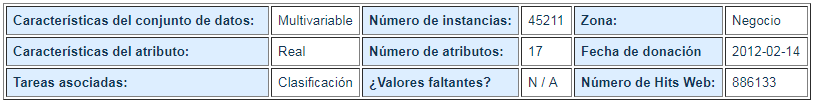
**Conjunto de datos de marketing bancario**

**Resumen:** Los datos están relacionados con las campañas de marketing directo (llamadas telefónicas) de una institución bancaria portuguesa. El objetivo de la clasificación es predecir si el cliente suscribirá un depósito a plazo (variable y).



Información del atributo:

Variables de entrada:

# datos del cliente del banco:

1 - edad (numérica)

2 - trabajo: tipo de trabajo (categórico: 'administrador', 'obrero', 'empresario', 'criada', 'administración', 'retirado' , 'trabajo por cuenta propia', 'servicios', 'estudiante', 'técnico', 'desempleado', 'desconocido')

3 - marital: estado civil (categórico: 'divorciado', 'casado', 'soltero', 'desconocido' '; nota:' divorciado 'significa divorciado o viudo)

4 - educación (categórico:' basic.4y ',' basic.6y ',' basic.9y ',' high.school ',' analfabeto ',' professional.course ',' university.degree ',' unknown ')

5 - default: ¿el crédito está en default? (categórico: 'no', '

sí ',' desconocido ') 6 - vivienda: ¿tiene un préstamo para vivienda? (categórico: 'no', 'sí', 'desconocido')

7 - préstamo: ¿tiene un préstamo personal? (categórico: 'no', 'sí', 'desconocido')

# relacionado con el último contacto de la campaña actual:

8 - contacto: tipo de comunicación de contacto (categórico: 'celular', 'teléfono')

9 - mes: último mes de año de contacto (categórico: 'jan', 'feb', ' mar ', ...,' nov ',' dec ')

10 - day\_of\_week: último día de contacto de la semana (categórico:' mon ',' tue ',' wed ',' thu ',' fri ')

11 - Duración: duración del último contacto, en segundos (numérico). Nota importante: este atributo afecta mucho al objetivo de salida (por ejemplo, si duration = 0, entonces y = 'no'). Sin embargo, la duración no se conoce antes de que se realice una llamada. Además, después del final de la llamada y obviamente se conoce. Así, esta entrada solo debe incluirse para fines de referencia y debe descartarse si la intención es tener un modelo predictivo realista.

# otros atributos:

12 - campaña: número de contactos realizados durante esta campaña y para este cliente (numérico, incluye el último contacto)

13 - días: número de días que pasaron después de que el cliente fue contactado por última vez de una campaña anterior (numérico; 999 significa que el cliente no estaba contactado previamente)

14 - anterior: número de contactos realizados antes de esta campaña y para este cliente (numérico)

15 - poutcome: resultado de la campaña de marketing anterior (categórico: 'falla', 'inexistente', 'éxito')

# social y económico atributos de contexto

16 - tasa de emp.var .: tasa de variación del empleo - indicador trimestral (numérico)

17 - cons.price.idx: índice de precios al consumidor - indicador mensual (numérico)

18 - cons.conf.idx: índice de confianza del consumidor - indicador mensual (numérico)

19 - euribor3m: euribor tasa de 3 meses - indicador diario (numérico)

20 - nr.empleados: número de empleados - indicador trimestral (numérico)

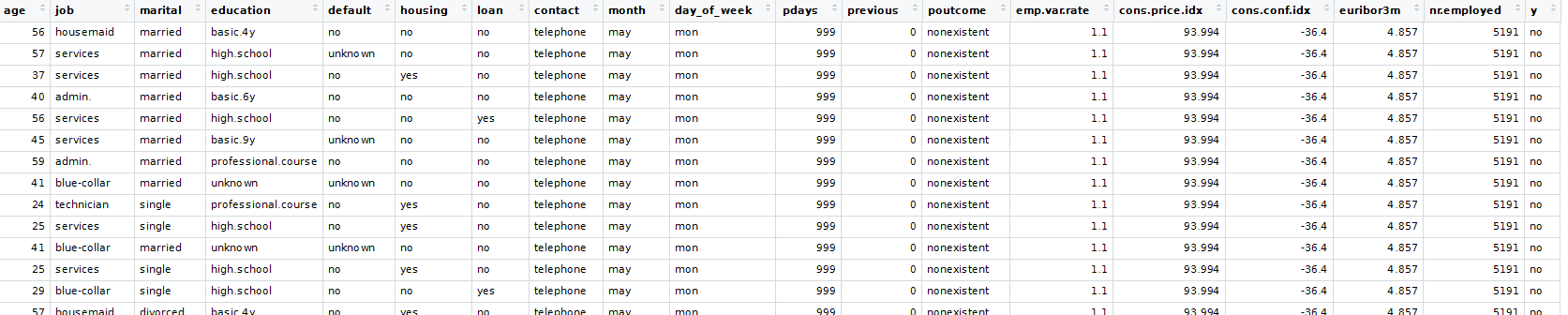
Variable de salida (deseada objetivo):

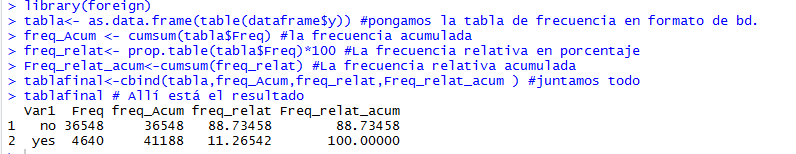
21 - y - ¿el cliente ha suscrito un depósito a plazo? (binario: 'sí', 'no')

Fuente: [https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#](https://archive.ics.uci.edu/ml/datasets/Bank+Marketing)

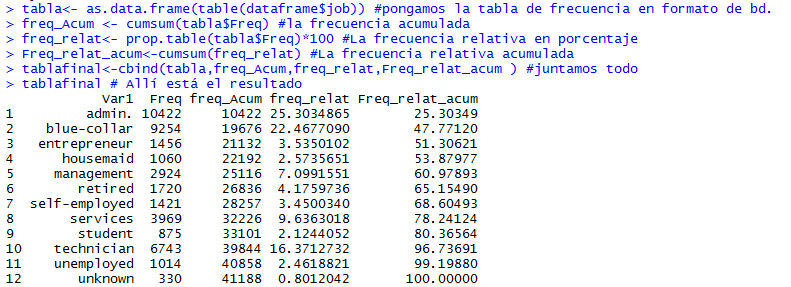
**Análisis RStudio**

Análisis exploratorio R

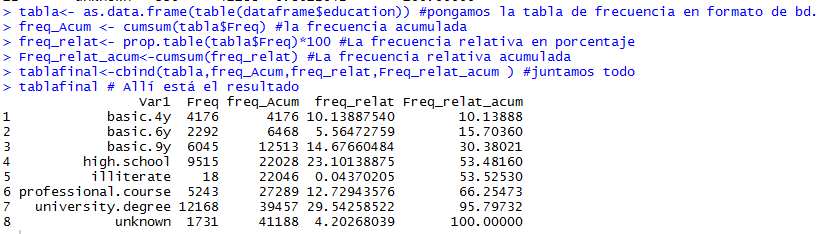




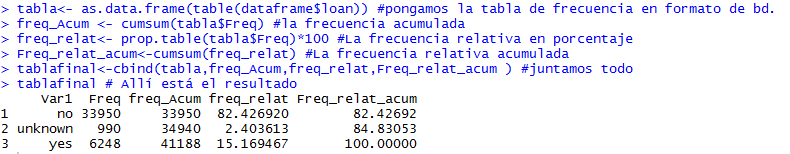
Se puede decir que el 88.73% de los clientes no se ha suscrito a un depósito de largo plazo, contra un 11.26% de clientes que si lo hicieron.



Se puede decir que el trabajo con más frecuencia es el cargo admin dado que compone el 25.30% de los trabajos de toda la muestra, también se puede decir que entre admin,blu-collar y entrepreneur compone la mitad de los trabajos con mayor frecuencia con el 51.30%



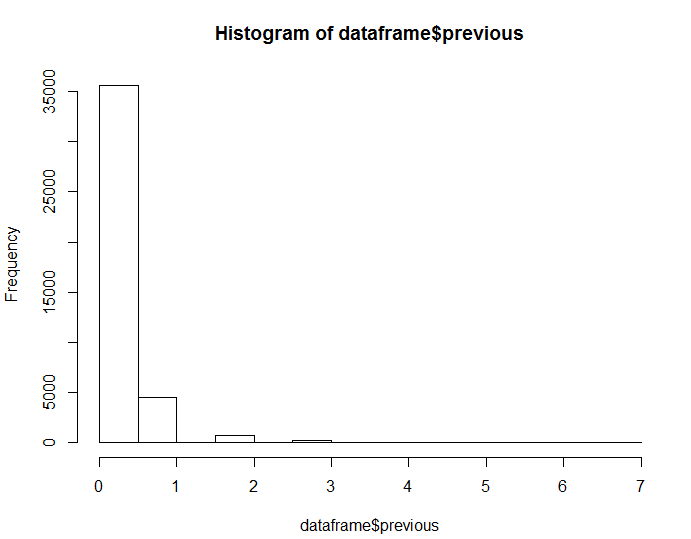
Se puede decir que los clientes con educación universitaria compone el 29.54%



Se puede decir que el 82.42% de los clientes no tienen prestamos de libre inversión.



Se puede decir que la media de los clientes se sitúa entre 40 años, también se observa clientes con edad atípica por encima de 70 años, también se observa que existen clientes con edad mínima de 18 años.



Previous: número de contactos realizados antes de esta campaña y para este cliente (numérico)

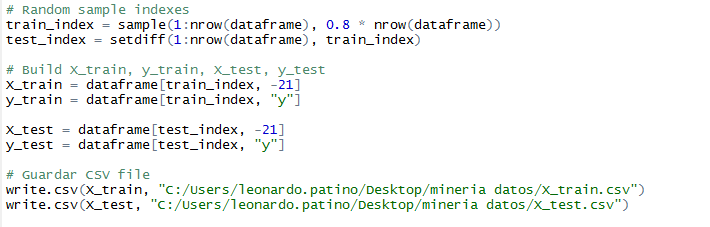
Se puede decir que a la gran mayoría de clientes no se les hicieron llamadas,

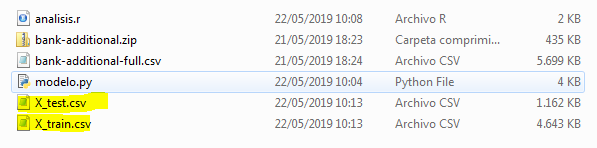
**MODELADO Python (sklearn)**

Para simular un conjunto de pruebas y trenes,

Dividiremos aleatoriamente estos datos en 80% de prueba y 20% de prueba.

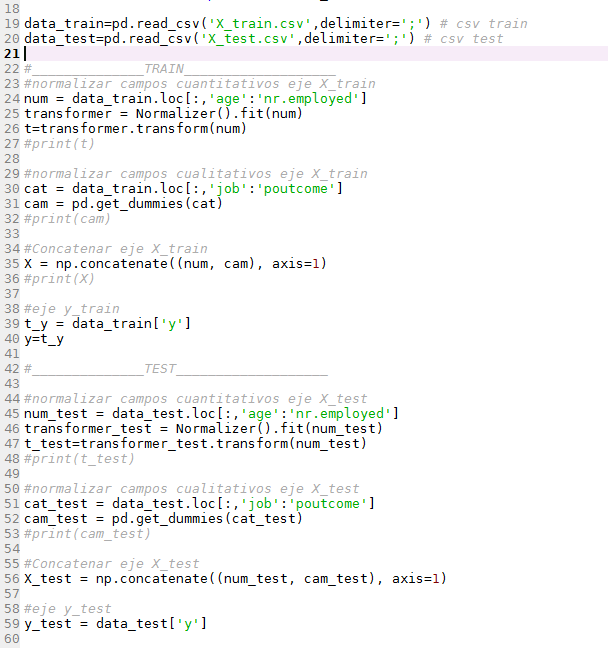
Para evitar que la información tenga un sesgo en la partición





**Normalización**

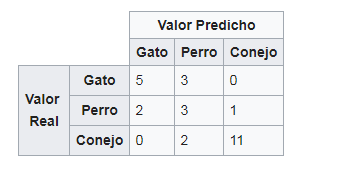
En muchos casos, es útil normalizar los datos para hacerle más fácil a la técnica de machine learning el aprendizaje. Por normalizar nos referimos a poner a todos los datos en una escala similar.

Se normalizan las variables cuantitativas con: Normalizer().fit(), y las variables cualitativas con: pd.get\_dummies()

**Matriz confusión**

es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en [aprendizaje supervisado](https://es.wikipedia.org/wiki/Aprendizaje_supervisado). Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

Si en los datos de entrada el número de muestras de clases diferentes cambia mucho la tasa de error del clasificador no es representativa de lo bien que realiza la tarea el clasificador. Si por ejemplo hay 990 muestras de la clase 1 y sólo 10 de la clase 2, el clasificador puede tener fácilmente un sesgo hacia la clase 1. Si el clasificador clasifica todas las muestras como clase 1 su precisión será del 99%. Esto no significa que sea un buen clasificador, pues tuvo un 100% de error en la clasificación de las muestras de la clase 2.



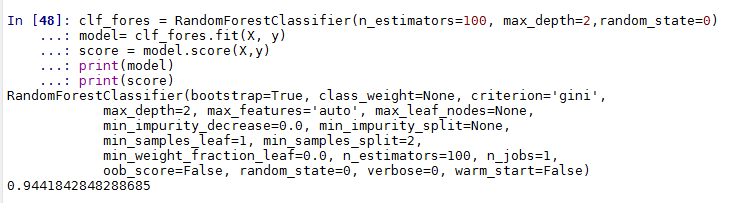
**Modelo 1**

Random forest

Un clasificador de bosque aleatorio.

Un bosque aleatorio es un meta estimador que se ajusta a una serie de clasificadores de árboles de decisión en varias submuestras del conjunto de datos y utiliza el promedio para mejorar la precisión predictiva y el ajuste excesivo del control. El tamaño de la sub-muestra siempre es el mismo que el tamaño de la muestra de entrada original, pero las muestras se extraen con reemplazo si bootstrap=True(predeterminado).

Este utiliza el promedio para mejorar la precisión predictiva y el ajuste excesivo del control.



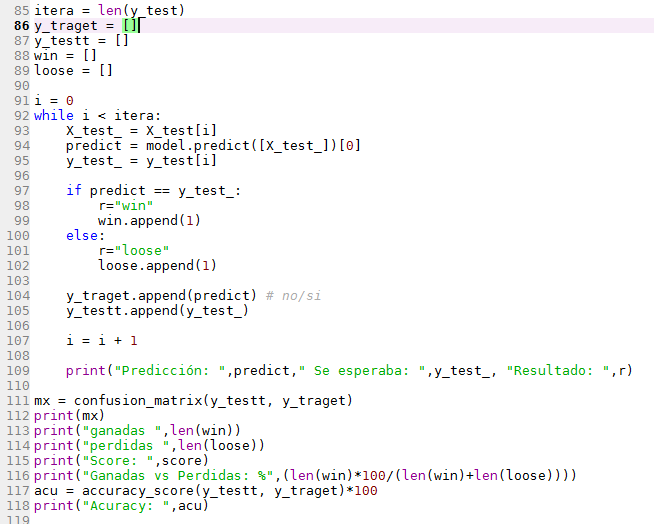
**Parametros:**

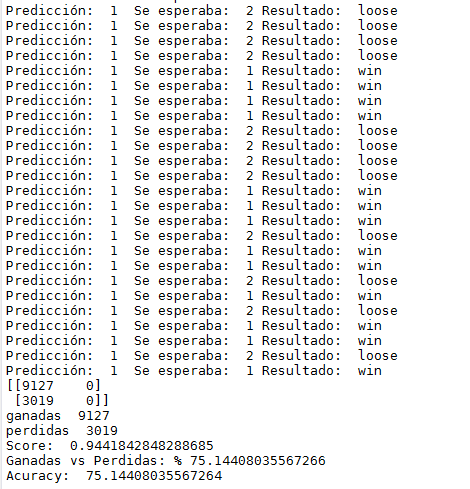
**n\_estimadores:** El número de árboles en el bosque.

**max\_depth:** La máxima profundidad del árbol. Si Ninguno, entonces los nodos se expanden hasta que todas las hojas estén puras o hasta que todas las hojas contengan menos muestras min\_samples\_split.

**random\_state:** Si int, random\_state es la semilla utilizada por el generador de números aleatorios; Si la instancia de RandomState, random\_state es el generador de números aleatorios; Si Ninguno, el generador de números aleatorios es la instancia de RandomState utilizada por np.random.

Predecir datos no vistos y\_test





**Modelo 2**

Redes neuronales

Las neuronas tienen conexiones, sinapsis con entradas, Son capaces de aprender de manera jerarquizada es decir la información se aprende por niveles donde las primeras capas se aprenden cosas concretas como que es una rueda etc, y las posteriores se utilizan para aprender conceptos más abstractos como que es una moto un carro etc.

A medida que hay más capas la información que se aprende es más abstracta e interesante.

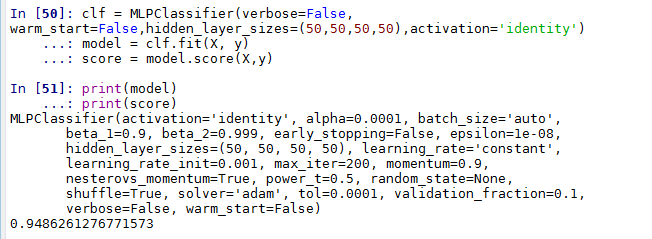
**Clasificador Perceptron multicapa.**

Este modelo optimiza la función de pérdida de registro utilizando LBFGS o descenso de gradiente estocástico.

MLPClassifier entrena iterativamente ya que en cada paso del tiempo se calculan las derivadas parciales de la función de pérdida con respecto a los parámetros del modelo para actualizar los parámetros.

También puede tener un término de regularización agregado a la función de pérdida que reduce los parámetros del modelo para evitar el sobreajuste.

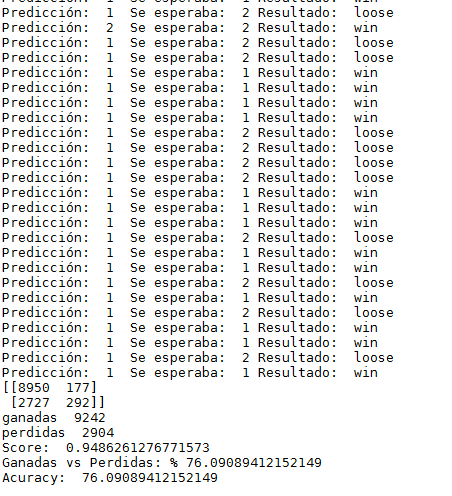
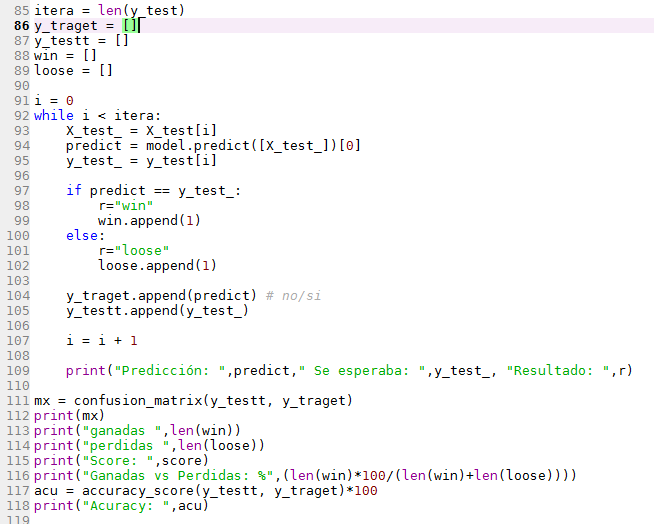
Esta implementación funciona con datos representados como matrices de números densos o matrices dispersas de valores de punto flotante.



**Parámetros:**

**hidden\_layer\_sizes:** Representa el número de neuronas en la capa oculta.

**Activación: *{'identidad', 'logística', 'tanh', 'relu'}, por defecto 'relu'***

* 'Identidad', activación sin operación, útil para implementar cuellos de botella ineales, devuelve f (x) = x
* 'logistic', la función sigmoide logística, devuelve f (x) = 1 / (1 + exp (-x)).
* 'tanh', la función tan hiperbólica, devuelve f (x) = tanh (x).'relu', la función de unidad lineal rectificada, devuelve f (x) = max (0, x)

**Modelo 3**

Regresión logística

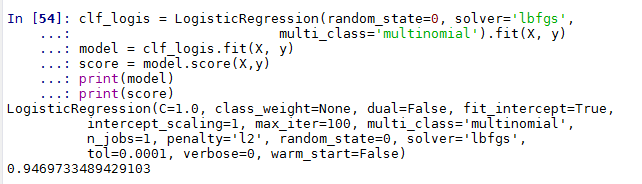
Es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica (una variable que puede adoptar un número limitado de categorías) en función de las variables independientes o predictoras. Es útil para modelar la [probabilidad](https://es.wikipedia.org/wiki/Probabilidad) de un evento ocurriendo como función de otros factores. El análisis de regresión logística se enmarca en el conjunto de [Modelos Lineales Generalizados (GLM por sus siglas en inglés)](https://es.wikipedia.org/wiki/Modelo_lineal_generalizado) que usa como función de enlace la función [logit](https://es.wikipedia.org/wiki/Logit" \o "Logit). Las probabilidades que describen el posible resultado de un único ensayo se modelan, como una función de variables explicativas, utilizando una función logística.

Clasificador de regresión logística (también conocido como logit, MaxEnt).

En el caso multiclase, el algoritmo de entrenamiento usa el esquema de uno contra descanso (OvR) si la opción 'multi\_class' está configurada en 'ovr', y usa la pérdida de entropía cruzada si la opción 'multi\_class' está configurada en 'multinomial '. (Actualmente, la opción 'multinomial' solo es compatible con los solucionadores 'lbfgs', 'sag', 'saga' y 'newton-cg').

Esta clase implementa la regresión logística regularizada utilizando las librerías 'liblinear', 'newton-cg', 'sag', 'saga' y 'lbfgs'. **Tenga en cuenta que la regularización se aplica por defecto** . Puede manejar tanto entradas densas como dispersas. Utilice matrices ordenadas por C o matrices CSR que contengan flotadores de 64 bits para un rendimiento óptimo; Cualquier otro formato de entrada será convertido (y copiado).

Los solucionadores 'newton-cg', 'sag' y 'lbfgs' solo admiten la regularización L2 con una formulación primaria, o ninguna regularización. El solver 'liblinear' es compatible con la regularización de L1 y L2, con una formulación dual solo para la penalización de L2. La regularización de Elastic-Net solo es compatible con el solucionador 'saga'.



**Parametros:**

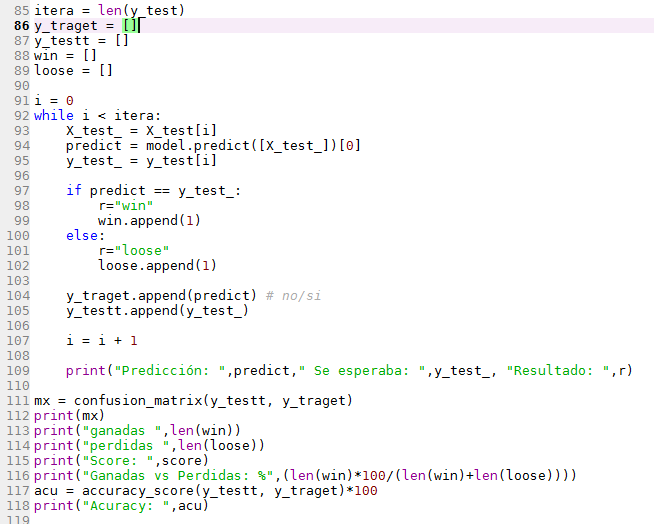
**random\_state:** La semilla del generador de números pseudoaleatorios para usar cuando se barajan los datos. Si int, random\_state es la semilla utilizada por el generador de números aleatorios; Si la instancia de RandomState, random\_state es el generador de números aleatorios; Si Ninguno, el generador de números aleatorios es la instancia de RandomState utilizada por np.random. Se utiliza cuando solver== 'sag' o 'liblinear'.

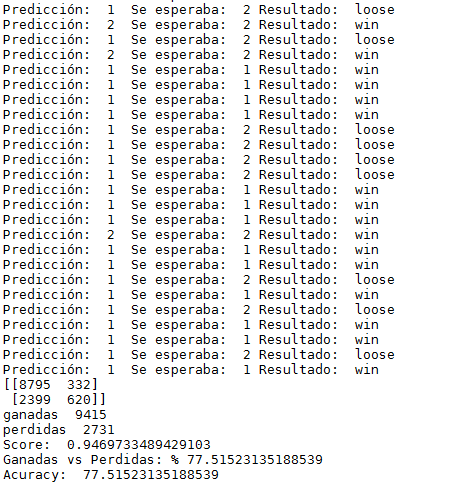
**solucionador :**

* Para conjuntos de datos pequeños, 'liblinear' es una buena opción, mientras que 'sag' y 'saga' son más rápidos para los grandes.
* Para problemas multiclase, solo 'newton-cg', 'sag', 'saga' y 'lbfgs' manejan la pérdida multinomial; 'liblinear' está limitado a esquemas de uno contra descanso.
* 'newton-cg', 'lbfgs', 'sag' y 'saga' manejan L2 o sin penalización
* 'liblinear' y 'saga' también manejan la penalización L1
* 'saga' también soporta penalización 'elasticnet'
* 'liblinear' no maneja ninguna penalización

Tenga en cuenta que la convergencia rápida 'sag' y 'saga' solo está garantizada en características con aproximadamente la misma escala. Puede preprocesar los datos con un escalador desde sklearn.preprocessing.

**multi\_class :** Si la opción elegida es 'ovr', entonces un problema binario es adecuado para cada etiqueta. Para "multinomial", la pérdida minimizada es la pérdida multinomial que se adapta a toda la distribución de probabilidad, *incluso cuando los datos son binarios* . 'multinomial' no está disponible cuando solver = 'liblinear'. 'auto' selecciona 'ovr' si los datos son binarios, o si solver = 'liblinear', y de otro modo selecciona 'multinomial'.





**Conclusión**

**Modelo1.Random forest**

Score: 94.41%

Acuracy: 75.14%

**Modelo2.Redes neuronales**

Score: 94.86%

Acuracy: 76.09%

**Modelo3.Regresión logistic**

Score: 94.41%

Acuracy: 77.51%