UNIVERSIDAD DE PIURA

“Análisis Machine learning con Python con data Heart Failure Prediction”

PRESENTAN:

Bussalleu Vicente, Gabriel Enrique

Cornejo Ancajima, Sahid Daniel

Medina Coronado, Fernando José Severino

Perales Yuyes, Ismael Alejandro

Fernández Palomino, Randy Joel

Pupuche Morales, Álvaro Felipe

DOCENTE:

Ing. Pedro Rotta

Piura, 06 de febrero de 2022

Machine learning con Python

1. Introducción:

Para poder implementar sistemas que resuelvan problemas de forma automática, resulta de gran ayuda el uso de recursos tecnológicos, donde hoy en día, es necesario para que distintas empresas ahorren costos y tiempo. La mayor ventaja de incorporar estas técnicas es el aspecto de la automatización del enriquecimiento del conocimiento basado en técnicas de autoaprendizaje con una mínima intervención humana en el proceso, por ejemplo, pueden almacenar datos de forma más rápida y automática con una codificación previa.

El Machine Learning es más útil cuando la investigación central tiene como objetivo obtener un mayor rendimiento predictivo. Existen múltiples objetivos de investigación separados para el ML que se benefician de un rendimiento predictivo mejorado, incluyendo aplicaciones de ingeniería y, si se usa con cuidado, para comprender la naturaleza de distintos datos.

Este trabajo busca desarrollar una herramienta que ayude a realizar un primer diagnóstico que pueda predecir si alguna persona tiene o no algún problema cardiaco.

2. Análisis del problema:

La base de datos escogida es *Heart Failure Prediction*. Para comenzar, definamos la variable en cuestión, la insuficiencia cardiaca es una enfermedad crónica que se va manifestando gradualmente y se intensifica con el tiempo, mostrando síntomas como la dificultad del corazón para bombear sangre.

Para su diagnóstico se utilizan métodos complejos que demandan más costos y tiempo para el paciente, provocando una serie de limitaciones que impiden un diagnóstico concreto y a corto plazo que ponen en riesgo la salud de una persona. El criterio de selección se basó en los siguientes indicadores: *ChestPainType, cholesterol, resting ECG, maxHR, exerciseAngina, oldPeak, StSlope, heartDisease.*

ChestPainType (tipo de dolor de pecho):

Cholesterol (colesterol):

RestingECG (Electrocardiograma en descanso):

MaxHR (Frecuencia máxima cardiaca):

ExerciseAngina (Angina inducida por el ejercicio):

OldPeak (pico)

ST Slope (the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]

HeartDisease (Enfermedad cardiovascular)

El objetivo prioritario es tener una herramienta que realice un diagnóstico rápido en tiempo real, para determinar si una persona sufre de esta enfermedad.

De acuerdo con las investigaciones realizadas, el uso del deep learning, es una gran herramienta para el procesamiento de datos, en ese sentido, se hará uso de este algoritmo para lograr el objetivo planteado.

Previo al desarrollo del modelo es necesario precisar que se ha hecho uso de dos herramientas principales, la primera es una librería virtual de código abierto que ofrece una gran variedad de herramientas de aprendizaje profundo de alto nivel que ayuda a los usuarios a conseguir resultados de una manera rápida y fácil. Esta biblioteca, se basa en el lenguaje de Python.

La segunda herramienta es el entorno de programación Google Colaboratory o Google Colab, el cual permite tener un acceso gratuito a GPUs de Google y compartir el contenido fácilmente con los integrantes del equipo de investigación.

3. Análisis de resultado

En primer lugar, el programa imprime la información en una tabla.

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

Luego nos brinda opciones para poder indagar en la información de la tabla

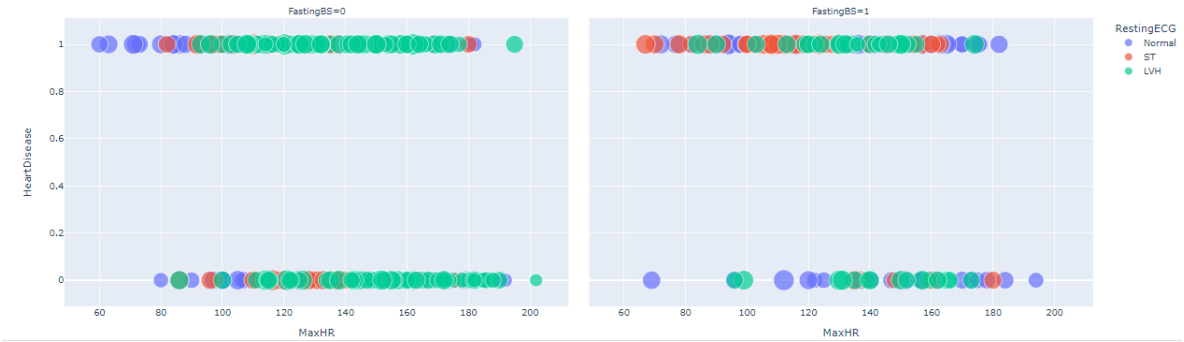
```
Presione A, si desea datos estadisticos de la data
Presione B, si desea visualizar una dispersion de datos
Presione C, si desea visualizar un histograma
Presione D, si no desea visualizar ningun grafico o dato estadistico
```

Dentro de la opción A tenemos 2 rutas donde la primera nos permitirán visualizar los datos estadísticos de toda la tabla, mientras que la segunda opción nos permite analizar datos estadísticos, pero de una sola columna, es decir si sólo buscamos datos específicos.

```
Que opcion elige:A
Presione A si desea los datos de toda la tabla
Presione B si desea datos especificos
```

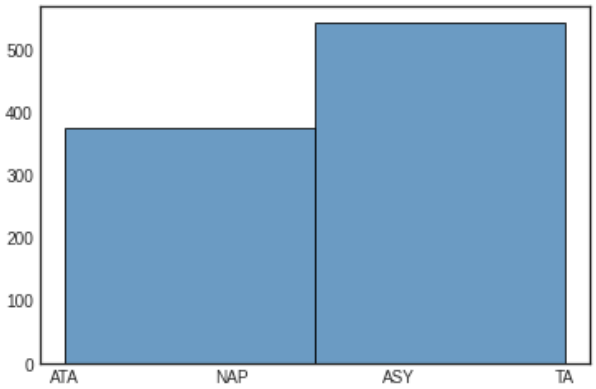
La opción B nos permite hacer una dispersión de los datos en función a la columna Heart Desease

```
Que opcion elige:B
['Age', 'Sex', 'ChestPainType', 'RestingBP', 'Cholesterol', 'FastingBS', 'RestingECG', 'MaxHR', 'ExerciseAngina', 'Oldpeak', 'ST_Slope']
Escoge una columna de interes: MaxHR

['Sex', 'ChestPainType', 'FastingBS', 'RestingECG', 'ExerciseAngina', 'ST_Slope']
Escoge una columna de separacion: RestingECG
Escoge otra columna de separacion: FastingBS

['Age', 'RestingBP', 'Cholesterol', 'FastingBS', 'MaxHR']
Escoge una columna para representar el tamaño: Age
```



La opción C tiene 2 rutas, la opción H y HD. La opción H nos permite ver cómo se distribuye la información de la columna, mientras que la opción HD nos permite diferenciar esta información con la variable Heart Desease.

```
Que opcion elige:C
Presione H si desea un histograma para una columna:
Presione HD si desea un hisograma diferenciada entre HeartDisease y la variable que ingrese:
Que opcion elige: H
Ingrese la columna: ChestPainType
Cantidad de intervalos que desea: 2
```



Manejo de features:

Las variables “Sex, ExerciseAngina, ChestPainType, RestingECG, ST_Slope″ le aplicamos la función LabelEncoder y al resto de variables le aplicamos el escalamiento de datos usando la función MinMaxScaler.

Resultado de las métricas para el modelo KNN sin PCA

```
El score en entrenamiento es:  0.885558583106267
El recall en entrenamiento es:  0.9334975369458128
El precision en entrenamiento es:  0.8692660550458715
```

```
El score en validación es:  0.8478260869565217
El recall en validación es:  0.9019607843137255
El precision en validación es:  0.8363636363636363
```

Resultado de las métricas para el modelo KNN con PCA

```
El score en entrenamiento es:  0.8719346049046321
El recall en entrenamiento es:  0.9211822660098522
El precision en entrenamiento es:  0.8577981651376146
```

```
El score en validación es:  0.8152173913043478
El recall en validación es:  0.8725490196078431
El precision en validación es:  0.8090909090909091
```

Resultado de las métricas para el modelo regresión logística sin PCA:

```
El score en entrenamiento es 0.8528610354223434
El recall en entrenamiento es 0.8679706601466992
La precision en entrenamiento es 0.8679706601466992
```

```
El score en validación es 0.8478260869565217
El recall en validación es 0.8888888888888888
La precision en validación es 0.8380952380952381
```

Resultado de las métricas para el modelo regresión logística con PCA:

```
El score en entrenamiento es 0.8038147138964578
El recall en entrenamiento es 0.8141809290953546
La precision en entrenamiento es 0.830423940149626
```

```
El score en validación es 0.7771739130434783
El recall en validación es 0.7878787878787878
La precision en validación es 0.7959183673469388
```

Resultado de las métricas para el modelo random forest sin PCA

```
El score en entrenamiento es 1.0
El recall en entrenamiento es 1.0
La precision en entrenamiento es 1.0
```

```
El score en validación es 0.8695652173913043
El recall en validación es 0.896
La precision en validación es 0.8682170542635659
```

Resultado de las métricas para el modelo random forest con PCA

```
El score en entrenamiento es 1.0
El recall en entrenamiento es 1.0
La precision en entrenamiento es 1.0
```

```
El score en validación es 0.8347826086956521
El recall en validación es 0.848
La precision en validación es 0.848
```

Resultado de las métricas para el modelo SVC sin PCA

```
El score en entrenamiento es 0.8837209302325582
El recall en entrenamiento es 0.9071618037135278
La precision en entrenamiento es 0.8837209302325582
```

```
El score en validación es 0.7956521739130434
El recall en validación es 0.7938931297709924
La precision en validación es 0.8387096774193549
```

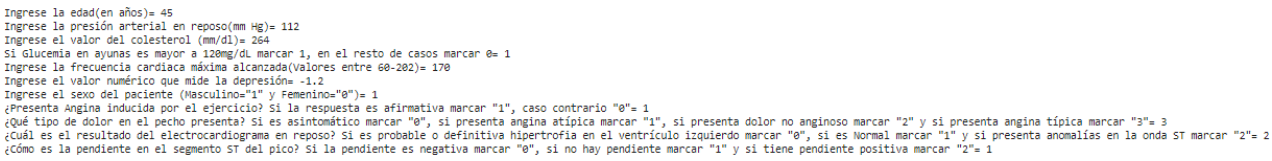
Resultado de las métricas para el modelo SVC con PCA

```
El score en entrenamiento es 0.8037790697674418
El recall en entrenamiento es 0.8793969849246231
La precision en entrenamiento es 0.8009153318077803
```

```
El score en validación es 0.8
El recall en validación es 0.8545454545454545
La precision en validación es 0.7580645161290323
```

Valor de predicción para cualquier dato:

Lo primero que hicimos es indicar valores para las siguientes variables, como se observa en la captura de pantalla.



Con los primeros 6 datos ingresados se formó una matriz 1x6 y se les realizó un escalamiento de datos. Con los otros 5 datos se formó una nueva matriz de 1x5 y luego ambas matrices se unieron en una sola matriz, y esa matriz de 1x11 sirvió para ingresarlo en los 4 modelos de machine learning que hemos analizado sin PCA.

4. Conclusiones:

- Para el modelo de Regresión Logística clasificación sin PCA se podría decir que obtenemos un modelo bueno pero que se podría mejorar con un modelo más implementado; también observamos que no hay sobreajuste ni sub-ajuste, ya que la diferencia es mínima entre los resultados que se han obtenido para la data de validación y para la data de entrenamiento. En cambio, para el modelo de Regresión Logística con PCA los resultados son diferentes. Ya que para empezar se observa a simple vista que el modelo sufre una disminución de todas las métricas evaluadas; esto se puede deber a que al aplicar el PCA la data sufre una reducción dimensional a 2 dimensiones, y por ende puede perder información valiosa para el modelo. Se puede decir que el PCA toma las variables que tienen más peso en el modelo, las usa para entrenar y validar el modelo, esto no siempre funciona; ya que vemos un mínimo sobreajuste del modelo.
- En materia de programación, se podría obtener un código menos extenso si en la implementación utilizábamos funciones.
- No hemos encontrado en la data ninguna observación con missing values =null
- Con respecto a las métricas del modelo KNN con y sin PCA se pueden observar que son buenas los valores de entrenamiento son altos y los valores de validación son muy cercanos a los de entrenamiento. Por lo que podemos indicar que no hay sobre ajuste.
- Nos gustaría que con la ayuda de algún médico donde nos pueda brindar datos reales e ingresar información a nuestros modelos y ver que tanto predicen.
- Para el modelo de Random Forest para clasificación sin PCA vemos un sobreajuste ya que las métricas que nos da para la data de entrenamiento es con valor 1, en cambio para la data de validación nos da unos valores que no son mayores a 0.9. Es igual para el modelo de Random Forest para clasificación con PCA, ya que se esperaba que mejorase el modelo, pero en vez de ello empeoró; ya que sigue saliendo para las métricas de entrenamiento el valor unitario y para las métricas de validación el modelo se empeoró. Esto se puede deber al número de árboles que se implementó, ya que lo normal, lo estándar es 100 árboles y en nuestro análisis se puso 120. Y como se mencionó el PCA no siempre es bueno, ya que se puede perder información valiosa.
- Para el modelo de SVC para clasificación sin PCA vemos un pequeño desbalance entre las métricas que nos arroja éste, ya que los valores de entrenamiento son un poco mayores que los valores de validación, esto se puede deber al hiper parámetro Scale, también podemos ver un pequeño sobreajuste; en cambio para el Modelo de SVC para clasificación con PCA, los resultados sufren una pequeña mejora, ya que la distancia entre los valores se acorta, esto se puede deber a las dimensiones evaluadas en el modelo, ya que como se dijo antes toma las variables que tienen más peso en el modelo y junto con estas elabora el modelo.