Stroke Prediction

Proyecto Final: Coder House- Grupo 7

Alumnos:

- Juan Pascual
- Lucas Ariel Saavedra
- Bartolomé Oscar Meritello
- Jose Mornaghi

Tabla de contenidos:

El documento se encuentra cross-referenced.

Plan de Investigación

Nuestro trabajo parte del análisis de un data set que obtuvimos de Kaggle (Kaggle Dataset Stroke).

El objetivo principal del trabajo es lograr predecir si un individuo sufrirá un accidente cerebrovascular (Stroke) o no.

¿Qué es un ACV (stroke)?
¿Cómo podemos prevenirlo?

Para contextualizar al lector, creemos útil analizar los motivos a través de ciertas preguntas cómo:

- ¿Porque una persona sufre un Stroke, y cuáles son sus motivos?
- ¿La edad, impacta en las probabilidades de sufrir esta enfermedad?
- ¿El estilo de vida de una persona la podría inducir a sufrir esta dolencia?
- ¿Cuán significativo es el factor edad, por ejemplo?
- ¿Ser fumador influye en la misma?
- ¿El nivel de glucosa en sangre?
- ¿El índice de masa corporal?
- ¿El tipo de trabajo?
- ¿Los hombres son más propensos a sufrir un ACV?
- ¿El lugar de residencia de una persona podría influir?
- Padecer de hipertensión o haber sufrido un ataque al corazón aumentan el riesgo, ¿Es esto cierto?

El data set utilizado presenta 12 variables, 11 de las cuales nos ayudarán (o no) a responder las preguntas planteadas previamente entorno a la variable objetivo "Stroke".

A continuación, describiremos brevemente cada una de estas variables, algunas categóricas y algunas numéricas:

- id: número identificatorio del individuo.
- gender: género del individuo estudiado (masculino, femenino)
- age: edad del individuo
- hypertension: parámetro de salud relacionado. ¿El individuo tiene hipertensión? Posibles respuestas: Si (1), No (0).
- heart_disease: parámetro de salud relacionado. ¿El individuo tiene alguna enfermedad cardíaca? Posibles respuestas: Si (1), No (0).
- ever_married: estado civil del individuo. ¿Está casado? Si, No.
- work_type: tipo de trabajo del individuo. Se toma en consideración la posibilidad de que el mismo sea un niño y en ese caso se lo marca como tal. Posibles respuestas: Privado, 'Self-employed', 'Govt_job', Niño, 'Never_worked'.

- ~ Residence_type: parámetro personal, tipo de residencia del individuo distinguiendo entre zona Urbana o Rural.
- ~ avg_glucose_level
- ~ smoking_status: parámetro de salud relacionado. ¿Es fumador el individuo? Posibles respuestas: Si (1), No (0).
- ~ bmi: parámetro de salud relacionado. Índice de masa del individuo.
- ~ stroke: variable 'Target' u Objetivo. ¿La persona sufrió un Stroke?

Incluimos el Dataset para un preview de cómo se nos presenta la información.

id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type
9046	Male	67	0	1	Yes	Private	Urban
51676	Female	61	0	0	Yes	Self-employed	Rural
31112	Male	80	0	1	Yes	Private	Rural
60182	Female	49	0	0	Yes	Private	Urban
1665	Female	79	1	0	Yes	Self-employed	Rural
56669	Male	81	0	0	Yes	Private	Urban
53882	Male	74	1	1	Yes	Private	Rural
10434	Female	69	0	0	No	Private	Urban
27419	Female	59	0	0	Yes	Private	Rural

Realizamos unas Tablas de Frecuencia para observar la cantidad de individuos que sufren un Stroke en nuestro data set.

	Frec_abs	frec_abs_acum	frec_rel_%	frec_rel_%_acum
0	4861	4861	95.1272	95.1272
1	249	5110	4.8728	100.0000

Reprocesamiento inicial de los Datos. Observamos los dtypes y si nuestro data set tiene algún factor faltante.

```
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   gender             5110 non-null   object 
 1   age                5110 non-null   float64
 2   hypertension       5110 non-null   int64  
 3   heart_disease      5110 non-null   int64  
 4   ever_married       5110 non-null   object 
 5   work_type          5110 non-null   object 
 6   Residence_type     5110 non-null   object 
 7   avg_glucose_level  5110 non-null   float64
 8   bmi                4909 non-null   float64
 9   smoking_status     5110 non-null   object 
 10  stroke             5110 non-null   int64  
dtypes: float64(3), int64(3), object(5)
```

```
gender                 0
age                    0
hypertension           0
heart_disease          0
ever_married           0
work_type              0
Residence_type         0
avg_glucose_level      0
bmi                  201
smoking_status         0
stroke                 0
dtype: int64
```

Podemos ver que la única variable que contiene datos NaN (nulos) es la del índice de masa corporal ('bmi').

En base a esta información, imputaremos los datos de la variable BMI con la función KNNImputer. Los valores faltantes de cada muestra se imputan utilizando el valor medio de los vecinos más cercanos de n_neighbors que se encuentran en el conjunto de entrenamiento.

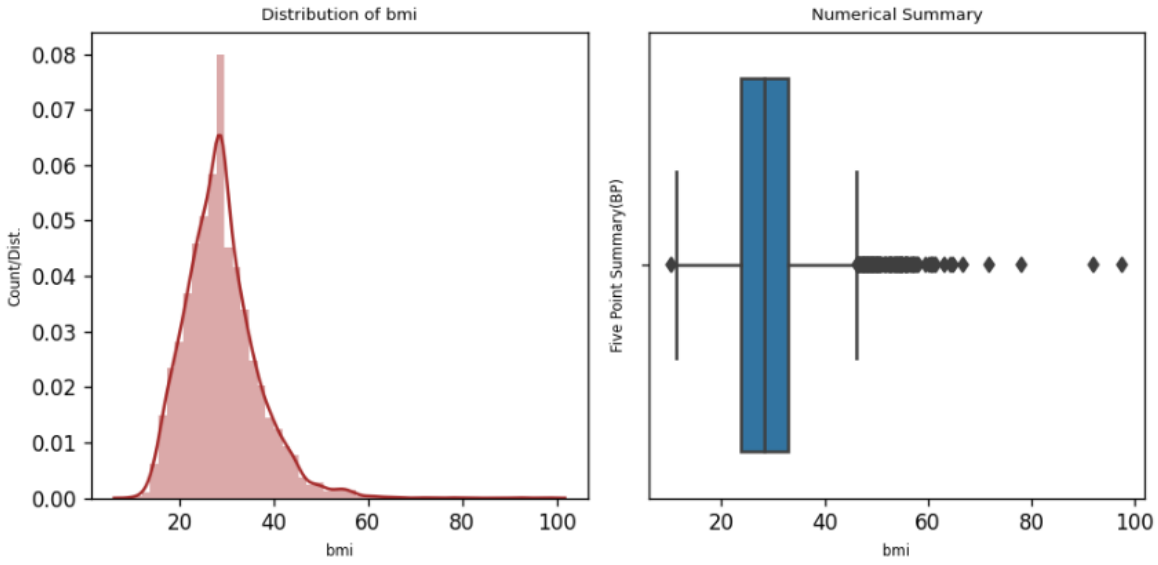
	count	mean	std	min	25%	50%	75%	max
bmi	4909.0	28.893237	7.854067	10.3	23.5	28.1	33.1	97.6
imputed_bmi	5110.0	28.893237	7.698018	10.3	23.8	28.4	32.8	97.6

De ahora en más, trabajaremos con el Data Set con los datos de BMI imputados, creando un nuevo archivo CSV y trabajaremos con el mismo.

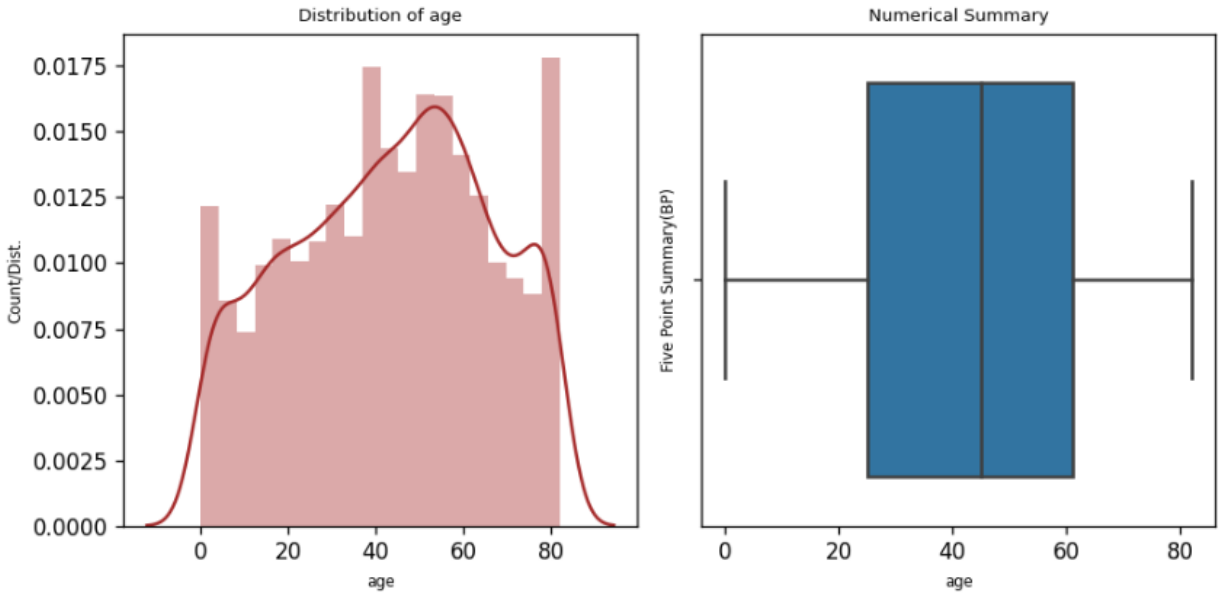
Analisis Univariado

Diseñamos una función que usaremos para realizar este análisis, que despliega un histograma y un gráfico de caja de porcentajes de la variable analizada.

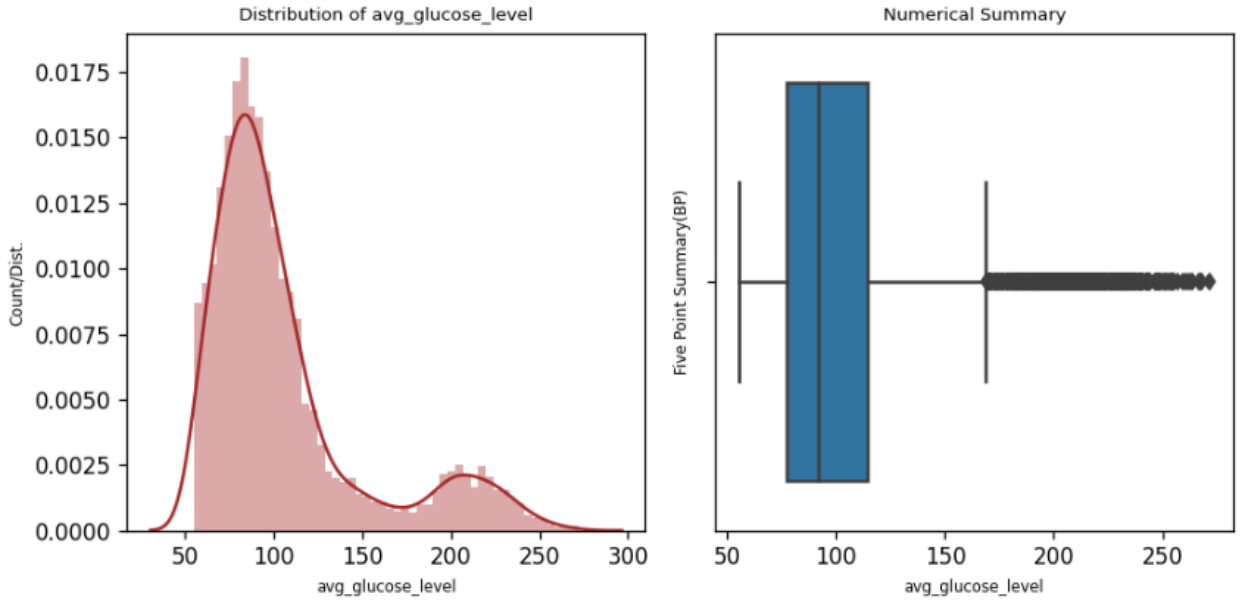
Índice de masa corporal (BMI)



Edad

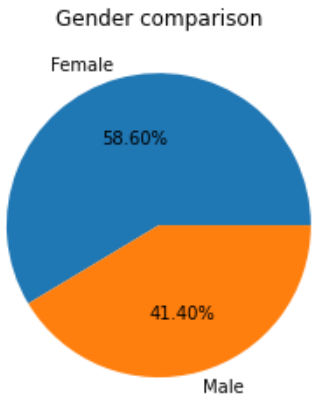


Nivel de glucosa

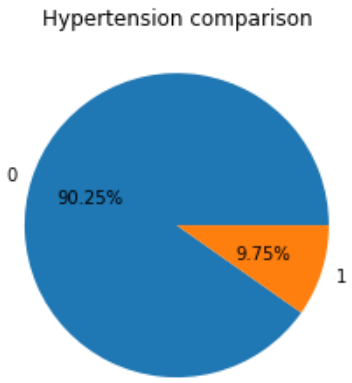


Ahora, por medio de gráficos Pie, analizamos la distribución de las variables categóricas:

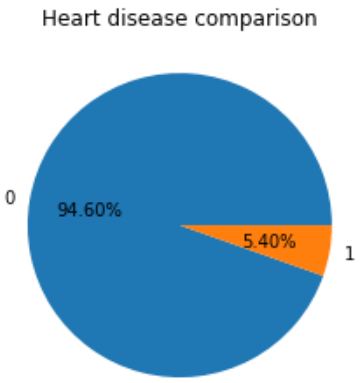
Género



Hipertensión



Enfermedades Cardíacas



Analizamos las frecuencias de cada variable categórica con el fin de observar alguna correlación o indicio en la posibilidad de un Stroke:

Smoking Status	**Frecuencia Absoluta**	**Frecuencia Absoluta Acumulada**	**Frecuencia Relativa (%)**	**Frecuencia Relativa Acumulada (%)**
Nunca Fumaron	1892	1892	37.03	37.03
Fumó	1544	3436	30.22	67.25
Fuma	884	4320	17.30	84.55
No se sabe	789	5109	15.45	100

La idea de las tablas de frecuencia reside en ver si hay algún factor que se destaque a lo largo del data set.

Género	**Frecuencia Absoluta**	**Frecuencia Absoluta Acumulada**	**Frecuencia Relativa (%)**	**Frecuencia Relativa Acumulada (%)**
Femenino	2994	2994	58.60	58.60
Masculino	2115	5109	41.40	100

Enfermades Cardíacas	**Frecuencia Absoluta**	**Frecuencia Absoluta Acumulada**	**Frecuencia Relativa (%)**	**Frecuencia Relativa Acumulada (%)**
No Tuvo	4833	4833	95	58
Tuvo	276	5109	5	100

Hipertensión	**Frecuencia Absoluta**	**Frecuencia Absoluta Acumulada**	**Frecuencia Relativa (%)**	**Frecuencia Relativa Acumulada (%)**
No Es hipertenso	4611	4611	90	90
Es hipertenso	498	5109	10	100

Es importante conocer cómo se distribuyen nuestros factores.

Estado Civil	**Frecuencia Absoluta**	**Frecuencia Absoluta Acumulada**	**Frecuencia Relativa (%)**	**Frecuencia Relativa Acumulada (%)**
Se casó	3353	3353	65	65
No se casó	1756	5109	35	100

Residencia	**Frecuencia Absoluta**	**Frecuencia Absoluta Acumulada**	**Frecuencia Relativa (%)**	**Frecuencia Relativa Acumulada (%)**
Urbano	2596	2596	51	51
Rural	2513	5109	49	100

Trabajo	**Frecuencia Absoluta**	**Frecuencia Absoluta Acumulada**	**Frecuencia Relativa (%)**	**Frecuencia Relativa Acumulada (%)**
Privado	2924	2924	57	57
Independiente	819	3743	16	73
¿Children?	687	4430	13	86
Gubernamental	657	5087	13	99
Nunca trabajó	22	5109	0.4	100

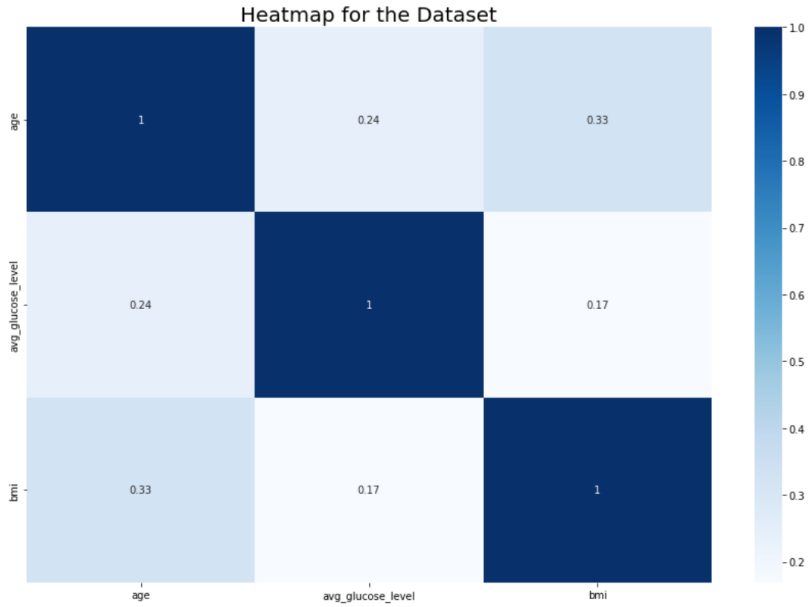
Por último, volvemos a observar la cantidad de personas que sufren un Stroke (ACV) en nuestro data set, para no perder el objetivo del trabajo.

ACV	Frecuencia Absoluta	Frecuencia Absoluta Acumulada	Frecuencia Relativa (%)	Frecuencia Relativa Acumulada (%)
No tuvo ACV	4860	4860	95	95
Tuvo ACV	249	5109	5	100

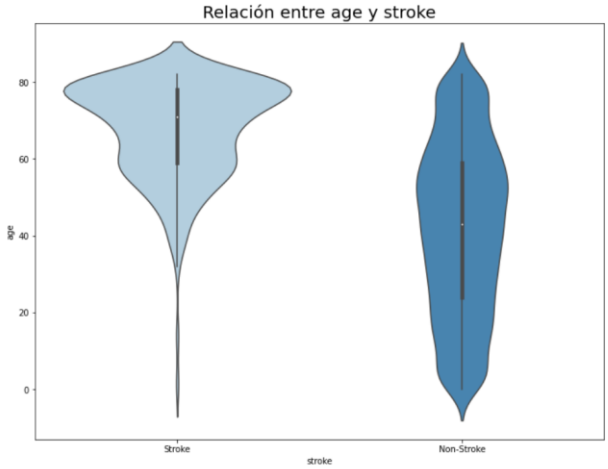
Analisis Bivariado

El análisis bivariado consiste en observar la relación entre dos variables.

Realizamos un HeatMap con las variables numéricas para ver el factor de correlación entre las mismas y el factor Stroke.



El heatmap nos muestra que la variable edad puede ser útil para clasificar cierto grupo etario con mayor propensión a sufrir un Stroke.



El grafico de violín de la izquierda nos muestra como se distribuye el factor Stroke en relación a la edad. A partir de los 40 años el Stroke se vuelve un riesgo presente, y la mayoría de los casos se observan en el rango etario 75-80.

El grafico de violín de la izquierda nos muestra cómo se distribuye el factor Stroke en relación al nivel de glucosa. Podemos apreciar una leve influencia de la variable avg_glucose_level (nivel de glucosa en sangre) sobre si una persona tuvo un Stroke o no. Debido a que la mayoría de los individuos tienen un nivel de glucosa cercano a 100, es lógico que linealmente también se observen la mayor cantidad de Strokes. Pero si es ligeramente divisible que hay una mayor proporción de sujetos con un avg_glucose_level cercano a 200, qué si tuvieron un Stroke, respecto de los que no. Tal vez esto nos ayude a ver que tener un avg_glucose_level > 170 puede significar un riesgo mayor.

En consonancia con las practicas de análisis bivariado, realizamos unos crosstab entre variables categóricas, y observamos algunas tendencias.

ever_married	No	Yes
stroke		
Non-Stroke	1727	3133
Stroke	29	220

ever_married	No	Yes
stroke		
Non-Stroke	0.338031	0.613232
Stroke	0.005676	0.043061

Proporcionalmente, aquellos sujetos que se casaron alguna vez, tienen significativamente más probabilidades de sufrir un Stroke frente aquellos que nunca lo hicieron.

Analizando la variable Heart Disease.

heart_disease	No	Yes
stroke		
Non-Stroke	4631	229
Stroke	202	47

De las 4631 personas que no tienen una afección cardiaca, solo 202 personas sufren un Stroke. Nos da una proporción menor al 5%.
El 20% de los sujetos que tuvieron/tienen una condición cardiaca, sufren un Stroke. Podríamos inferir cierta relación entre Heart Disease y Stroke.

Variable Hipertensión.

hypertension	No	Yes
stroke		
Non-Stroke	4428	432
Stroke	183	66

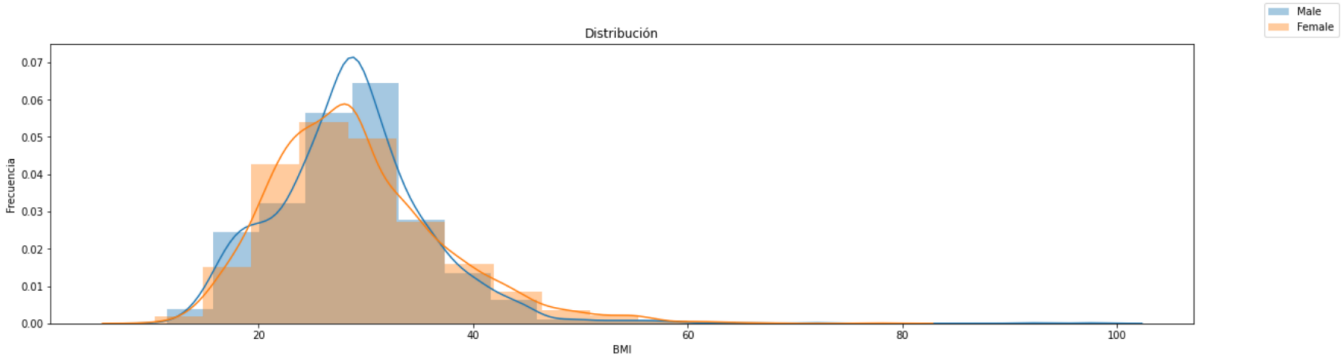
De las 4428 personas que no tienen hipertensión, solo 183 personas sufren un Stroke. Nos da una proporción cercana al 4%.
Diferente es la situación de aquellos que tienen hipertensión. De los 432, hay 66 personas que sufrieron un Stroke, una proporción mayor al 15%.

Al igual que Heart Disease, la Hipertensión es un factor a considerar.

Analisis Multivariado

Realizar un análisis multivariado nos permite unir factores con mayor correlación para observar que conjunto de individuos, con ciertas características que comparten, son los mas expuestos a sufrir un Stroke.

Gráfico de Distribución entre los géneros en función del BMI

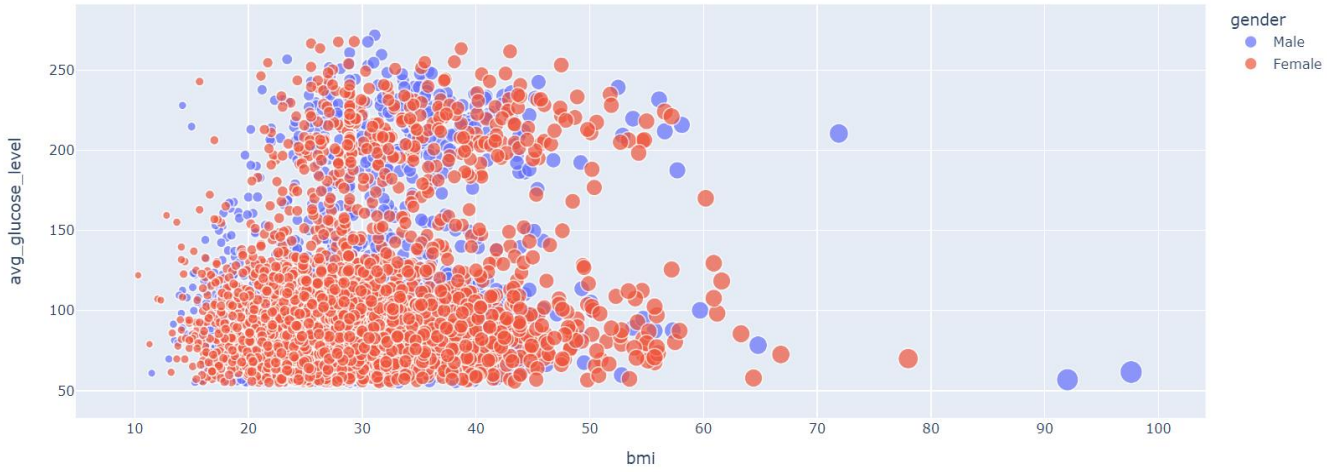


Podemos observar que el nivel de BMI de los hombres se distribuye de una manera más homogénea, que el de las mujeres.

La mayor concentración de BMI en hombres se haya aproximadamente en 30, mientras que el de las mujeres se distribuye más equitativamente entre 20 y 30.

Hay una ligera concentración mayor en las mujeres de BMI mayor a 40 en relación a los hombres, pero el género masculino es el de outliers más notorios.

Representación de la relación entre niveles de glucosa y BMI diferenciado por genero

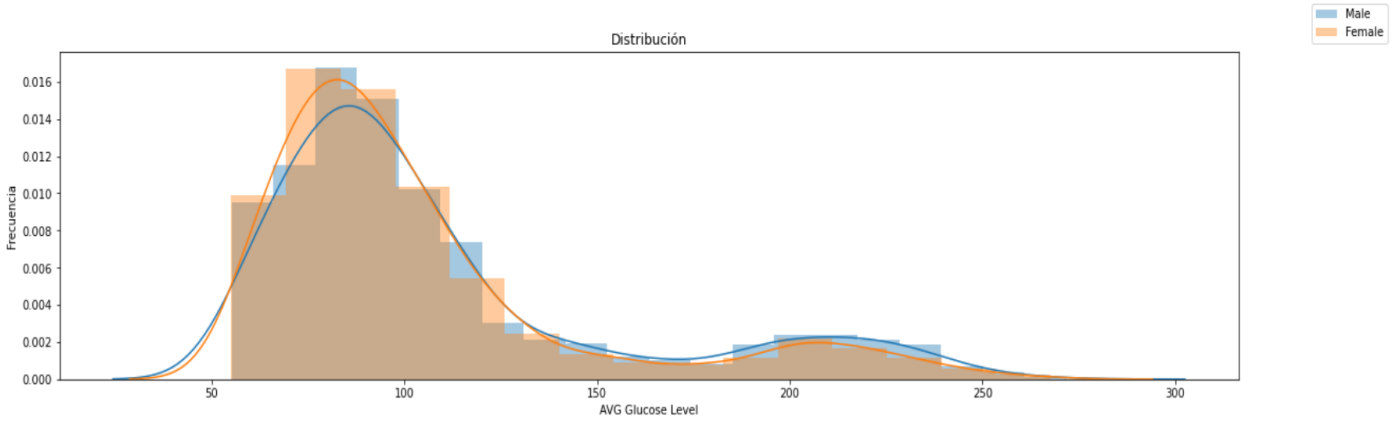


El grafico desea mostrar que relación puede llegar a haber entre nivel de glucosa y BMI.

Si bien en el grafico no podemos inferir una relación lineal entre ambas, siendo un valor determinado de BMI asociado generalmente a un intervalo especifico, la medicina ha demostrado que hay una directa

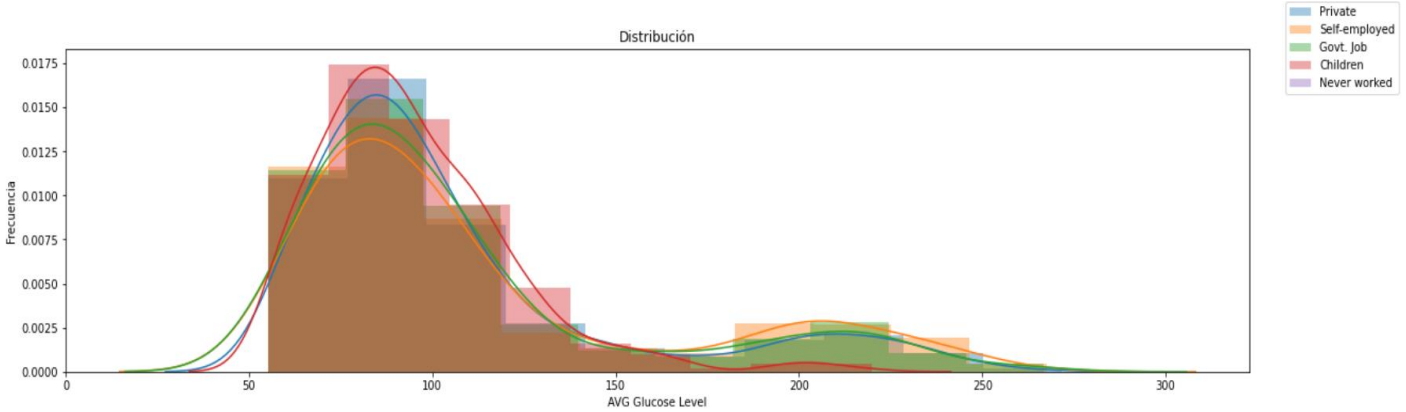
correlación entre un BMI elevado y un nivel de glucosa elevado. Un nivel de glucosa elevado conlleva a diabetes, y se da generalmente en persona de BMI elevado con una ingesta exagerada de azucares procesados.

Distribución entre los géneros en función del nivel de glucosa



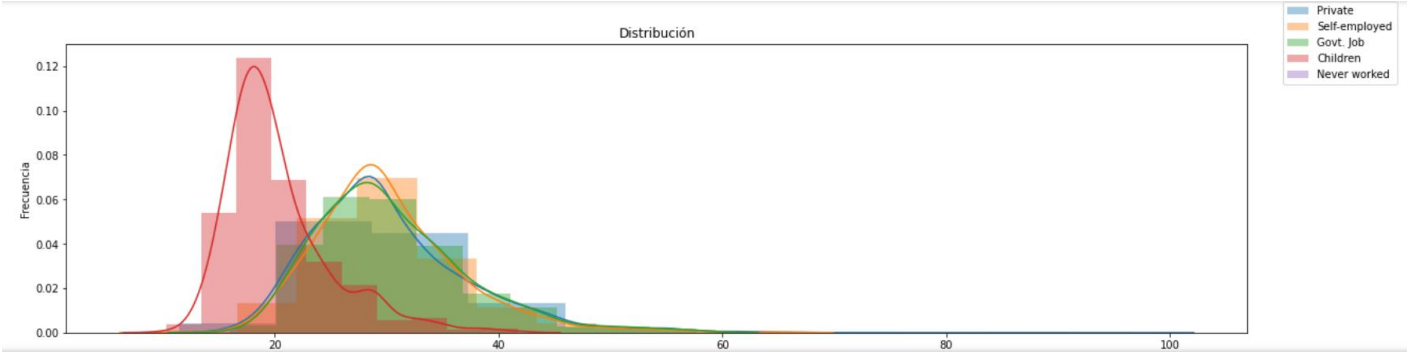
Ambos géneros se distribuyen de manera muy similar, con la mayoría de los integrantes del data set encontrándose entre los valores 60-100 de glucosa.

Distribución entre los diferentes tipos de trabajo en función del nivel de glucosa



El grafico intenta mostrar la diferencia en base a los tipos de trabajo. Es fácil inferir que los niños tienen un menor nivel de glucosa en general, y que los self employed son el grupo de personas de cola más pesada.

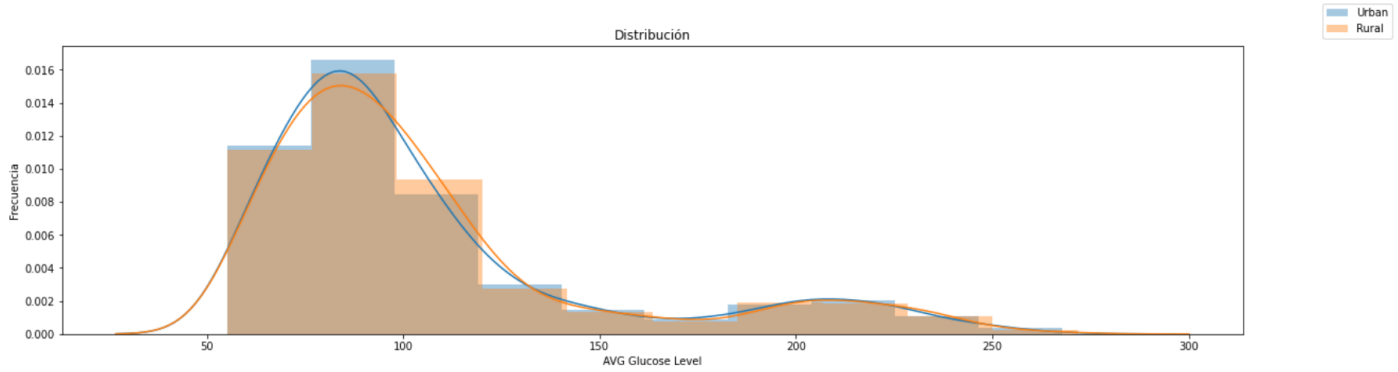
Distribución entre los diferentes tipos de trabajo en función del BMI



Este grafico a diferencia del anterior utiliza el BMI como indicador. El mismo patrón se repite en los niños, siendo este grupo el de menor BMI.

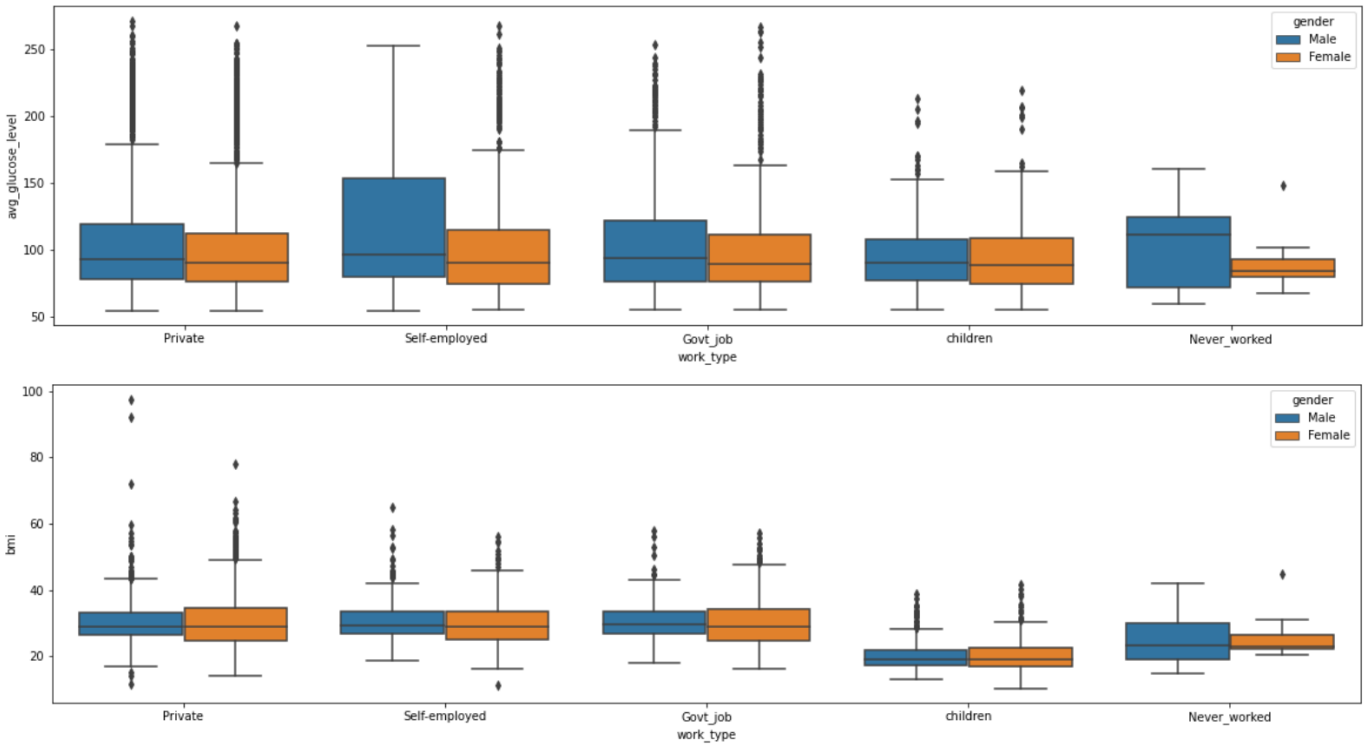
El resto de los grupos se distribuyen similarmente.

Distribución entre los diferentes residencias en función del nivel de glucosa



Se observan diferencias despreciables entre residentes urbanos y rurales con respecto a sus niveles de glucosa.

Análisis del tipo de trabajo con relación a glucosa y BMI



Modelos de Clasificación

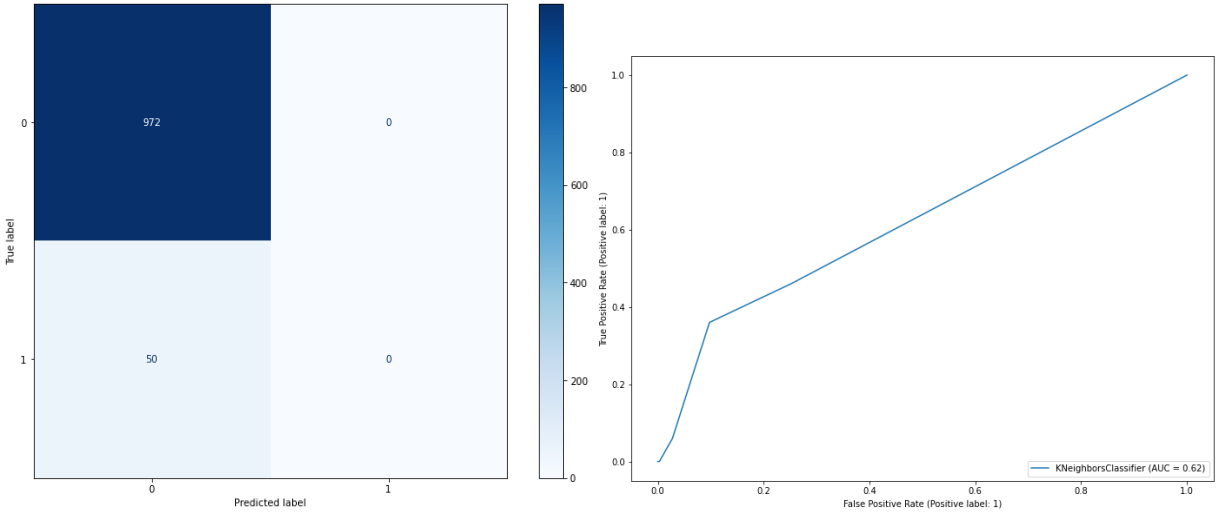
Un modelo de clasificación nos permite poder clasificar datos en un conjunto finito de categorías. En contraste con los de regresión, no se busca predecir un número real.
El objetivo es predecir un resultado de carácter binario, es decir si es hombre/mujer, gato/perro, ganador/no ganador, etc. En nuestro caso, el resultado categórico que queremos predecir es si No va a sufrir un Stroke (ACV), o sí.

Antes de comenzar con el modelo, debemos hacer algunos ajustes en el data set.
Los modelos de clasificación requieren convertir las variables categóricas en variables numéricas, que catalogamos como "dummies".
Este procedimiento nos permite utilizar dichas variables en los modelos, que solo aceptan inputs numéricos. Por lo que debemos reconvertir las variables categóricas que utilizamos para los gráficos, en numéricas.

KNN (K-Nearest Neighbour)

Este modelo de Machine Learning supervisado utiliza un concepto de determinación en el environment. Los valores de los data points, se determinan por los data points cercanos. Haciendo una analogía para entender el concepto, las personas son afectadas, influenciadas, por las personas que los rodean. El modelo mide la distancia entre los distintos data points, y los agrupa en base a clases. Los puntos de dichas clases, tienen una gran cantidad de características en común, por eso son agrupados dentro de esas especificas clases. Es importante conocer bien el data set y realizar las visualizaciones correctas, para

entender cuántas clases existen dentro de nuestro data set. De dicha manera vamos a poder realizar una clasificación más fidedigna.




```
Accuracy for our training dataset is : 95.11%
```

Conclusiones de KNN:

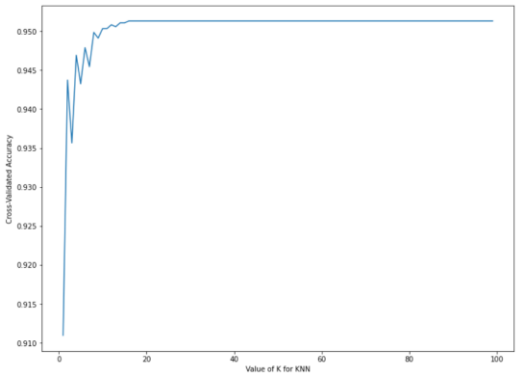
De la matriz de confusión podemos extraer algunas conclusiones. Cuando realizamos el train test split, seleccionamos al azar, el 20% de nuestro data set.
Nuestro modelo es muy efectivo en la predicción de True Negatives, ya que obtenemos un 95.11%. Son casos en lo que predecimos un No Stroke, y el data set confirma nuestra estimación.
Observando el data set, y la proporción de Stroke/No Stroke, en conjunto con la matriz de confusión, podemos dilucidar una limitación del modelo. La limitación reside en el caso de los False Negative (3er cuadrante), estimo un No Stroke, cuando Si lo era. Esta métrica es muy importante, dado que siempre que realmente una persona sufra un Stroke, el modelo debe predecirlo para tratar y prevenir.
Hay 50 casos de este tipo en el subset que utilizamos para entrenar el modelo, y no pudimos predecir ninguno.
Esta limitación genera un riesgo de salud, si se utilizara este modelo como input en un Hospital, ya que no tiene la capacidad de realmente predecir un Stroke en las situaciones en las que sucede.
La curva ROC lo demuestra. Cuando el valor de AUC es aproximadamente 0.5, el modelo no tiene la capacidad de distinguir entre clases positivas y negativas.

Hiperparametros:

Un método para mejorar la accuracy del modelo, es analizar exhaustivamente la cantidad de KNeighbors (hyperparameters) a elegir en base a las características de nuestro data set.

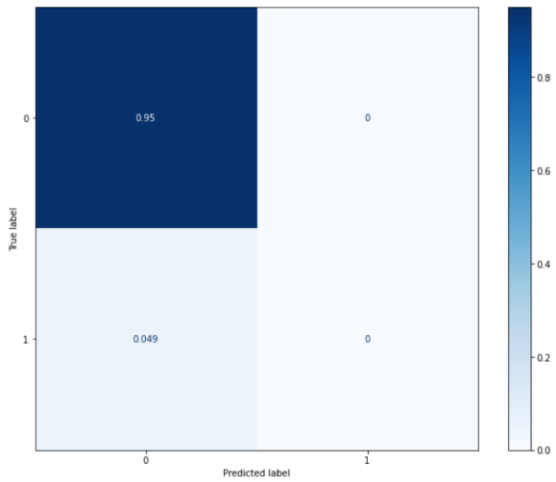
GridSearchCV es una herramienta que nos permite realizar específicamente esa tarea. Al programa le definimos un valor de K entre 1 y 100. Debido a que KNN es computacionalmente intenso, y lento mientras aumenta la base de datos, limita el approach y nos obliga a realizar una cantidad sensata de folds.

```
KNeighborsClassifier(n_neighbors=16)
```



Utilizando GridSearch hicimos una búsqueda exhaustiva, y el valor de n_neighbors con mejor accuracy score es 16.

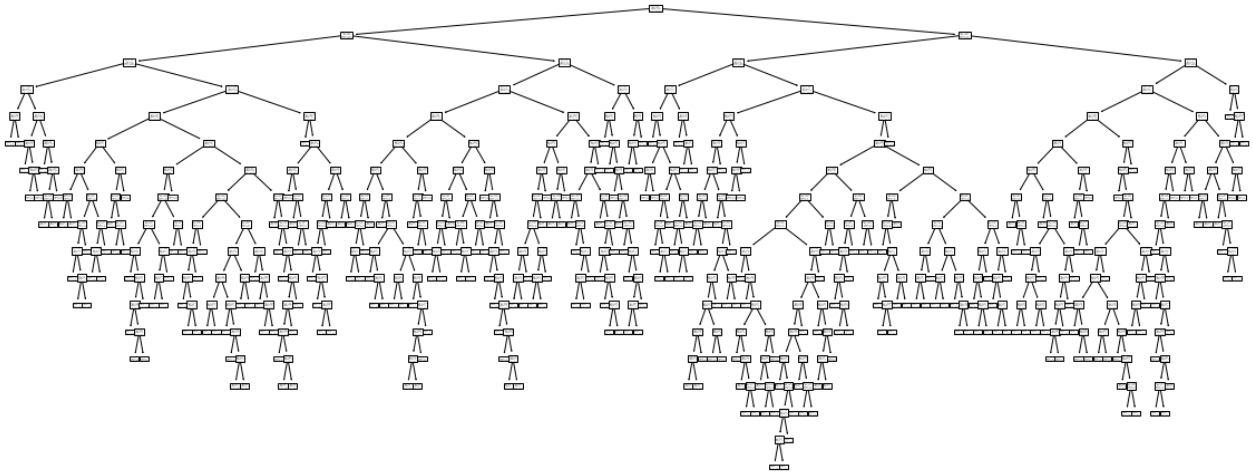
```
Accuracy for our training dataset with tuning is : 95.13%
```



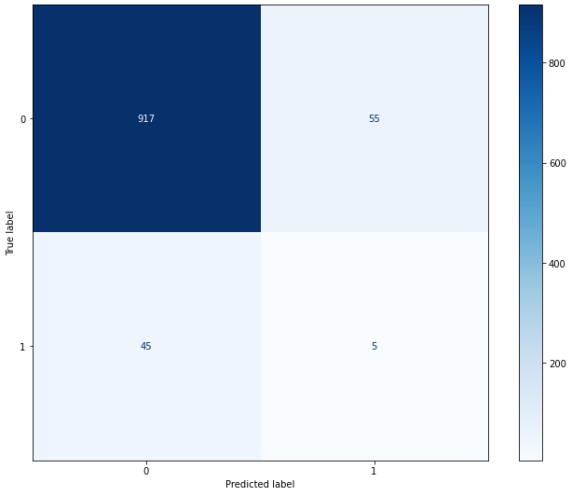
Podemos ver una mejora en la capacidad de predicción modelo, ya que obtenemos un 95.13% de True Negatives, casos en lo que predecimos un No Stroke, y el data set confirma nuestra estimación. La mejora porcentual en la estimación no implica un cambio nominal debido a que una mejora del 0.02%, en un subdataset de tan pocos individuos, no es significativo.

Decision Tree

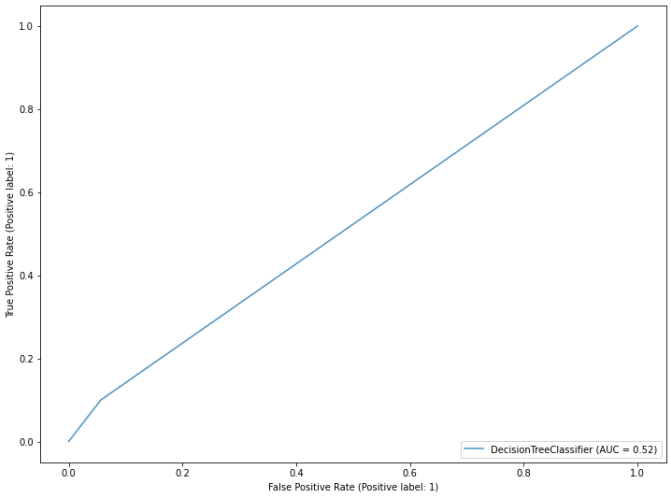
El concepto del modelo de Arboles de Decision (Decision Trees), es construir un árbol de nodes y branches, que constan de información y decisiones, acorde a cuál es el mejor camino para ajustar nuestro train set para lograr la mejor predicción. Se construyen separando recursivamente nuestro train-set usando los features que mejor ajustan para nuestro objetivo.



Matriz de Confusión



ROC Curve Decision Tree



Conclusiones Decision Tree

La capacidad de predicción de True Negatives es del 90.22%, casos en lo que predecimos un No Stroke, y el data set confirma nuestra estimación (1er cuadrante).
Y los casos en los que predecimos un Stroke, y el data set lo confirma (4to cuadrante).
La limitación está en un elemento importante que nos demuestra la matriz de confusión. En 45 casos, el Árbol de Decision predice que esas personas No sufrirán un Stroke, cuando realmente si lo sufren. Esta métrica es muy importante, define si nuestro modelo es capaz de predecir los casos de Stroke verdaderos. Ya sea porque predice un No stroke y No sucede (1er cuadrante), o porque predice un Stroke y sucede (4to cuadrante).
Seguimos observando la limitación de nuestros modelos en dicho caso.
La curva ROC lo demuestra. Cuando el valor de AUC es aproximadamente 0.5, el model no tiene la capacidad de distinguir entre clases positivas y negativas.

RandomForest

Random forest consiste en una gran cantidad de decision trees individual que operan como un ensemble. Cada tree individual define su predicción de clase, y la clase con más votos entre todos los árboles individuales, se vuelve la predicción de nuestro modelo Random Forest.

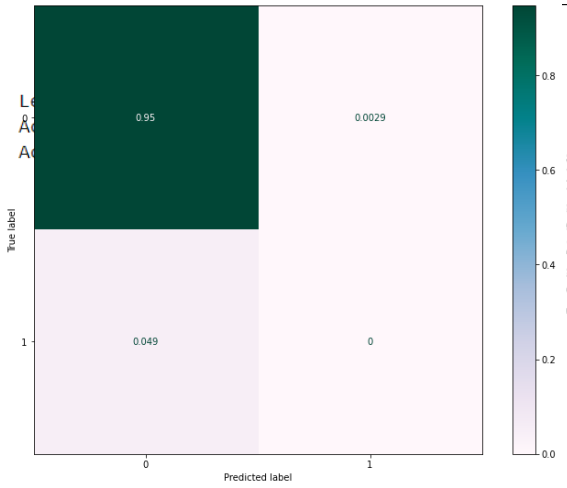
```
Accuracy for our training dataset with RandomForest is : 94.81%

 Confusion Matrix:
 [[969   3]
  [ 50   0]]
```

Classification Report

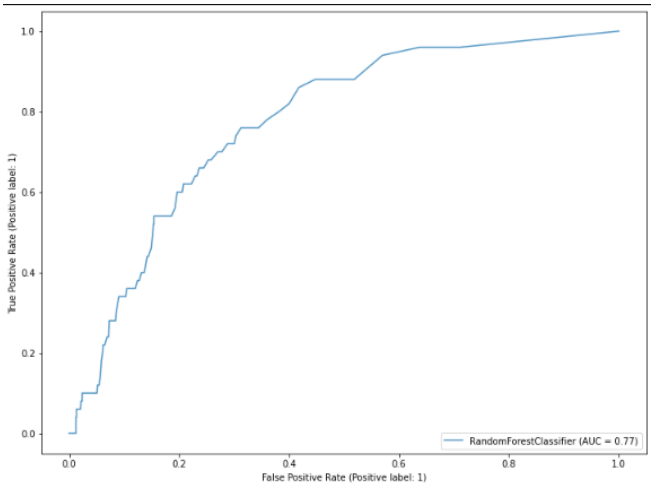
	precision	recall	f1-score	support
0	0.95	1.00	0.97	972
1	0.00	0.00	0.00	50
accuracy			0.95	1022
macro avg	0.48	0.50	0.49	1022
weighted avg	0.90	0.95	0.93	1022

Matriz de Confusion

```
Confusion Matrix:
[[970   2]
 [ 48   2]]
```



ROC Curve RandomForest



Conclusiones RandomForest

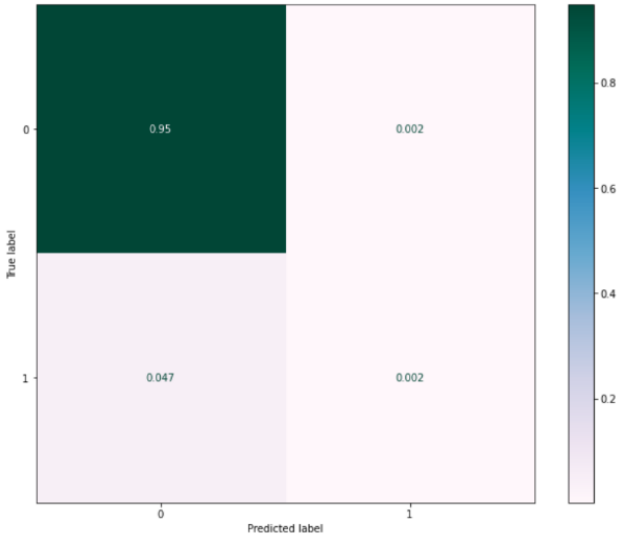
De la matriz de confusión podemos extraer algunas conclusiones.
Nuestro modelo es muy efectivo en la predicción de True Negatives, ya que obtenemos un 94.81%, casos en lo que predecimos un No Stroke, y el data set confirma nuestra estimación.
En este caso incurrimos en el error de False Positives con 3 individuos. Es el caso de estimar un Stroke, cuando no lo era. Esta limitación no genera ningún riesgo de salud, ya que solo recaería en gastos y atención extra. A diferencia del KNN, este modelo si incurre en ese error.
Comparado con Decision Tree, una mejoría en la capacidad de no incurrir en False Positives, estimar un Stroke cuando no lo era (55 DS vs 3 RF). Pero más importante aún es observar que perdemos toda capacidad de predecir True Positives, casos en los que predecimos un Stroke, y se verifica (5 DS vs 0 RF).
La curva ROC tiene una mejor pendiente, pero no es congruente con la matriz de confusión.

Gradient Boosting Classifier

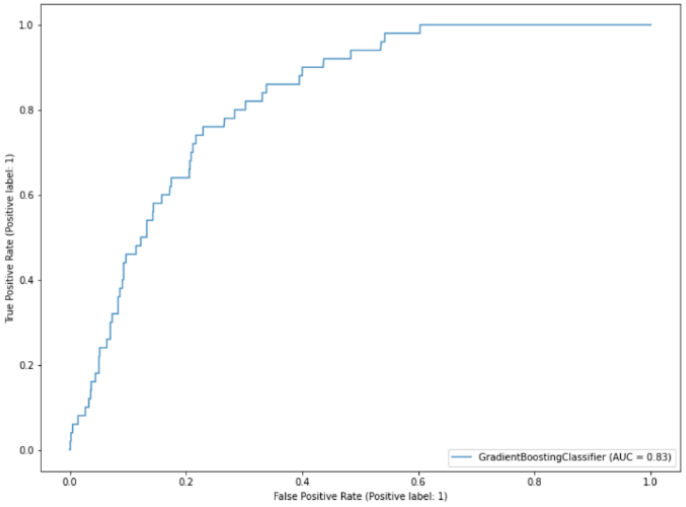
Gradient Tree Boosting Classifier es un modelo que nace del metodo de boosting, que consiste en crear un ensemble. Empieza por fittear un modelo inicial (en este caso un árbol de decision) a nuestra data. El segundo modelo se enfoca en predecir adecuadamente en los casos en los que el primer modelo tiene un mal rendimiento. Se espera que la combinación de ambos modelos tenga un mejor rendimiento. Este proceso se repite, y cada modelo sucesivo intenta corregir las limitaciones de todos los anteriores.
Realizamos un proceso de selección de distintos learning rates. El mejor accuracy score lo obtuvimos con un valor de learning rate = 1.

```
Accuracy for GBC is : 95.11%

Classification Report
              precision    recall  f1-score   support

           0       0.95      1.00      0.97       972
           1       0.50      0.04      0.07        50

    accuracy                           0.95      1022
   macro avg       0.73      0.52      0.52      1022
weighted avg       0.93      0.95      0.93      1022
```

Matriz de Confusión



ROC Curve Gradient Boosting Classifier



Conclusiones GBC (Gradient Boosting Classifier)

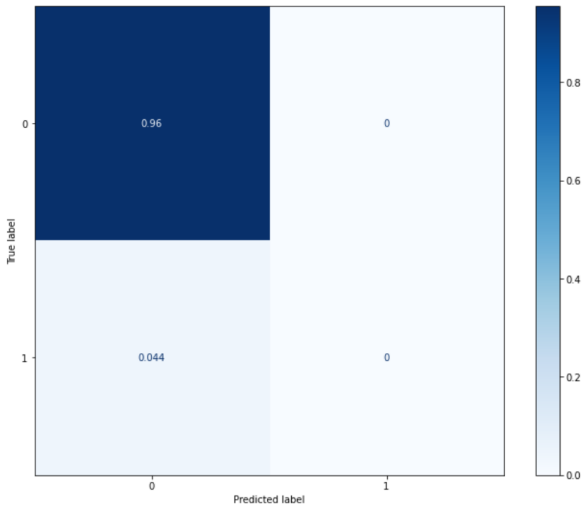
Junto con Decision Trees y Random Forest, GBC también incurre en el error de False Positive, situación en la cual predecimos un Stroke, cuando no sucede. Esta métrica es relativamente importante, dada la importancia de predecir todos los casos verdaderos en los que una persona realmente Stroke. Se disminuye ligeramente dicho error, ya que solo observamos 2 casos.
Es muy eficiente en la situación de True Negatives (95.11%), casos en lo que predecimos un No Stroke, y el data set confirma nuestra estimación (1er cuadrante).
A diferencia de Random Forest, este modelo si logra predecir 2 casos de True Positive, individuos que predecimos que tendrán un Stroke, y se verifica.
De todas maneras, seguimos observando la limitación de nuestros modelos en este caso. De 50 individuos que sufrirán un Stroke, solo podemos predecir 2.

SVC (Support Vector Classification)

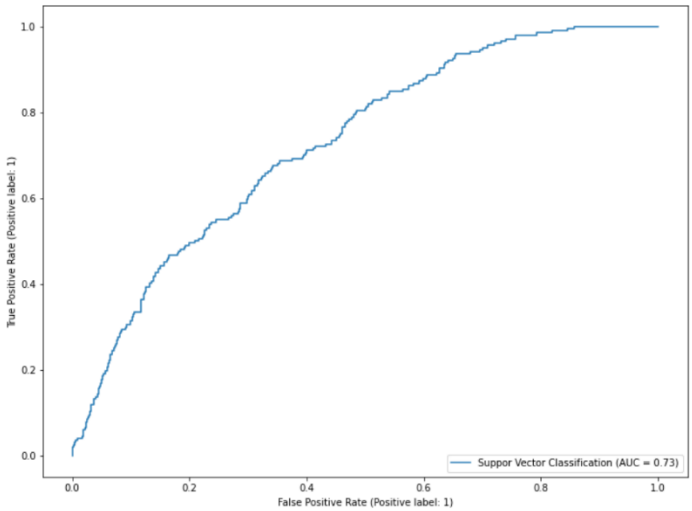
Creamos el modelo SVM (support vector machine). Un support vector machine construye un hyper-plano, o una serie de hyper-planos en n dimensiones, que podemos utilizar para nuestro análisis de clasificación. La matemática del modelo determina el mejor conjunto de vectores los cuales maximizan el margen entre los vectores (es decir maximizan la distancia entre los inputs y sus subclases) sin incurrir en misclasificar dicho sample, es decir sin adjudicar ese input en una subclase no optima.

```
Accuracy of SVC is : 95.59%


Confusion Matrix:
[[976   0]
 [ 45   0]]
```

Matriz de Confusion



ROC Curve Support Vector Classification



Conclusiones SVC

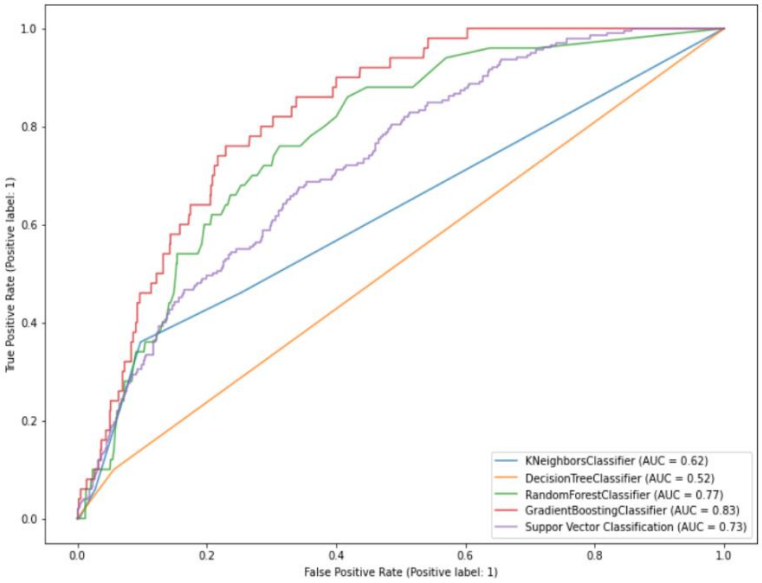
Habíamos aclarado que la limitaciones de nuestras predicciones residen en el caso de los True Positive (4to cuadrante). Casos en los que predecimos un Stroke, y de hecho sufren un Stroke.
El modelo SVC, continua con esa limitación.
Mejora la capacidad de predicción de los True Negatives, casos en lo que predecimos un No Stroke, y el data set confirma nuestra estimación.
Por ende, no incurre en el error de False Positives, situación en la cual predecimos un Stroke, cuando se da un No Stroke.

Conclusión de Modelos de Machine Learning

Utilizamos varios modelos de clasificación.

1. KNN (K-Nearest Neighbor)
2. Decision Tree
3. Random Forest
4. Gradient Boosting Classifier
5. SVC (Support Vector Classification)

Luego de implementar cada uno, realizamos los debidos análisis. La conclusión final es que el metodo GBC (Gradient Boosting Classifier) es el que obtiene mejores resultados.




```
Accuracy for GBC is : 95.11%

Confusion Matrix GBC:
[[970   2]
 [ 48   2]]
```

En todos los modelos obtuvimos resultados similares, lo que nos permitió observar la limitación de nuestras predicciones, que siempre reside en el caso de los True Positives. Casos en los que el predict de un Stroke, coincide con el true Stroke.

De todos los modelos, Decision Tree fue el que tuvo una mejor capacidad de predecir los True Positives (5 DS vs 2 GBC), pero incurre mucho en el error de False Positives, en el cual estimo un Stroke, cuando no lo era (55 DS vs 2 GBC).

El modelo GBC no es el de mejor capacidad de predicción de los True Negatives, pero por un margen muy ligero (GBC 970 vs SVC 976).

Si es el modelo de mejor curva ROC (AUC area under curve). La curva ROC es curva de probabilidades que grafica el ratio de verdaderos positivos contra lo falsos positivos, en varios threshold values.

Observando todos los modelos, sus curvas ROC, y ponderando sus limitaciones, creemos que, en la mayoría de los casos, **GBC** debería ser el que mejor ajuste.

Adecuación del DataSet y Modelos de Clasificación con Analisis de Hiperparametros

Al realizar el análisis y arribar a las conclusiones, pudimos observar que el problema del resultado de los modelos, está en el sampling. Nuestro data set contiene muy pocos individuos que sufren un Stroke, menos del 5% de todo el data set. Por lo que consideramos que para lograr un análisis más significativo de que modelo ajustaría mejor, dado el caso de contener un data set con estas características, seria realizar algún ajuste con el sampling de nuestra muestra.

Un elemento que ayudaría al análisis de los modelos, seria reajustar la escala de las variables.

Standard Scaling: Estandarizar un conjunto de datos implica reescalar la distribución de valores para que la media de los valores observados sea 0 y la desviación estándar sea 1. Es útil aplicarlo en nuestro data set, debido a que tenemos valores en diversas escalas. Muchas características se presentan en valor

binario. Otras variables como 'avg_glucose_level', tienen una escala entre 5-277, o como 'bmi', que tiene valores entre 10 y 98. Realizar una estandarización nos permite reescalar todas las variables para que tengan una ponderación similar.

SMOTE (Synthetic Minority Oversampling Technique) es una técnica de oversampling donde samples sinteticas son generadas para una clase minoritaria. Este algoritmo ayuda a compensar el problema de overfitting causado por oversampling random.

```
Before OverSampling, counts of label '1': 199
Before OverSampling, counts of label '0': 3888

After OverSampling, the shape of train_X: (7776, 15)
After OverSampling, the shape of train_y: (7776,)

After OverSampling, counts of label '1': 3888
After OverSampling, counts of label '0': 3888
```

El algoritmo genero un nuevo conjunto de datos sintéticos en el cual todos los individuos sufren un Stroke. De esta manera tenemos un data set que contiene 50% de individuos que sufren un Stroke, y 50% de individuos que no sufren un Stroke.

Métricas a destacar para el Analisis de nuestro modelos (con SMOTE realizado)

1) Recall:

Es la habilidad del modelo de encontrar todos los casos relevantes.
Matemáticamente, definimos recall como el número de true positives, dividido por el number de True Positives más el número de False Negatives.

2) Accuracy:

Es la habilidad del modelo de encontrar los casos acertados sobre todas las predicciones.
Matemáticamente representa el ratio de la suma entre True Positives + True Negatives sobre la suma de todas las predicciones.
Accuracy = True Positives + True Negatives / True Positives + False Positives+ True Negatives + False Negatives

3) Precision:

La habilidad del modelo de clasificación para clasificar solo los data points relevantes.
Matemáticamente la precisión representa el número de True Positives dividido sobre True Positives más el número de False Positives.

4) F1:

Es la media armónica de precisión y el recall.
Matemáticamente es: F1 = 2*Recall*Precision / Recall + Precision

Modelos de Clasificación para Analisis

Utilizamos los mismos modelos que en el análisis anterior, y le agregamos dos:

1) Logistic Regression
2) K-Neighbors
3) Decision Tree
4) Random Forest
5) Gradient Boosting Classifier
6) Support Vector Classification
7) XGBoost

	Model	Accuracy	K-Fold Mean Accuracy	Std. Deviation	ROC AUC	Precision	Recall	F1
0	Logistic Regression	0.744618	0.787297	1.480280	0.732942	0.127208	0.72	0.216216
1	KNeighbors	0.778865	0.901106	0.710104	0.580206	0.084906	0.36	0.137405
4	GradientBoostingClassifier	0.854207	0.878612	5.398265	0.600844	0.122137	0.32	0.176796
5	Support Vector Classification	0.855186	0.937888	1.369396	0.553930	0.091667	0.22	0.129412
2	Decision Tree	0.882583	0.906511	2.805091	0.596790	0.142857	0.28	0.189189
3	Random Forest	0.898239	0.960781	2.110151	0.548107	0.114286	0.16	0.133333
6	XGBoost	0.926614	0.955644	6.565044	0.544053	0.162162	0.12	0.137931

Realizamos un dataframe que contiene todas las métricas de nuestros modelos para mejor análisis y selección.

Utilizando **GridSearchCV**, realizamos una optimización de los HiperParametros para poder mejorar nuestra clasificación.

GridSearchCV Recall

Vamos a ponderar los hiperparametros de los modelos según su mejor valor de Recall.
Creemos que dada la dificultad de poder estimar cuando una persona va a sufrir un Stroke, deberíamos concentrarnos en obtener el mejor score Recall.

Obtuvimos los parámetros adecuados para obtener un mejor recall (explicitados en el notebook) y volvemos a correr los modelos con los parámetros ajustados.

	Model with Hyperparameters Recall	Accuracy	K-Fold Mean Accuracy	Std. Deviation	ROC AUC	Precision	Recall	F1
0	Logistic Regression	0.742661	0.787425	1.527401	0.731914	0.126316	0.72	0.214925
4	GradientBoostingClassifier	0.763209	0.828193	2.101282	0.723745	0.130769	0.68	0.219355
5	Support Vector Classification	0.795499	0.884391	1.377946	0.636379	0.112195	0.46	0.180392
1	KNeighbors	0.769080	0.897377	0.812518	0.584547	0.084821	0.38	0.138686
2	Decision Tree	0.879648	0.914487	2.794663	0.633189	0.165138	0.36	0.226415
6	XGBoost	0.896282	0.934809	4.697631	0.594506	0.158537	0.26	0.196970
3	Random Forest	0.900196	0.962196	2.158066	0.549136	0.117647	0.16	0.135593

GridSearchCV Accuracy

Utilizamos GridSearch de nuevo, pero vamos a ponderar como scoring, el mejor valor de Accuracy.
Creemos que de esta manera estamos haciendo una selección exhaustiva, para poder llegar a un modelo que resigne lo menor posible. Es decir que obtenga la mejor combinación entre Accuracy y Recall.

	Model with Hyperparameters Accuracy	Accuracy	K-Fold Mean Accuracy	Std. Deviation	ROC AUC	Precision	Recall	F1
0	Logistic Regression	0.742661	0.787425	1.527401	0.731914	0.126316	0.72	0.214925
4	GradientBoostingClassifier	0.763209	0.828193	2.101282	0.723745	0.130769	0.68	0.219355
5	Support Vector Classification	0.795499	0.884391	1.377946	0.636379	0.112195	0.46	0.180392
2	Decision Tree	0.879648	0.914487	2.794663	0.633189	0.165138	0.36	0.226415
6	XGBoost	0.896282	0.934809	4.697631	0.594506	0.158537	0.26	0.196970
1	KNeighbors	0.863014	0.939172	0.825241	0.558045	0.098214	0.22	0.135802
3	Random Forest	0.898239	0.962454	2.175242	0.548107	0.114286	0.16	0.133333

Conclusiones

Recapitulando, en todos los análisis previos a la readecuación de nuestro data set, encontrábamos el mismo problema. La limitación de nuestras predicciones, siempre residía en el caso de los True Positives. Dada la poca disponibilidad de casos de Stroke, nuestro modelos nunca lograban predecirlos.

Realizar una readecuación de nuestro data set nos otorgó una mejor capacidad de predicción.

Reescalando los valores y utilizando SMOTE, creamos un data set sintético con una proporción de 50% de individuos que sufren un Stroke.

Ya con la readecuación de nuestro data set, y con alguno de los hiperparametros definidos en el análisis previo a la readecuación, probamos los modelos.

Armamos un dataframe de los Modelos con SMOTE aplicado.

Luego, dado que nuestro data set no es el mismo al inicial, reutilizamos GridSearch para buscar los mejores HiperParametros.

Lo hicimos ponderando distintos elementos:

1) Accuracy
2) Recall

Podemos observar que al realizar algunas diferencias entre el GridSearch inicial y los siguientes.

La mayoría de los modelos mejoran en las variables de Precision, Recall y F1.

Para Logistic Regression ya habíamos encontrado los mejores hiperparametros que mejor ponderen ambos elementos, tanto la Accuracy como el Recall.

Para el GradientBoostingClassifier logramos una mejora significativa. En el modelo inicial sin GridSearch (dfSMOTE) habíamos logrado un Recall de 0.32. En el modelo de GridSearch con los hiperparametros ponderados hacia un mejor valor de Recall, logramos mejorarlo hasta 0.68, resignando solamente Accuracy del 85% al 76%.

Creemos que los modelos que ajustan mejor son la Logistic Regression y el GradientBoostingClassifier.

Llegamos a esta conclusión observando las métricas que obtuvimos.

Si bien ambos son los modelos de peor performance en cuanto a Accuracy, nos basamos en el análisis previo a SMOTE para concluir que la Accuracy no es una métrica que requiera tanta ponderación.

Sabemos que nuestros modelos anteriores, con solo el 5% de los individuos que sufrían un Stroke, tenían todos una Accuracy mayor al 90%.

Sin embargo, no eran capaces de predecir un Stroke. Nuestras predicciones de True Positives eran, en el mejor de los casos 5 Strokes sobre 50 en nuestro set de entrenamiento.

```
Logistic Regression :
[[725 247]
 [ 14  36]]
Accuracy Score:  0.7446183953033269

K-Fold Validation Mean Accuracy: 78.73 %

Standard Deviation: 1.48 %

ROC AUC Score: 0.73

Precision: 0.13

Recall: 0.72

F1: 0.22
```

```
GradientBoostingClassifier :
[[746 226]
 [ 16  34]]
Accuracy Score:  0.7632093933463796

K-Fold Validation Mean Accuracy: 82.82 %

Standard Deviation: 2.10 %

ROC AUC Score: 0.72

Precision: 0.13

Recall: 0.68

F1: 0.22
```

De estos modelos, **Logistic Regression** fue el que requirió una menor manipulación de datos.

Por ende, en casos en los cuales no es posible dedicarle tiempo y atención a data wrangling, debería ser el que obtenga mejores resultados en diversas situaciones.

Observando ambas matrices de confusión, podemos ver leves diferencias.

Gradient Boosting Classifier tiene un valor de accuracy mayor, incurre menos en el error de tipo False Positive (GBC 226 – LR 247). Logistic Regression predice dos casos mas de True Positive, pero a costa de esos 21 casos de False Positive.

La elección creemos que reside en un criterio personal.

Aclaraciones

Aclaraciones sobre la Matriz de Confusión:

La matriz esta invertida.
Si es True Negative (1er cuadrante), estimo un No Stroke, cuando no lo era.
Si es False Positive (2do cuadrante), estimo un Stroke, cuando no lo era.
Si es False Negative (3er cuadrante), estimo un No Stroke, cuando lo era.
Si es True Positive (4to cuadrante), estimo un Stroke, cuando era Stroke.

Aclaraciones sobre ROC Curve Analisis

Una curva de ROC curves delinea los positivos verdaderos en el eje "Y", y los falsos positivos en el eje "X".
Es una herramienta grafica que nos ayuda a observar la efectividad del modelo.

La zona superior izquierda es la zona "ideal", donde la tasa de falsos positivos es cero, y la tasa de positivos verdaderos es uno.