





Instituto Tecnologico Superior De Valladolid

"2020, Año de la Benemérita madre de la Patria, Leona Vicario"

Actividad:

Software U4

Tema:

Documentación del software

Asignatura:

Inteligencia Artificial

Integrantes:

Darwin Alexis Ciau Cahun

Juan Carlos Estrella Rodriguez

Pedro Marrufo Fernandez

Jesus Enrique Balam Batun

Juan Jose Cosgaya Uh

Profesor:

Lic. Jorge Pool Cen

Ing. En Sistemas Computacionales

8° Semestre "A"

Fecha De Entrega:

Viernes 05 de junio de 2020















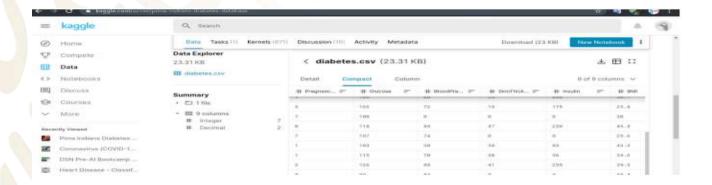


Instrucciones: Desarrollar con python y scikit learning un código que genera una toma de desición con un algoritmo de clasificación, por ejemplo el problema del Titanic que determina si sobrevive o no, puede apoyarse en los siguientes enlaces: https://www.aprendemachinelearning.com/arbol-de-decision-en-python-clasificacion-yprediccion/ El entregable es un programa documentado en un informe con un glosario de las funciones.

Como inicio, se debe realizar la instalación de las librerías necesarias para este programa y entorno o programa fundamental para el desarrollo que son Sklearn Y Júpiter notebook. (Este programa y librería se usaron en la unidad 3 en cuestión de programa Titanic)



Se le considera fundamental un análisis de lo que estamos trabajando en este caso de los datos, se puede determinar qué es lo que contiene la base de datos y que es lo que estamos usando en este caso de los pacientes con diabetes, se puede apreciar datos de edad, en este caso de mujeres de 21 años con algunos puntos como presión arterial, índice de masa corporal entre otros. Esto se puede analizar en la página Kaggle.











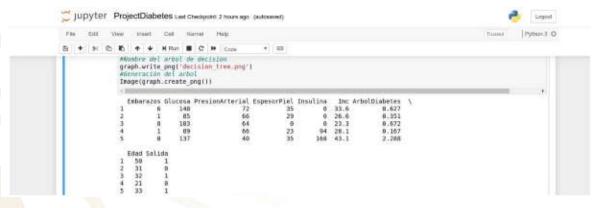








DEPARTAMENTO DE SERVICIO SOCIAL Y RESIDENCIAS PROFESIONALES En Jupyter se escribe el código de visualización de datos en este caso de la cabecera principal de la base de datos y se imprime en este caso se puede dar un análisis o visualizar, los datos componentes en el dataset.



Se visualiza la predicción del porcentaje de exactitud del árbol de decisión, en este caso se puede apreciar en la imagen que posee 0.7099.

5 33 1
Porcentaje de exactitud: 0.70995670995671
Precisión Pegresión Logística:

De aquí hay que implementar los tres tipos de precisión en este caso los modelos que se vieron con anterioridad en la unidad 3, estos modelos son necesarios en este caso del programa de pacientes con diabetes.

rorecincaje de exacercada variososoriososoria

Precisión Regresión Logística:

0.7653631284916201

Precisión Soporte de Vectores:

0.7597765363128491

Precisión Vecinos más Cercanos:

0.8286778398510242

Con esos tres modelos de precisiones implementados, se puede determinar los pacientes con el problema diabetes, se puede determinar la edad y la salida en este caso si sobrevive o no.











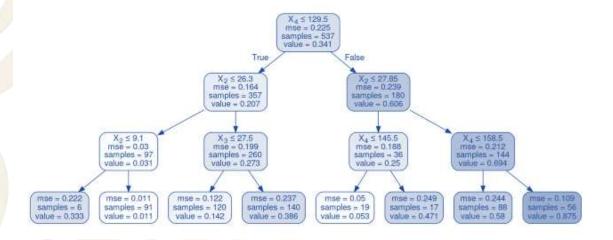






Predicción Regresión Logistica: CIAL Y RESIDENCIAS PROFESIONALES Edad Salida "2020 1 cario" Predicción Soporte de Vectores: Edad Salida Predicción Vecinos más Cercanos: Edad Salida

Gracias a estos modelos se puede determinar un árbol de decisiones para este programa.



















DEPARTAMENTO DE SERVICIO SOCIAL Y RESIDENCIAS PROFESIONALES Glosario de funciones

"2020, Año de la Benemérita madre de la Patria, Leona Vicario"

• pima = pd.read csv("diabetes.csv", header=None, names=col names):

Función que permite cargar el dataset donde se hará las predicciones.

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

Función que realiza la eliminación de la primera fila con el fin de evitar la compatibilidad de tipo de datos que se encuentren en el código.

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) #

Fu<mark>nción que divid</mark>e 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 e imprime 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

Y pred = svc.predict(X test)

Función para predecir con soporte de vectores

Y pred = knn.predict(X test)

Función para predecir con vecinos más cercanos.









