

# INTELIGENCIA ARTIFICIAL.

*Informe de software*



## Integrantes

**Carlos Eduardo Cupul.**

**Henry Ronaldo Uicab Rodríguez.**

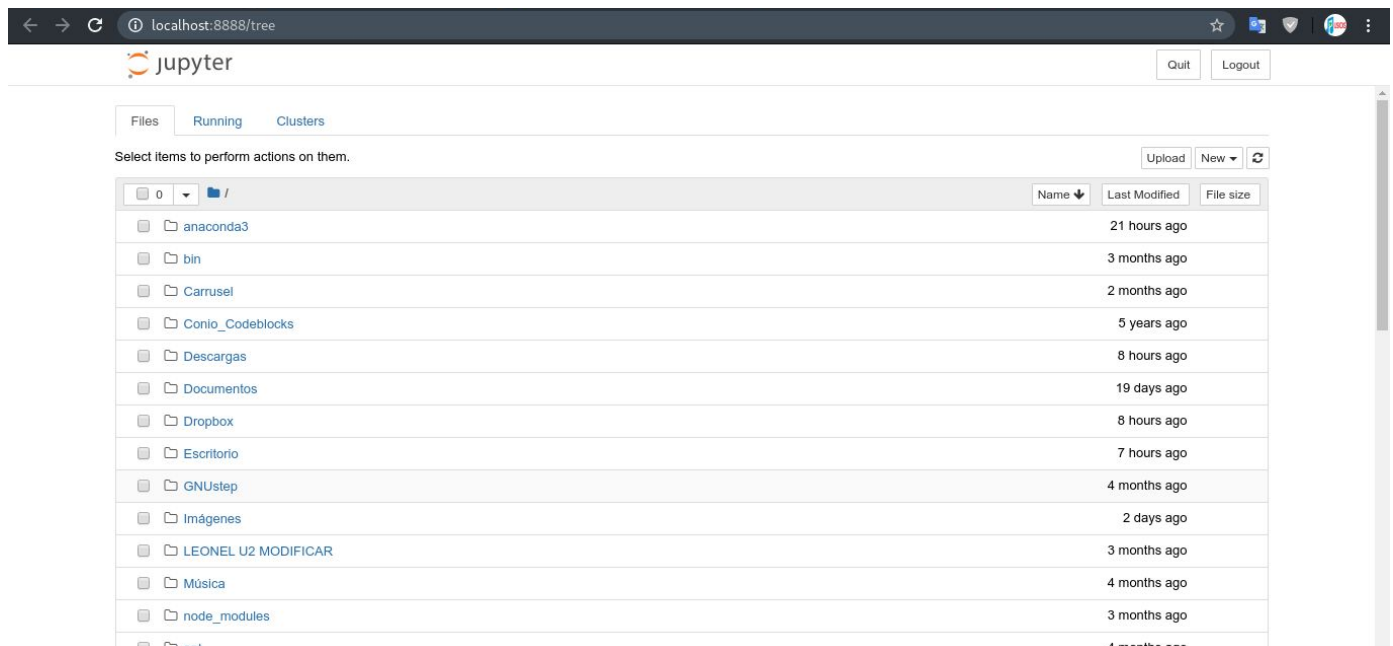
**Luis Adrián Balam Espadas.**

04/06/2020

8.º “A” ING. EN SISTEMAS COMPUTACIONALES  
INTELIGENCIA ARTIFICIAL

Predecir de forma diagnóstica si un paciente tiene diabetes o no con base a ciertas mediciones de diagnóstico incluidas en un conjunto de datos.

Primeramente se realizó la instalación de las librerías y el entorno de desarrollo como son el SKLEARN Y JÚPITER NOTEBOOK.



Para comenzar con el análisis de los datos, primero hay que saber que referente a la base de datos usado son de mujeres de al menos de 21 años de edad con datos como la presión arterial, índice de masa corporal, edad, etc.

The screenshot shows the Kaggle Data Explorer interface for the 'diabetes.csv' file (23.31 KB). The 'Summary' section indicates 9 columns: 1 Integer and 8 Decimals. The 'Compact' view of the data is shown below.

| # | Pregnanc... | # Glucose | # BloodPre... | # SkinThick... | # Insulin | # BMI |
|---|-------------|-----------|---------------|----------------|-----------|-------|
| 1 | 1           | 183       | 66            | 23             | 846       | 36.1  |
| 5 |             | 166       | 72            | 19             | 175       | 25.8  |
| 7 |             | 100       | 0             | 0              | 0         | 30    |
| 0 |             | 118       | 84            | 47             | 230       | 45.8  |
| 7 |             | 107       | 74            | 0              | 0         | 29.6  |
| 1 |             | 103       | 30            | 38             | 83        | 43.3  |
| 1 |             | 115       | 70            | 30             | 96        | 34.6  |
| 3 |             | 126       | 88            | 41             | 235       | 39.3  |
| 0 |             | 00        | 0.0           | 0              | 0         | 0.0   |

Se realizó la impresión de la cabeza de la base de datos para visualizar los datos que compone el dataset.

The screenshot shows a Jupyter Notebook window titled 'ProjectDiabetes'. The code cell contains the following Python code:

```
#Nombre del arbol de decisión
graph.write_png('decision_tree.png')
#Generación del arbol
Image(graph.create_png())
```

Below the code, a table of data is displayed:

|   | Embarazos | Glucosa | PresionArterial | EspesorPiel | Insulina | Imc  | ArbolDiabetes |
|---|-----------|---------|-----------------|-------------|----------|------|---------------|
| 1 | 6         | 148     | 72              | 35          | 0        | 33.6 | 0.627         |
| 2 | 1         | 85      | 66              | 29          | 0        | 26.6 | 0.351         |
| 3 | 8         | 183     | 64              | 0           | 0        | 23.3 | 0.672         |
| 4 | 1         | 89      | 66              | 23          | 94       | 28.1 | 0.167         |
| 5 | 0         | 137     | 40              | 35          | 168      | 43.1 | 2.288         |

Below the table, another table is shown with the headers 'Edad' and 'Salida':

|   | Edad | Salida |
|---|------|--------|
| 1 | 50   | 1      |
| 2 | 31   | 0      |
| 3 | 32   | 1      |
| 4 | 21   | 0      |
| 5 | 33   | 1      |

Se demuestra la predicción de el porcentaje de exactitud del árbol de decisión para después generar.

```
5 33 1
Porcentaje de exactitud: 0.70995670995671
Precisión Regresión Logística:
```

Se implementan tres modelos para predecir la precisión quién de los pacientes presentan Diabetes. Estos modelos son regresión logística, soporte de vectores y vecinos más cercanos.

```
Porcentaje de exactitud: 0.70995670995671
Precisión Regresión Logística:
0.7653631284916201
Precisión Soporte de Vectores:
0.7597765363128491
Precisión Vecinos más Cercanos:
0.8286778398510242
```

A través de los modelos implementados se muestran las salidas de quién de todos tienen la enfermedad de diabetes con cada uno de los modelos.

Predicción Regresión Logística:

Edad Salida

|   |    |   |
|---|----|---|
| 1 | 50 | 1 |
| 2 | 31 | 1 |
| 3 | 32 | 1 |
| 4 | 21 | 1 |
| 5 | 33 | 1 |

Predicción Soporte de Vectores:

Edad Salida

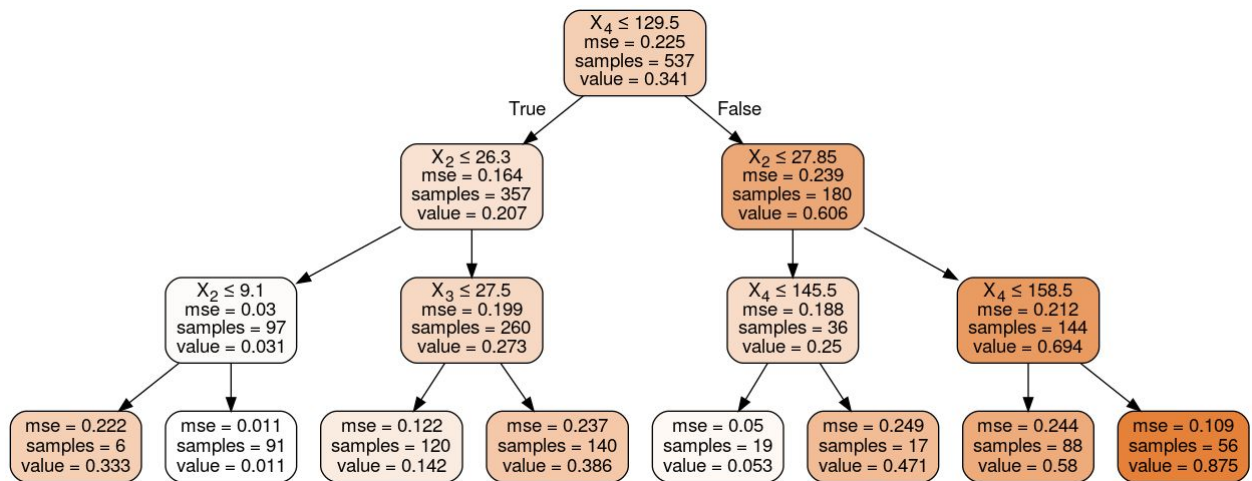
|   |    |   |
|---|----|---|
| 1 | 50 | 1 |
| 2 | 31 | 0 |
| 3 | 32 | 1 |
| 4 | 21 | 0 |
| 5 | 33 | 1 |

Predicción Vecinos más Cercanos:

Edad Salida

|   |    |   |
|---|----|---|
| 1 | 50 | 0 |
| 2 | 31 | 0 |
| 3 | 32 | 0 |
| 4 | 21 | 0 |
| 5 | 33 | 0 |

Con los modelos de predicciones se define un árbol de decisiones para el problema planteado.



## Glosario de funciones

**pima = pd.read\_csv("diabetes.csv", header=None, names=col\_names):**

Esta función permite cargar el dataset o base de datos del cual hará las predicciones.

**pima = pima.drop(0, axis=0)**

Esta función realiza la eliminación de la primera fila con el fin de evitar la compatibilidad de tipo de datos.

**print(pima.head())**

Función que permite imprimir la cabeza del dataset o base de datos.

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=1) #**

Esta función divide los datos de entrenamiento y prueba dando como prioridad la cantidad del 30% en el conjunto de pruebas.

**clf = clf.fit(X\_train,y\_train)**

Función para entrenar el árbol de decisiones.

**print("Porcentaje de exactitud:",metrics.accuracy\_score(y\_test, y\_pred))**

Función que mide el porcentaje de exactitud del árbol de decisión.

**decision\_tree.fit(X\_train, y\_train)**

Función para indicar el dataset o base de datos de entrenamiento al árbol de decisiones.

**Y\_pred = logreg.predict(X\_test)**

Función para predecir con regresión logística con base a datos de prueba.

**Y\_pred = svc.predict(X\_test)**

Función para predecir con soporte de vectores con base a datos de prueba.

**Y\_pred = knn.predict(X\_test)**

Función para predecir con vecinos más cercanos con base a datos de prueba.

**pima.drop(['Embarazos'], axis = 1, inplace=True)**

Función para eliminar la columna de embarazos

**out\_logreg = pd.DataFrame({ 'Edad' : ids, 'Salida': prediccion\_logreg })**

Funcion de regresion logistica.

**out\_svc = pd.DataFrame({ 'Edad' : ids, 'Salida': prediccion\_svc })**

Función de soporte de vectores.

```
out_knn = pd.DataFrame({ 'Edad' : ids, 'Salida': prediccion_knn })
```

Función de vecinos más cercanos.

```
export_graphviz(decision_tree, out_file=dot_data, filled=True,  
rounded=True, special_characters=True)
```

Función para darle características al árbol de decisiones.

```
graph.write_png('decision_tree.png')
```

Nombre del archivo

```
Image(graph.create_png())
```

Función para generar el árbol de decisiones.