |
| --- | --- | --- |
| **0 .9714** | **0.9214** | **0.9037** |

**Code:**

**clear all;**

**close all;**

**clc;**

**%% Data**

**data=xlsread('DryBeans.xlsx','Dry\_Beans\_Dataset','A2:R13612');**

**Y=[data(:,18)]; %output**

**X=[data(:,1:16)]; % inputs**

**%% Partition**

**cv=cvpartition(Y,'holdout',0.15);**

**% use load mymodelcv.mat**

**%Train Data**

**Xtrain=X(training(cv),:); % 85%**

**Ytrain=Y(training(cv));**

**% Test data**

**Xtest=X(test(cv),:); % 15%**

**Ytest=Y(test(cv));**

**%% Data Scaling**

**%Split the Ytrain in dummy variables**

**dYtrain=zeros(size(Ytrain,1),7);**

**for j=1:7**

**for i=1:size(Ytrain,1)**

**if Ytrain(i,1)==j-1**

**dYtrain(i,j)=1;**

**else**

**dYtrain(i,j)=0;**

**end**

**end**

**end**

**%Split the Ytest in dummy variables**

**dYtest=zeros(size(Ytest,1),7);**

**for j=1:7**

**for i=1:size(Ytest,1)**

**if Ytest(i,1)==j-1**

**dYtest(i,j)=1;**

**else**

**dYtest(i,j)=0;**

**end**

**end**

**end**

**%% Model creation**

**load netproyecto.mat**

**% net=feedforwardnet(10);**

**% net.trainFcn='trainrp';**

**%trainscg**

**% net=train(net,Xtrain',dYtrain'); % Training**

**%% TRAIN**

**%% Simulation**

**Yhtrain=net(Xtrain');**

**Yhtrain=round(Yhtrain);**

**Yhtrain=Yhtrain';**

**%% Correction**

**for j=1:7**

**for i=1:size(Yhtrain,1)**

**if Yhtrain(i,j)<0**

**Yhtrain(i,j)=0;**

**end**

**if Yhtrain(i,j)>1**

**Yhtrain(i,j)=1;**

**end**

**end**

**end**

**Atrain1=confusionmat(dYtrain(:,1),Yhtrain(:,1));**

**figure(1)**

**confusionchart(Atrain1)**

**Atrain2=confusionmat(dYtrain(:,2),Yhtrain(:,2));**

**figure(2)**

**confusionchart(Atrain2)**

**Atrain3=confusionmat(dYtrain(:,3),Yhtrain(:,3));**

**figure(3)**

**confusionchart(Atrain3)**

**Atrain4=confusionmat(dYtrain(:,4),Yhtrain(:,4));**

**figure(4)**

**confusionchart(Atrain4)**

**Atrain5=confusionmat(dYtrain(:,5),Yhtrain(:,5));**

**figure(5)**

**confusionchart(Atrain5)**

**Atrain6=confusionmat(dYtrain(:,6),Yhtrain(:,6));**

**figure(6)**

**confusionchart(Atrain6)**

**Atrain7=confusionmat(dYtrain(:,7),Yhtrain(:,7));**

**figure(7)**

**confusionchart(Atrain7)**

**%% Metrics**

**Accu1=sum(diag(Atrain1))/sum(sum(Atrain1));**

**Accu2=sum(diag(Atrain2))/sum(sum(Atrain2));**

**Accu3=sum(diag(Atrain3))/sum(sum(Atrain3));**

**Accu4=sum(diag(Atrain4))/sum(sum(Atrain4));**

**Accu5=sum(diag(Atrain5))/sum(sum(Atrain5));**

**Accu6=sum(diag(Atrain6))/sum(sum(Atrain6));**

**Accu7=sum(diag(Atrain7))/sum(sum(Atrain7));**

**Accu=mean([Accu1 Accu2 Accu3 Accu4 Accu5 Accu6 Accu7]);**

**%Precision**

**Pre1=Atrain1(2,2)/(Atrain1(2,2)+Atrain1(1,2));**

**Pre2=Atrain2(2,2)/(Atrain2(2,2)+Atrain2(1,2));**

**Pre3=Atrain3(2,2)/(Atrain3(2,2)+Atrain3(1,2));**

**Pre4=Atrain4(2,2)/(Atrain4(2,2)+Atrain4(1,2));**

**Pre5=Atrain5(2,2)/(Atrain5(2,2)+Atrain5(1,2));**

**Pre6=Atrain6(2,2)/(Atrain6(2,2)+Atrain6(1,2));**

**Pre7=Atrain7(2,2)/(Atrain7(2,2)+Atrain7(1,2));**

**Pre=mean([Pre1 Pre2 Pre3 Pre4 Pre5 Pre6 Pre7]);**

**%Recall**

**Rec1=Atrain1(2,2)/(Atrain1(2,2)+Atrain1(2,1));**

**Rec2=Atrain2(2,2)/(Atrain2(2,2)+Atrain2(2,1));**

**Rec3=Atrain3(2,2)/(Atrain3(2,2)+Atrain3(2,1));**

**Rec4=Atrain4(2,2)/(Atrain4(2,2)+Atrain4(2,1));**

**Rec5=Atrain5(2,2)/(Atrain5(2,2)+Atrain5(2,1));**

**Rec6=Atrain6(2,2)/(Atrain6(2,2)+Atrain6(2,1));**

**Rec7=Atrain7(2,2)/(Atrain7(2,2)+Atrain7(2,1));**

**Rec=mean([Rec1 Rec2 Rec3 Rec4 Rec5 Rec6 Rec7]);**

**[Accu Pre Rec]**

**%% TEST**

**%% Simulation**

**Yhtest=net(Xtest');**

**Yhtest=round(Yhtest);**

**Yhtest=Yhtest';**

**%% Correction**

**for j=1:7**

**for i=1:size(Yhtest,1)**

**if Yhtest(i,j)<0**

**Yhtest(i,j)=0;**

**end**

**if Yhtest(i,j)>1**

**Yhtest(i,j)=1;**

**end**

**end**

**end**

**Atest1=confusionmat(dYtest(:,1),Yhtest(:,1));**

**figure(8)**

**confusionchart(Atest1)**

**Atest2=confusionmat(dYtest(:,2),Yhtest(:,2));**

**figure(9)**

**confusionchart(Atest2)**

**Atest3=confusionmat(dYtest(:,3),Yhtest(:,3));**

**figure(10)**

**confusionchart(Atest3)**

**Atest4=confusionmat(dYtest(:,4),Yhtest(:,4));**

**figure(11)**

**confusionchart(Atest4)**

**Atest5=confusionmat(dYtest(:,5),Yhtest(:,5));**

**figure(12)**

**confusionchart(Atest5)**

**Atest6=confusionmat(dYtest(:,6),Yhtest(:,6));**

**figure(13)**

**confusionchart(Atest6)**

**Atest7=confusionmat(dYtest(:,7),Yhtest(:,7));**

**figure(14)**

**confusionchart(Atest7)**

**%% Metrics**

**Accu1=sum(diag(Atest1))/sum(sum(Atest1));**

**Accu2=sum(diag(Atest2))/sum(sum(Atest2));**

**Accu3=sum(diag(Atest3))/sum(sum(Atest3));**

**Accu4=sum(diag(Atest4))/sum(sum(Atest4));**

**Accu5=sum(diag(Atest5))/sum(sum(Atest5));**

**Accu6=sum(diag(Atest6))/sum(sum(Atest6));**

**Accu7=sum(diag(Atest7))/sum(sum(Atest7));**

**Accu=mean([Accu1 Accu2 Accu3 Accu4 Accu4 Accu6 Accu7]);**

**%Precision**

**Pre1=Atest1(2,2)/(Atest1(2,2)+Atest1(1,2));**

**Pre2=Atest2(2,2)/(Atest2(2,2)+Atest2(1,2));**

**Pre3=Atest3(2,2)/(Atest3(2,2)+Atest3(1,2));**

**Pre4=Atest4(2,2)/(Atest4(2,2)+Atest4(1,2));**

**Pre5=Atest5(2,2)/(Atest5(2,2)+Atest5(1,2));**

**Pre6=Atest6(2,2)/(Atest6(2,2)+Atest6(1,2));**

**Pre7=Atest7(2,2)/(Atest7(2,2)+Atest7(1,2));**

**Pre=mean([Pre1 Pre2 Pre3 Pre4 Pre5 Pre6 Pre7]);**

**%Recall**

**Rec1=Atest1(2,2)/(Atest1(2,2)+Atest1(2,1));**

**Rec2=Atest2(2,2)/(Atest2(2,2)+Atest2(2,1));**

**Rec3=Atest3(2,2)/(Atest3(2,2)+Atest3(2,1));**

**Rec4=Atest4(2,2)/(Atest4(2,2)+Atest4(2,1));**

**Rec5=Atest5(2,2)/(Atest5(2,2)+Atest5(2,1));**

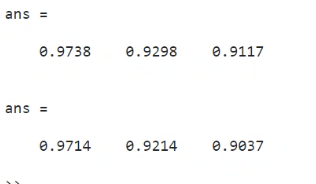
**Rec6=Atest6(2,2)/(Atest6(2,2)+Atest6(2,1));**

**Rec7=Atest7(2,2)/(Atest7(2,2)+Atest7(2,1));**

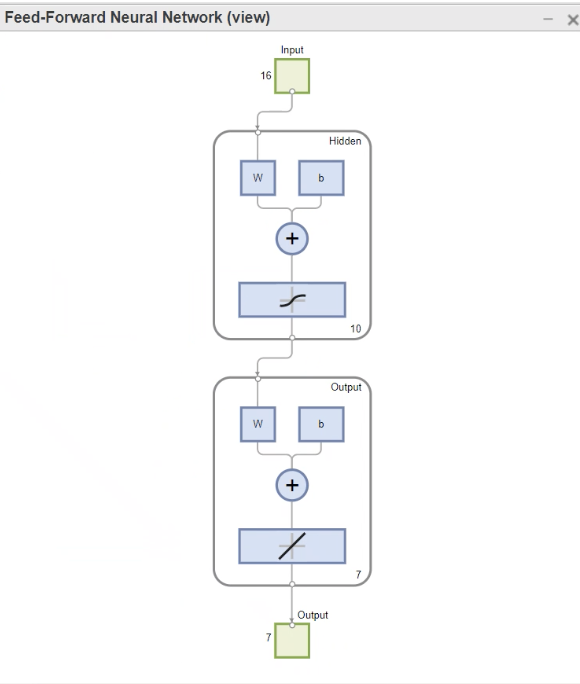
**Rec=mean([Rec1 Rec2 Rec3 Rec4 Rec5 Rec6 Rec7]);**

**[Accu Pre Rec]**

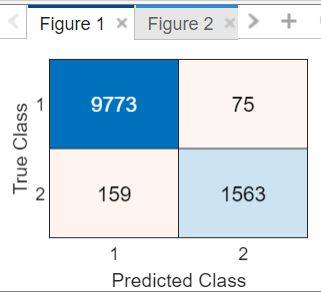
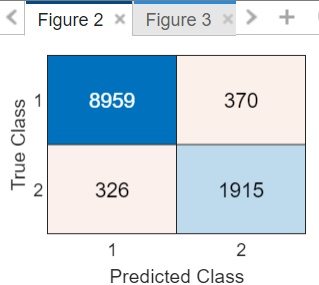
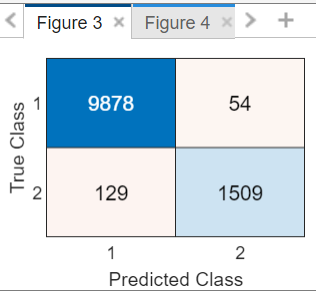
**Results:**

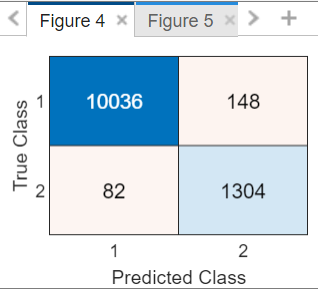
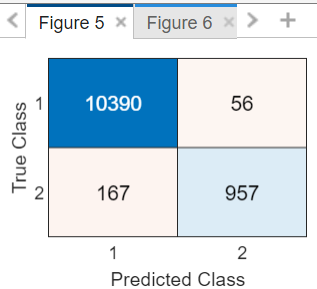
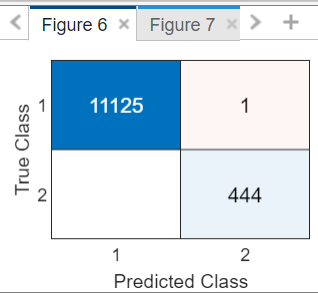
****

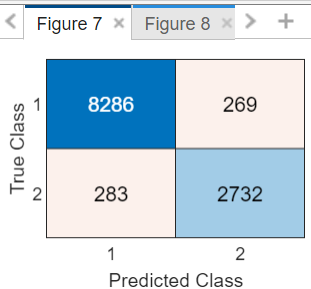
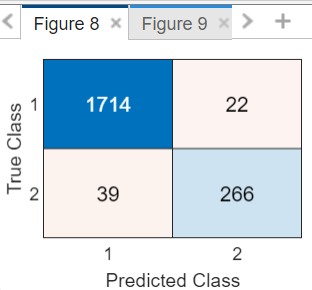
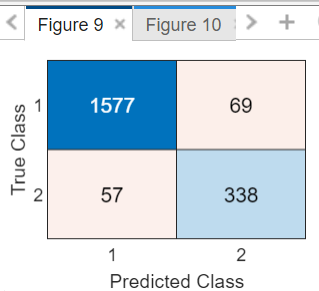
**Diagram:**

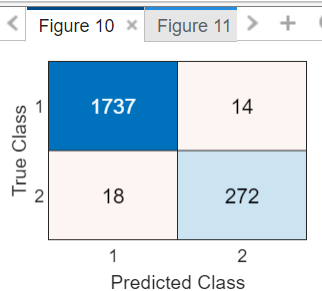
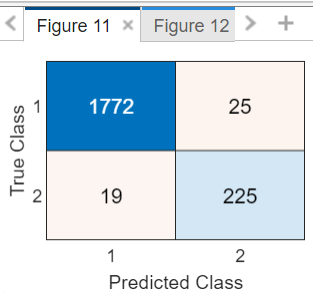
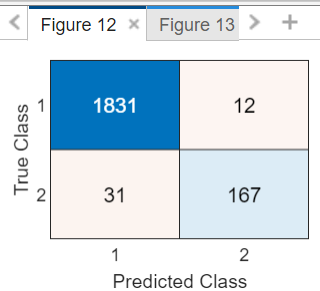
****

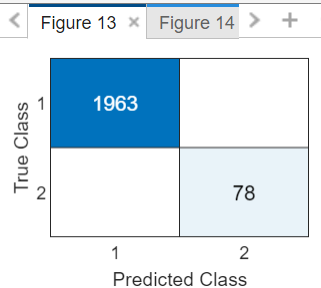
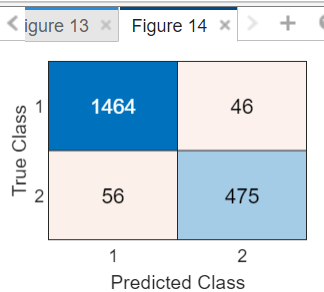
**Confusion Matrices:**

**  **

**  **

**  **

**  **

** **

**Conclusiones:**

During this project we decided to use multilayer perceptron to find the best model to classify our data, even though it might be more complex than one layer perceptron. It also allowed us to use dummy variables and make our results more precise. We faced some problems finding our model, as we didn’t apply the correction part from the beginning, but by using “unique” we were able to identify that we had non binary numbers when we were building our confusion matrices. In the end we were very surprised and satisfied with the result we obtained, as our metrics for train and test were higher than our goal.