CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

APRENDIZAJE INTELIGENTE

6° "A"

TERCERA EVALUACIÓN PARCIAL

Profesor: Francisco Javier Luna Rosas

Alumnos:

Espinoza Sánchez Joel Alejandro

Gómez Garza Dariana

González Arenas Fernando Francisco

Fecha de Entrega: Aguascalientes, Ags.**, 6** de junio de 2021

Tercera Evaluación Parcial

Introducción

El aprendizaje automático es el campo de las ciencias de la computación dentro de la inteligencia artificial, que trata de desarrollar técnicas que permitan que las computadoras aprendan, donde un agente aprende cuando su desempeño mejora con la experiencia; es decir, cuando la habilidad no estaba presente en su genotipo o rasgos de nacimiento. De forma más concreta, los investigadores del aprendizaje de máquinas buscan algoritmos y heurísticas para convertir muestras de datos en programas de computadora, sin tener que escribir los últimos explícitamente. Los modelos o programas resultantes deben ser capaces de generalizar comportamientos e inferencias para un conjunto más amplio (potencialmente infinito) de datos.

El campo de actuación del aprendizaje automático se apoya con el de la estadística inferencial, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automático incorpora las preocupaciones de la complejidad computacional de los problemas pues, muchos son de clase NP-hard, por lo que gran parte de la investigación realizada en aprendizaje automático está enfocada al diseño de soluciones factibles a esos problemas. El aprendizaje automático también está estrechamente relacionado con el reconocimiento de patrones y puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos. Por lo tanto es un proceso de inducción del conocimiento.

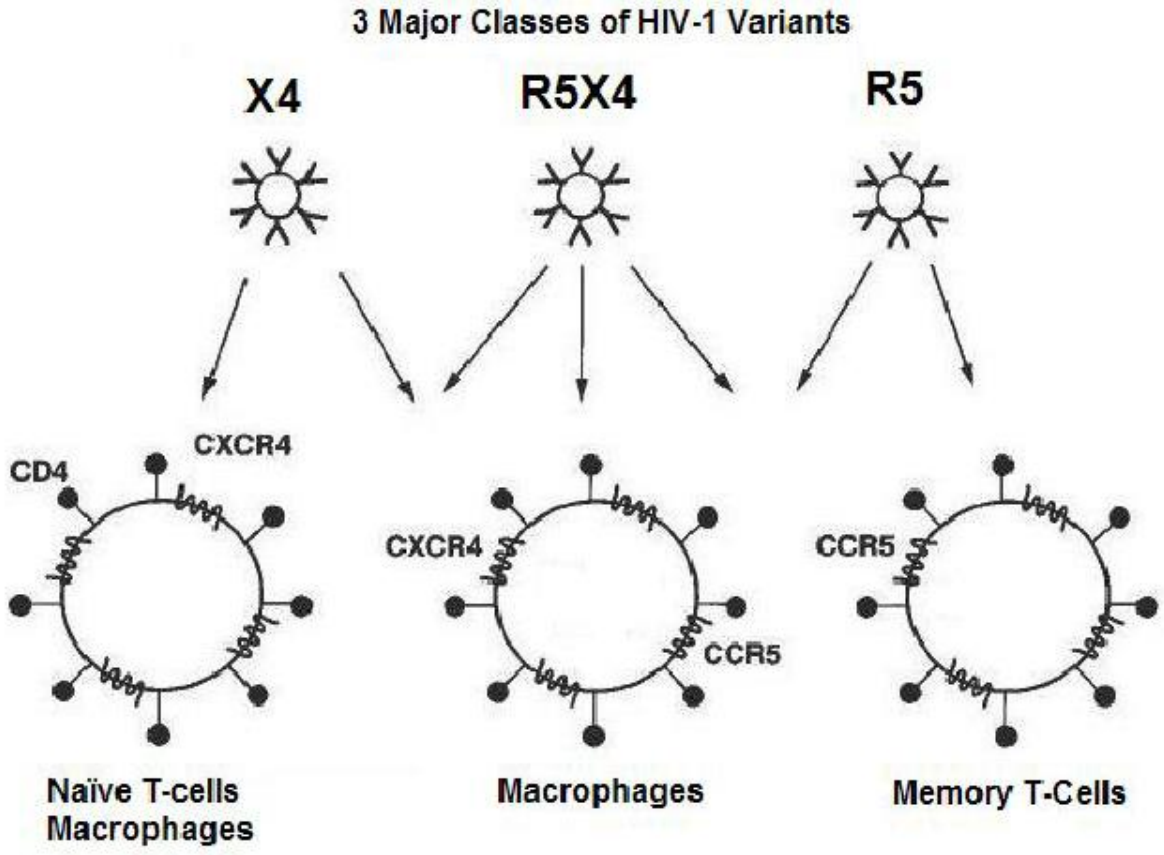
El aprendizaje automático tiene una amplia gama de aplicaciones, incluyendo motores de búsqueda, diagnósticos médicos, detección de fraude en el uso de tarjetas de crédito, análisis del mercado de valores, clasificación de secuencias de ADN, reconocimiento del habla y del lenguaje escrito, juegos y robótica.

Debido a su amplia aplicación en el reconocimiento de patrones, el aprendizaje automático puede ser usado para la predicción de fenómenos en múltiples campos. Tal es el caso de la presente investigación, donde se aplicará el aprendizaje automático dentro de la medicina.

En el campo de la medicina, la predicción de los correceptores del virus de inmunodeficiencia humana (VIH) es importante, pues existen tres variantes importantes de estos:

- R5
- X4
- R5X4

Las variantes R5 utilizan los correceptores CCR5 mientras que las variantes X4 se enlazan a los correceptores CXCR4, sin embargo, las variantes R5X4 pueden utilizar ambos correceptores pero es necesario realizar una predicción óptima ya que las variantes X4 y R5X4 son asociadas con una mayor velocidad de propagación del virus.



Para ello, a continuación se tratarán algunos modelos de aprendizaje supervisado y no supervisado y evaluar su eficiencia con respecto a este caso de estudio.

Procedimiento

Después de la extracción (véase anexo 1) y construcción del conjunto de datos, se tiene un conjunto de la siguiente forma:

Tipo	Virus	Aminoacido	PI	Negativo	Positivo	C	H	N	O	S	NumAtomos	Alifatico	Hidrofatico	Volumen	Masa
R5	AF062012	AGYAILKCNC	8.67	10	13	562	916	164	179	6	1827	86.81	-0.339	12185	13009.88
R5	AF231045	CTRPSNNTRT	8.9	2	4	165	264	52	53	2	536	66.86	-0.514	3561	3870.46
R5	U08810	VKETQMNW	8.57	73	82	3935	6199	1095	1146	35	12410	91.61	-0.243	82829	88299.9
R5	U51296	acaccagggc c	5.14	0	0	2009	3332	688	830	157	7016	28.2	0.821	49021	55424.32
R5	AF407161	MRAMGIQM	8.63	81	92	4289	6764	1208	1245	42	13548	91.86	-0.242	90473	96515.5
R5	AB253429	FFRENLAFQC	8.41	130	136	5146	8088	1376	1492	24	16126	85.64	-0.473	107235	113869.3
R5	U08645	IQIRSENITNN	10.4	7	18	460	767	143	131	2	1503	92.17	-0.483	9944	10443.42
R5	U08647	IQIRTENITNN	10.6	5	17	456	759	143	128	2	1488	93.37	-0.438	9853	10339.34
R5	U08795	MGSKWSKSS	6.27	11	10	433	677	127	131	3	1371	68.64	-0.591	9111	9837.78
R5	AB253429	MDPVDPSLE	9.28	10	19	489	778	150	149	9	1575	33.76	-1.325	10510	11414
R5	AY288084	MILGIIIICNA	8.35	70	76	3691	5829	1027	1096	33	11676	90.47	-0.254	77928	83178.92
R5	AF307753	SENITNNAKN	7.82	4	5	328	521	99	104	2	1054	79.13	-0.441	7021	7561.72
R5	AF411964	MEQAPADQC	8.05	11	12	498	775	157	141	2	1573	83.33	-0.739	10462	11264.7
R5	U08823	NLTNNAKIIIV	8.76	11	14	568	898	160	171	4	1801	79.56	-0.436	11998	12814.9
R5	AF411965	MEQPPEDQC	7.02	13	13	511	788	158	141	3	1601	79.17	-0.875	10653	11480.69
R5	U92051	MRVMGIQRN	8.48	84	92	4274	6713	1177	1230	40	13434	91.57	-0.24	89686	95546.15
R5	AF255218	atggaaaact tt	5.21	0	0	2197	3712	684	945	108	7646	29.39	0.59	52455	58278.64
R5	AY010759	FTNNAKTIIVC	9.86	6	13	451	726	134	134	3	1448	79.34	-0.587	9624	10248.7
R5	AY010804	FSDNTKIIIVC	9.57	8	13	445	718	134	136	2	1435	81.43	-0.599	9534	10167.78
R5	U08670	VIIRSENITDN	10.22	6	14	461	752	142	131	3	1489	95.33	-0.469	9894	10459.06
R5	U08798	MRVKEMRKL	8.76	51	62	2666	4226	748	802	33	8475	78.66	-0.349	56662	60640.33

El conjunto de datos se construyó con las siguientes características:

- Tipo
- Virus
- Aminoácido
- Punto Isoeléctrico
- Negatividad
- Positividad
- Átomos de Carbono
- Átomos de Hidrógeno
- Átomos de Nitrógeno
- Átomos de Oxígeno
- Átomos de Azufre
- Número Atómico
- Índice Alifático
- Coeficiente Hidrofático
- Volumen
- Masa
- Escala de pH
- Área
- BStrand
- Turn
- Peso Molar

Se aplicarán tres algoritmos de aprendizaje no supervisado y tres algoritmos de aprendizaje supervisado:

- K-Means
- Clustering Jerárquico
- Análisis de Componentes Principales
- Support Vector Machine
- Naive Bayes
- Random Forest

Obtención y Procesamiento de Datos

El procesamiento del algoritmo K-Means es el siguiente:

```
#%% Ejecución de código
import pandas as pd
#import seaborn as sns
import sklearn.cluster as cluster

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAnS.csv')

#sns.pairplot(df[['Masa', 'HPScale', 'Surface', 'Alpha', 'BStrand', 'Turn', 'MolWeight']])

kmeans = cluster.KMeans(n_clusters = 2)

kmeans = kmeans.fit(df[['PI', 'Negativo', 'Positivo', 'C', 'H', 'N', 'O', 'S', 'NumAtomos', 'Alifatico',
                        'Hidrofatico', 'Volumen', 'Masa', 'HPScale', 'Surface', 'Alpha', 'BStrand',
                        'Turn', 'MolWeight']])

print(kmeans.cluster_centers_)

df['Clusters'] = kmeans.labels_

df['Clusters'].value_counts()

print(df.head)

df.to_csv('TablaAS.csv', index = False)
```

El procesamiento del algoritmo Clustering Jerárquico es el siguiente:

```
#%% Ejecución de código
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import AgglomerativeClustering

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAnS.csv')

X = df.iloc[:,[3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]].values

#dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))

# Acomodar CJ al dataset

CJ = AgglomerativeClustering(n_clusters = 2, affinity = 'euclidean', linkage = 'ward')
y_CJ = CJ.fit_predict(X)

# Visualizar los clusters
plt.scatter(X[y_CJ == 0,0], X[y_CJ == 0,1], s = 100, c = 'red', label = 'Clúster 1')
plt.scatter(X[y_CJ == 1,0], X[y_CJ == 1,1], s = 100, c = 'green', label = 'Clúster 2')
plt.title('CJ')
plt.legend()
plt.show()
```

El procesamiento del algoritmo PCA es el siguiente:

```
#%% Ejecución de código
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAS.csv')

features = ['PI', 'Negativo', 'Positivo', 'C', 'H', 'N', 'O', 'S', 'NumAtomos', 'Alifatico',
            'Hidrofatico', 'Volumen', 'Masa', 'HPScale', 'Surface', 'Alpha', 'BStrand',
            'Turn', 'MolWeight']

# Separando features
x = df.loc[:,features].values

# Separando el objetivo
y = df.loc[:,'Clusters'].values

# Estandarizando features
x = StandardScaler().fit_transform(x)

# Proyección PCA
pca = PCA(n_components = 2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents, columns = ['PC1', 'PC2'])
finalDf = pd.concat([principalDf, df[['Clusters']]], axis = 1)

# Visualizar Proyección 2D
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('PC1')
ax.set_ylabel('PC2')
ax.set_title('PCA')

targets = [0, 1]
colors = ['r', 'g']

for target, color in zip(targets, colors):
    index = finalDf['Clusters'] == target
    ax.scatter(finalDf.loc[index, 'PC1'], finalDf.loc[index, 'PC2'], c = color, s = 50)
ax.grid
```

El procesamiento del algoritmo Support Vector Machine es el siguiente:

```
#%% Ejecución de código
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAS.csv')

X = df.drop(['Tipo', 'Virus', 'Aminoacido', 'Clusters'], axis = 1)
Y = df['Clusters']

xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size = 0.20)

classifier = SVC(kernel = 'linear')
classifier.fit(xtrain, ytrain)

ypred = classifier.predict(xtest)
print(ypred)

print(classification_report(ytest, ypred))
```

El procesamiento del algoritmo Naive Bayes es el siguiente:

```
#%% Ejecución de código
import random
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import precision_score
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import GaussianNB

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAS.csv')

X = df.drop(['Tipo', 'Virus', 'Aminoacido', 'Clusters'], axis = 1)
Y = df['Clusters']

xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size = 0.20)

classifier = GaussianNB()
classifier.fit(xtrain, ytrain)

ypred = classifier.predict(xtest)

matriz = confusion_matrix(ytest, ypred)
print('Matriz de Confusión')
print(matriz)

precision = precision_score(ytest, ypred)
print('Precisión del modelo')
print(precision)
```

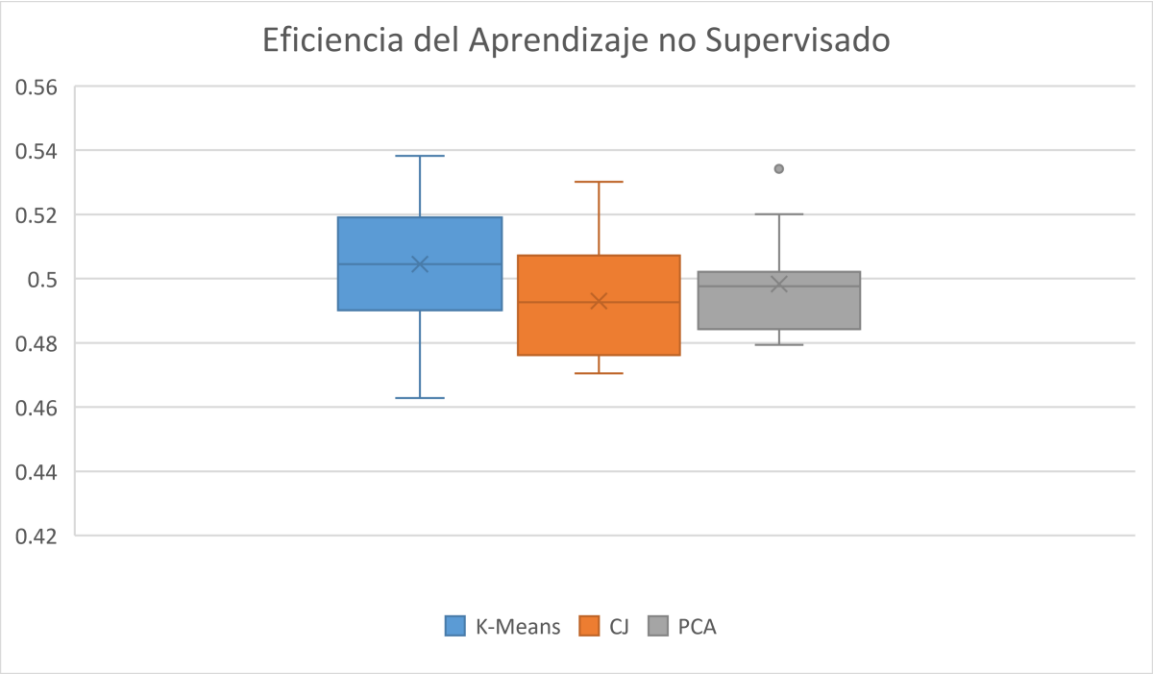
El procesamiento del algoritmo Random Forest es el siguiente:

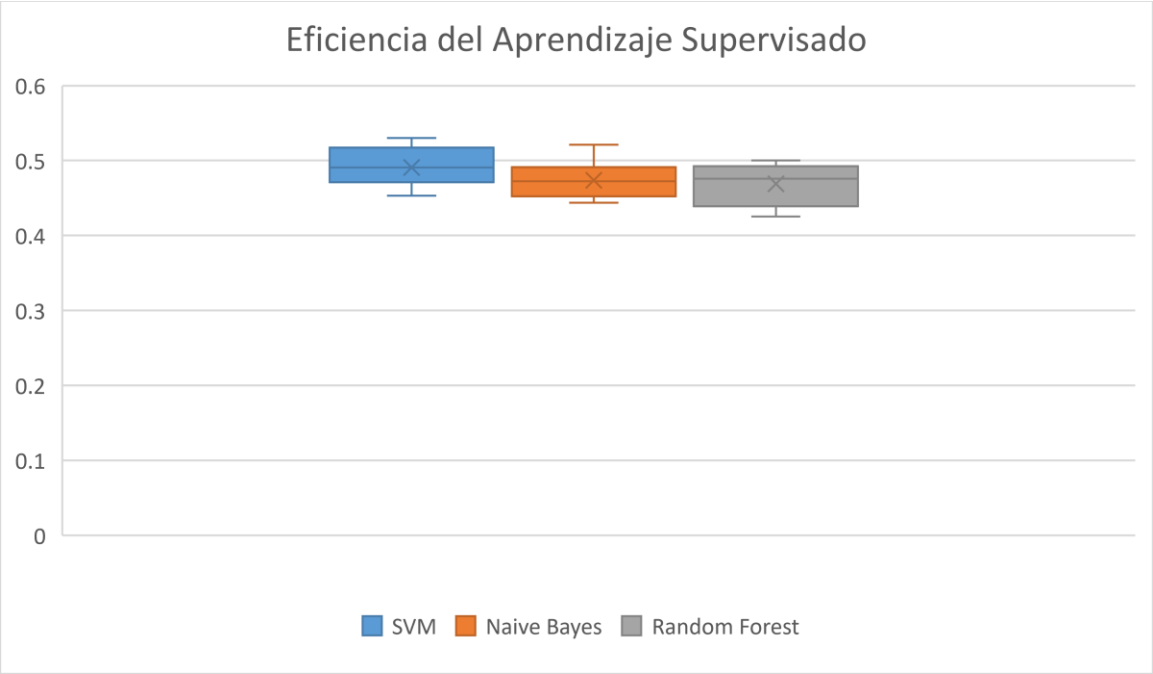
```
#%% Ejecución de código
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import precision_score
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAS.csv')

X = df.drop(['Tipo', 'Virus', 'Aminoacido', 'Clusters'], axis = 1)
Y = df['Clusters']

xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size = 0.25)

sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.fit_transform(xtest)

classifier = RandomForestClassifier(n_estimators = 4)
classifier.fit(xtrain, ytrain)

ypred = classifier.predict(xtest)

matriz = confusion_matrix(ytest, ypred)
print('Matriz de Confusión')
print(matriz)

precision = precision_score(ytest, ypred)
print('Precisión del modelo')
print(precision)
```

De la misma manera, se analizaron los algoritmos para observar su eficiencia al problema y puede observarse su eficiencia en los cuadros y gráficos a continuación:

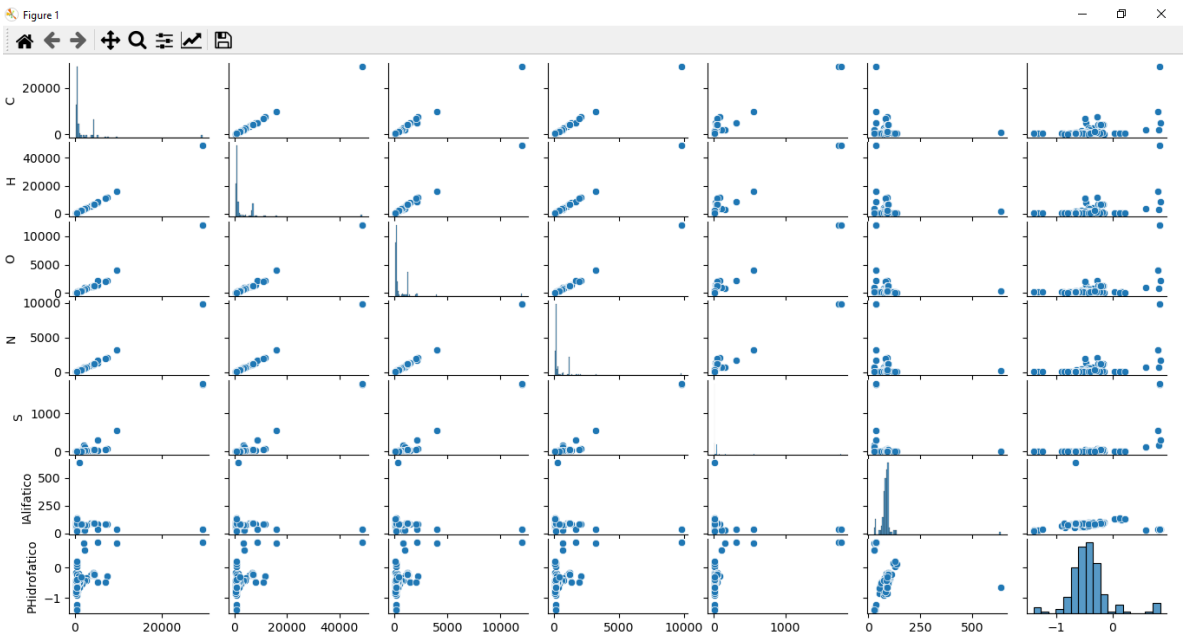
Ejecución	K-Means	CJ	PCA
1	0.4896	0.4705	0.4923
2	0.4901	0.4902	0.5021
3	0.5191	0.5123	0.4842
4	0.4969	0.4926	0.5003
5	0.4628	0.5072	0.5201
6	0.5382	0.4759	0.5342
7	0.5102	0.5301	0.4794
8	0.5294	0.4821	0.4833
9	0.4989	0.4935	0.4898
10	0.5101	0.4762	0.4976
Total	0.50453	0.49306	0.49834

Ejecución	SVM	Naive Bayes	Random Forest
1	0.4532	0.4802	0.4827
2	0.4712	0.4912	0.4939
3	0.4921	0.4672	0.4685
4	0.5173	0.4525	0.4389
5	0.5027	0.4912	0.4757
6	0.5201	0.4483	0.5002
7	0.4893	0.4723	0.4927
8	0.5302	0.5211	0.4265
9	0.4793	0.4439	0.4833
10	0.4534	0.4692	0.4256
Total	0.49088	0.47371	0.4688


Eficiencia del Aprendizaje no Supervisado
0.56
0.54
0.52
0.5
0.48
0.46
0.44
0.42
K-Means
CJ
PCA


Eficiencia del Aprendizaje Supervisado
0.6
0.5
0.4
0.3
0.2
0.1
0
SVM
Naive Bayes
Random Forest

La eficiencia del algoritmo es muy baja debido a que no existen propiedades y características que no son altamente visibles entre los dos grupos de clasificación, pues puede observarse a continuación la gráfica a par de propiedad a propiedad donde no puede observarse dos nubes de puntos definidas:



Aún cuando se encontraron algunas propiedades como las siguientes:

- Usar Negativo, Positivo ya da un avance de separación
- Usar Negativo, Positivo, PI da otro avance de separación
- Alifático e Hidrofático no son útiles
- NumAtomos y PI no son útiles
- NumAtomos y Negativo no son útiles
- 'C', 'H', 'N', 'O', 'S' no son útiles
- PI presenta un sesgo débil de separación
- Negativo no presenta sesgos para separar claramente
- Volumen, masa (por separados) no son útiles

Aclaración de Características

La construcción del dataset del aprendizaje no supervisado es el ya mostrado previamente:

Tipo	Virus	Aminoacido	PI	Negativo	Positivo	C	H	N	O	S	NumAtomos	Alifatico	Hidrofatico	Volumen	Masa	HPScale	Surface	Alpha	BStrand	Turn
R5	AF062012	AGYAILKCNC	8.67	10	13	562	916	164	179	6	1827	86.81	-0.339	12185	13009.88	-40.4	85.94	112.18	116.82	
R5	AF231045	CTRPSNNTRT	8.9	2	4	165	264	52	53	2	536	66.86	-0.514	3561	3870.46	-18	25.43	29.94	35.05	
R5	U08810	VKETQMNW	8.57	73	82	3935	6199	1095	1146	35	12410	91.61	-0.243	82829	88299.9	-189.7	569.93	775.1	777.45	
R5	U51296	acaccagggc c	5.14	0	0	2009	3332	688	830	157	7016	28.2	0.821	49021	55424.32	564.6	526.25	568.6	638.13	
R5	AF407161	MRAMGIQM	8.63	81	92	4289	6764	1208	1245	42	13548	91.86	-0.242	90473	96515.5	-205.2	620.69	853.91	846.91	
R5	AB253429	FFRENLAFQC	8.41	130	136	5146	8088	1376	1492	24	16126	85.64	-0.473	107235	113869.3	-474.8	717.84	1026.11	966.93	
R5	U08645	IQIRSENITNN	10.4	7	18	460	767	143	131	2	1503	92.17	-0.483	9944	10443.42	-44.4	65.8	93.42	92.48	
R5	U08647	IQIRTENITNN	10.6	5	17	456	759	143	128	2	1488	93.37	-0.438	9853	10339.34	-40.3	65.86	94.2	90.71	
R5	U08795	MGSKWSKSS	6.27	11	10	433	677	127	131	3	1371	68.64	-0.591	9111	9837.78	-52	63.04	86.43	82.65	
R5	AB253429	MDPVDPSLE	9.28	10	19	489	778	150	149	9	1575	33.76	-1.325	10510	11414	-133.8	69.64	90.47	86.44	
R5	AY288084	MILGIIIICNA/	8.35	70	76	3691	5829	1027	1096	33	11676	90.47	-0.254	77928	83178.92	-187.3	537.62	731.81	741.33	
R5	AF307753	SENITNNAKI	7.82	4	5	328	521	99	104	2	1054	79.13	-0.441	7021	7561.72	-30.4	49.88	63.72	67.42	
R5	AF411964	MEQAPADQ(	8.05	11	12	498	775	157	141	2	1573	83.33	-0.739	10462	11264.7	-70.9	69.64	101.27	90.27	
R5	U08823	NLTNNAKIIIV	8.76	11	14	568	898	160	171	4	1801	79.56	-0.436	11998	12814.9	-49.7	82.25	110.96	110.2	
R5	AF411965	MEQPPEDQC	7.02	13	13	511	788	158	141	3	1601	79.17	-0.875	10653	11480.69	-84	69.65	99.5	90.65	
R5	U92051	MRVMGIQRM	8.48	84	92	4274	6713	1177	1230	40	13434	91.57	-0.24	89686	95546.15	-201.3	613.37	846.63	835.75	
R5	AF255218	atggaaaact tt	5.21	0	0	2197	3712	684	945	108	7646	29.39	0.59	52455	58278.64	403.8	511.82	601.74	666.82	
R5	AY010759	FTNNAKTIIV	9.86	6	13	451	726	134	134	3	1448	79.34	-0.587	9624	10248.7	-53.4	64.86	87.95	86.5	
R5	AY010804	FSDNTKIIIVC	9.57	8	13	445	718	134	136	2	1435	81.43	-0.599	9534	10167.78	-54.5	64.84	88.07	86.46	
R5	U08670	VIIRSENITDN	10.22	6	14	461	752	142	131	3	1489	95.33	-0.469	9894	10459.06	-42.2	65.22	91.45	90.85	
R5	U08798	MRVKEMRKL	8.76	51	62	2666	4226	748	802	33	8475	78.66	-0.349	56662	60640.33	-189.8	393.57	527.26	528.42	
R5	U08710	IVIRSENFTDN	9.56	9	14	464	750	136	136	2	1488	92.2	-0.458	9880	10456.26	-41.7	65.59	93.52	90.1	
R5	M26727	MEQAPEDQC	6.07	14	12	512	791	151	148	2	1604	84.38	-0.717	10659	11476.95	-68.8	69.54	103.49	92.88	
R5	AJ418532	MKAKGIRKN	9	132	167	7435	11778	2090	2165	75	23543	88.86	-0.275	157192	167497.73	-408.1	1079.35	1479.5	1471.05	
R5	AJ418479	MRAKEIRKN'	8.48	85	93	4348	6846	1204	1254	39	13691	92.81	-0.217	91348	97298.58	-185.9	626.31	864.19	853.78	
R5	AJ418495	MRAKETRKN	9.07	78	98	4348	6862	1214	1254	40	13718	89.75	-0.253	91525	97487.54	-216.6	625.24	863.99	857.22	
R5	AJ418514	MRVKGIRKN	8.91	87	104	4332	6874	1226	1251	41	13724	93.15	-0.25	91544	97458.12	-213.9	624.63	863.78	856.83	
R5	AJ418521	MKVKGMRKI	8.75	81	93	4322	6810	1196	1244	37	13609	93.24	-0.186	90752	96613.07	-158.7	624.51	855.57	857.6	
R5	U23487	MESRWQVM	9.81	16	29	1006	1560	292	272	7	3137	76.67	-0.654	20882	22307.62	-125.6	138.35	191.22	187.41	
R5	U04900	GIRSKNFTDN	9.35	7	11	545	863	165	162	4	1739	80.98	-0.356	11593	12430.14	-39.9	81.65	106.15	109.5	
R5	AF022258	EEVVIRSENF	7.93	13	14	636	1001	185	201	5	2028	73.26	-0.551	13535	14598.49	-71.1	92.73	124.23	126.34	
R5	AF258957	LAEEEVVIRSE	8.55	12	14	556	892	160	172	3	1783	83.63	-0.43	11858	12648.69	-48.6	81.28	110.67	110.38	
R5	AF021477	CTRPNNNTR	10.03	4	11	318	517	97	93	2	1027	75.87	-0.754	6807	7233.48	-47.5	44.82	61.17	61.43	

El añadido para el aprendizaje supervisado es una columna extra de la predicción del aprendizaje no supervisado:

ht	Clusters
.81	0
.35	0
.25	0
.09	0
.75	0
.59	0
..15	0
'.04	0
.09	0
..01	0
.09	0
.47	0
..72	0
.59	0
'.03	0
.86	0

Conclusiones

Joel Alejandro Espinoza Sánchez: Gracias al examen, nos permitimos explorar el planteamiento de un problema de aprendizaje desde la selección de los algoritmos, ya que no todos los algoritmos están diseñados para la solución de una misma tarea. Su diversidad permite la solución de problemas orientados a distintos tipos de datos, predicciones y objetivos.

Pude orientar personalmente un enfoque de investigación para analizar el mejor algoritmo con base en lo que deseábamos predecir y el conjunto de datos con lo que realizaríamos el procedimiento.

Dariana Gómez Garza: En este examen final pudimos agrandar el conocimiento que teníamos en aprendizaje supervisado y no supervisado, ya que nunca habíamos hecho la construcción desde cero del dataset.

Es interesante la amplia gama de opciones que hay para realizar este tipo de programas, por ejemplo, encontramos mucha información sobre tipos de algoritmos y cómo era más sencillo clasificarlo así en lugar de realizarlo con el aprendizaje automático; pero el punto de este último parcial era tratar de realizarlo con un óptimo dataset

Me gustó como quedó nuestro examen final pero tiene muchos puntos de mejora sobre todo en el dataset.

Fernando Francisco González Arenas: La predicción de variantes del VIH por medio de aprendizaje automático es una técnica muy útil que tiene múltiples usos en la medicina alrededor del mundo en la actualidad, se puede usar para fines de seguridad y gran aplicación dentro del rastreo de enfermedad y el progreso del VIH.

Con la realización de esta práctica investigamos formas de hacer un dataset útil para detectar el correceptor de una variante del virus, lo cual puede tener muchas aplicaciones y ventajas prácticas en el futuro, incluso para futuros proyectos de los integrantes de este mismo equipo.

Referencias

- Expasy. (2021). *Swiss Bioinformatics Resource Portal*. Junio 4, 2021, de Swiss Institute of Bioinformatics Sitio web: https://www.expasy.org/
- Igual, L. (2017). *Introduction to Data Science*. Barcelona: Springer.
- National Library of Medicine. (2021). *NCBI Home*. Junio 4, 2021, de National Center for Biotechnology Investigation Sitio web: https://www.ncbi.nlm.nih.gov/
- Rebala, G. (2019). *An Introduction to Machine Learning*. California: Springer.
- Sarkar, D. (2018). *Practical Machine Learning with Python*. Chicago: Apress.
- Unpingco, J. (2019). *Python for Probability, Statistics and Machine Learning*. California: Springer.

Anexos

Anexo 1: Extracción de datos para el conjunto de análisis.

Primero se realizó una tabla estándar que después se adaptaría para aprendizaje supervisado y no supervisado. Se tomó la tabla 1 del artículo *Prediction of R5, X4 and R5X4 HIV-1 Coreceptor Usage with evolved neural networks* para comenzar con nuestro registro. La tabla es la siguiente:

R5X4 (D-tropic)	R5		X4
AB014795	AF062012	U08716	AB014785
AF062029	L03698	U39259	AB014791
AF062031	AF231045	AF204137	AB014796
AF062033	AY669778	M38429	AB014810
AF107771	U08810	U27443	U48267
U08680	U51296	U79719	U08666
U08682	AF407161	U04909	AF069672
U08444	AB253421	U04918	AF355319
U08445	U08645	U04908	AF355336
AF355674	U08647	U08450	M14100
AF355647	U08795	AF112542	A04321
AF355630	AB253429	M63929	X01762
AF355690	AY288084	U66221	L31963
M91819	AF307753	AF491737	U08447
AF035532	AF411964	U08779	AF355660
AF035533	U08823	L22084	AF355748
AF259019	AF411965	U27413	AF355742
AF259025	U92051	AF005495	AF355706
AF259021	AF355318	U52953	AF180915
AF259041	AY010759	AF321523	AF180903
AF258970	AY010804	L22940	AF035534
AF258978	AY010852	U45485	AF259050
AF021607	U08670	AB023804	AF258981
AF204137	U08798	U08453	AF259003
AF112925	AY669715	AF307755	AF021618
M17451	U08710	AF307750	AF128989
K02007	U16217	AY043176	M17449
U39362	M26727	AY158534,	AF075720
AF069140	AJ418532	AX455917	U48207
AF458235	AJ418479	AY043173	U72495
AF005494	AJ418495	AF307757	AY189526
	AJ418514	U08803	AF034375
	AJ418521	U88824	AF034376
	U23487	U69657	U27408
	U04900	AF355326	AF411966
	AF022258	U88826	U27399
	AF258957	U08368	U08822
	AF021477	U27426	U08738
		AJ006022	U08740
			U08193
			AF355330

Nuestra tabla de registros con estos datos tiene la siguiente apariencia:

	A	B	C
1	Virus	Número de acceso	
2	R5	AF062012	
3	R5	L03698	
4	R5	AF231045	
5	R5	AY669778	
6	R5	U08810	
7	R5	U51296	
8	R5	AF407161	
9	R5	AB253429	
10	R5	U08645	
11	R5	U08647	
12	R5	U08795	
13	R5	AB253429	
14	R5	AY288084	
15	R5	AF307753	
16	R5	AF411964	
17	R5	U08823	
18	R5	AF411965	
19	R5	U92051	
20	R5	AF255218	
21	R5	AY010759	
22	R5	AY010804	
23	R5	A010852	
24	R5	U08670	
25	R5	U08798	
26	R5	AY669715	

Se le agregaron columnas adicionales al entrar al sitio web del Centro Nacional para la Investigación en Biotecnología (en inglés, NCBI) donde se toma este número de acceso de cada tipo de virus y al ingresarlo dentro del sitio web del

NCBI se muestra un desglose completo de dicho tipo de virus como se muestra a continuación:

```
LOCUS       AF062012                 357 bp    DNA     linear   VRL 26-JUL-2016
DEFINITION  HIV-1 isolate KH.002 from Cambodia, envelope glycoprotein V3 region
            (env) gene, partial cds.
ACCESSION   AF062012
VERSION     AF062012.1
KEYWORDS    .
SOURCE      Human immunodeficiency virus 1 (HIV-1)
  ORGANISM  Human immunodeficiency virus 1
            Viruses; Riboviria; Pararnavirae; Artverviricota; Revtraviricetes;
            Ortervirales; Retroviridae; Orthoretrovirinae; Lentivirus.
REFERENCE   1  (bases 1 to 357)
  AUTHORS   Menu,E., Reynes,J.M., Muller-Trutwin,M.C., Guillemot,L.,
            Versmisse,P., Chiron,M., An,S., Trouplin,V., Charneau,P.,
            Fleury,H., Barre-Sinoussi,F. and Sainte Marie,F.F.
  TITLE     Predominance of CCR5-dependent HIV-1 subtype E isolates in Cambodia
  JOURNAL   J. Acquir. Immune Defic. Syndr. Hum. Retrovirol. 20 (5), 481-487
            (1999)
   PUBMED   10225231
REFERENCE   2  (bases 1 to 357)
  AUTHORS   Menu,E., Reynes,J.-M., Muller-Trutwin,M.C., Guillemot,L.,
            Versmisse,P., Chinon,M., Lay,K.S., Truplin,V., Charneau,P.,
            Fleury,H., Barre-Sinoussi,F. and Flye Sainte Marie,F.
  TITLE     Direct Submission
  JOURNAL   Submitted (27-APR-1998) Departement de Virologie, Unite de Biologie
            des Retrovirus, Institut Pasteur, 25 Rue du Dr Roux, Paris 75015,
            France
FEATURES             Location/Qualifiers
     source          1..357
                     /organism="Human immunodeficiency virus 1"
                     /proviral
                     /mol_type="genomic DNA"
                     /isolate="KH.002"
                     /db_xref="taxon:11676"
                     /country="Cambodia"
     gene            <1..>357
                     /gene="env"
     CDS             <1..>357
                     /gene="env"
                     /note="V3 region"
                     /codon_start=1
                     /product="envelope glycoprotein"
                     /protein_id="AAD27934.1"
                     /translation="AGYAILKCNDKNFNGTGPCKNVSSVQCTHGIKPVVSTQLLLNGS
                     LAEEEIIIRSENLTNNAKTIIVHLNKSVEINCTRPSNNTRTSITMGPGQVFYRTGDII
                     GDIRKAYCEINGTKWNE"
ORIGIN
        1 gctggttatg cgattttaaa gtgtaatgat aagaatttca atgggacagg gccatgtaaa
       61 aatgtcagct cagtacaatg cacacatgga attaagccag tggtatcaac tcaattgctg
      121 ttaaatggca gtctagcaga agaagagata ataatcagat ctgaaaatct cacaaacaat
      181 gccaaaacca taatagtgca ccttaataaa tctgtagaaa tcaattgtac cagaccctct
      241 aacaatacaa gaacgagtat aactatggga ccaggacaag tattctatag aacaggagac
      301 ataataggag atataagaaa agcatattgt gagattaatg gaacaaaatg gaatgaa
//
```

El atributo translation se llevará a otra base de datos llamada Expasy y en su enlace, https://web.expasy.org/protparam/, donde se ingresará la secuencia de aminoácidos que se extrajo de la base de datos pasada.

Anexo 2: Código de K-Means orientado al VIH en Python.

```
#%% Presentación
'''
   Universidad Autónoma de Aguascalientes

         Centro de Ciencias Básicas
 Departamento de Ciencias de la Computación
          Aprendizaje Inteligente
                   6° "A"

         Tercera Evaluación Parcial

   Profesor: Francisco Javier Luna Rosas
  Alumnos:
      Espinoza Sánchez Joel Alejandro
      Gómez Garza Dariana
      González Arenas Fernando Francisco

  Fecha de Entrega: 6 de junio del 2021

Descripción: KMeans orientado al VIH
'''

#%% Ejecución de código
import pandas as pd
#import seaborn as sns
import sklearn.cluster as cluster

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAnS.csv')

#sns.pairplot(df[['Masa', 'HPScale', 'Surface', 'Alpha', 'BStrand',
'Turn', 'MolWeight']])

kmeans = cluster.KMeans(n_clusters = 2)

kmeans = kmeans.fit(df[['PI', 'Negativo', 'Positivo', 'C', 'H', 'N',
'O', 'S', 'NumAtomos', 'Alifatico',
                        'Hidrofatico',      'Volumen',      'Masa',
'HPScale', 'Surface', 'Alpha', 'BStrand',
                        'Turn', 'MolWeight']])
```

```
print(kmeans.cluster_centers_)

df['Clusters'] = kmeans.labels_

df['Clusters'].value_counts()

print(df.head)

df.to_csv('TablaAS.csv', index = False)
```

Anexo 3: Código de CJ orientado al VIH en Python.

```
#%% Presentación
'''
   Universidad Autónoma de Aguascalientes

         Centro de Ciencias Básicas
 Departamento de Ciencias de la Computación
          Aprendizaje Inteligente
                   6° "A"

         Tercera Evaluación Parcial

   Profesor: Francisco Javier Luna Rosas
  Alumnos:
      Espinoza Sánchez Joel Alejandro
      Gómez Garza Dariana
      González Arenas Fernando Francisco

  Fecha de Entrega: 6 de junio del 2021

Descripción: CJ orientado al VIH
'''


#%% Ejecución de código
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import AgglomerativeClustering

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAnS.csv')

X                                                                    =
df.iloc[:,[3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]].val
ues
```

```
#dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))

# Acomodar CJ al dataset

CJ = AgglomerativeClustering(n_clusters = 2, affinity = 'euclidean',
linkage = 'ward')
y_CJ = CJ.fit_predict(X)

# Visualizar los clusters
plt.scatter(X[y_CJ == 0,0], X[y_CJ == 0,1], s = 100, c = 'red', label
= 'Clúster 1')
plt.scatter(X[y_CJ == 1,0], X[y_CJ == 1,1], s = 100, c = 'green',
label = 'Clúster 2')
plt.title('CJ')
plt.legend()
plt.show()
```

Anexo 4: Código de PCA orientado al VIH en Python.

```
#%% Presentación
'''
   Universidad Autónoma de Aguascalientes

         Centro de Ciencias Básicas
 Departamento de Ciencias de la Computación
          Aprendizaje Inteligente
                   6° "A"

         Tercera Evaluación Parcial

   Profesor: Francisco Javier Luna Rosas
  Alumnos:
      Espinoza Sánchez Joel Alejandro
      Gómez Garza Dariana
      González Arenas Fernando Francisco

  Fecha de Entrega: 6 de junio del 2021

Descripción: PCA orientado al VIH
'''


#%% Ejecución de código
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

```
df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAS.csv')

features = ['PI', 'Negativo', 'Positivo', 'C', 'H', 'N', 'O', 'S',
'NumAtomos', 'Alifatico',
            'Hidrofatico', 'Volumen', 'Masa', 'HPScale', 'Surface',
'Alpha', 'BStrand',
            'Turn', 'MolWeight']

# Separando features
x = df.loc[:,features].values

# Separando el objetivo
y = df.loc[:,'Clusters'].values

# Estandarizando features
x = StandardScaler().fit_transform(x)

# Proyección PCA
pca = PCA(n_components = 2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents, columns =
['PC1', 'PC2'])
finalDf = pd.concat([principalDf, df[['Clusters']]], axis = 1)

# Visualizar Proyección 2D
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('PC1')
ax.set_ylabel('PC2')
ax.set_title('PCA')

targets = [0, 1]
colors = ['r', 'g']

for target, color in zip(targets, colors):
    index = finalDf['Clusters'] == target
    ax.scatter(finalDf.loc[index,    'PC1'],    finalDf.loc[index,
'PC2'], c = color, s = 50)
ax.grid
```

Anexo 5: Código de Support Vector Machine orientado al VIH en Python.

```
#%% Presentación
'''
   Universidad Autónoma de Aguascalientes
```

```
        Centro de Ciencias Básicas
 Departamento de Ciencias de la Computación
         Aprendizaje Inteligente
                  6° "A"

        Tercera Evaluación Parcial

   Profesor: Francisco Javier Luna Rosas
  Alumnos:
      Espinoza Sánchez Joel Alejandro
      Gómez Garza Dariana
      González Arenas Fernando Francisco

  Fecha de Entrega: 6 de junio del 2021

Descripción: SVM orientado al VIH
'''


#%% Ejecución de código
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAS.csv')

X = df.drop(['Tipo', 'Virus', 'Aminoacido', 'Clusters'], axis = 1)
Y = df['Clusters']

xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size =
0.20)

classifier = SVC(kernel = 'linear')
classifier.fit(xtrain, ytrain)

ypred = classifier.predict(xtest)
print(ypred)

print(classification_report(ytest, ypred))
```

Anexo 6: Código de Naive Bayes orientado al VIH en Python.

```
#%% Presentación
```

```
'''
   Universidad Autónoma de Aguascalientes

         Centro de Ciencias Básicas
 Departamento de Ciencias de la Computación
          Aprendizaje Inteligente
                   6° "A"

         Tercera Evaluación Parcial

   Profesor: Francisco Javier Luna Rosas
  Alumnos:
      Espinoza Sánchez Joel Alejandro
      Gómez Garza Dariana
      González Arenas Fernando Francisco

  Fecha de Entrega: 6 de junio del 2021

Descripción: Naive Bayes orientado al VIH
'''

#%% Ejecución de código
import random
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import precision_score
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import GaussianNB

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAS.csv')

X = df.drop(['Tipo', 'Virus', 'Aminoacido', 'Clusters'], axis = 1)
Y = df['Clusters']

xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size =
0.20)

classifier = GaussianNB()
classifier.fit(xtrain, ytrain)

ypred = classifier.predict(xtest)

matriz = confusion_matrix(ytest, ypred)
```

```
print('Matriz de Confusión')
print(matriz)

precision = precision_score(ytest, ypred)
print('Precisión del modelo')
print(precision)
```

Anexo 7: Código de Random Forest orientado al VIH en Python.

```
#%% Presentación
'''
   Universidad Autónoma de Aguascalientes

         Centro de Ciencias Básicas
 Departamento de Ciencias de la Computación
          Aprendizaje Inteligente
                   6° "A"

         Tercera Evaluación Parcial

   Profesor: Francisco Javier Luna Rosas
  Alumnos:
      Espinoza Sánchez Joel Alejandro
      Gómez Garza Dariana
      González Arenas Fernando Francisco

  Fecha de Entrega: 6 de junio del 2021

Descripción: Random Forest orientado al VIH
'''


#%% Ejecución de código
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import precision_score
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier

df = pd.read_csv('C:/Users/alexe/Desktop/Examen/TablaAS.csv')

X = df.drop(['Tipo', 'Virus', 'Aminoacido', 'Clusters'], axis = 1)
Y = df['Clusters']
```

```
xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size =
0.25)

sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.fit_transform(xtest)

classifier = RandomForestClassifier(n_estimators = 4)
classifier.fit(xtrain, ytrain)

ypred = classifier.predict(xtest)

matriz = confusion_matrix(ytest, ypred)
print('Matriz de Confusión')
print(matriz)

precision = precision_score(ytest, ypred)
print('Precisión del modelo')
print(precision)
```