

**INTELIGENCIA ARTIFICIAL**  
  
**TRABAJO PRÁCTICO OBLIGATORIO GRUPAL Nº1**  
**“Machine Learning (ML) - Marketing”**

| Turno: Noche | | | |
| --- | --- | --- | --- |
| Integrantes: | | | |
| 1. | Di Fresco, Lucas | | LU: |
| 2. | Gugliemone, Lucas | | LU: 1110022 |
| 3. | Lacuesta, Gastón Axel | | LU: 1117695 |
| 4. | Rondan, Ignacio | | LU: 1068437 |
| 5. | Orozo, Patricio | | LU: 1058382 |
| 6. | Vera, Samira | | LU: 1101684 |
| Profesor: Parkinson, Christian | | | |
| Fecha: 28/11/2022 | | Cuatrimestre: 2C 2022 | |

[**Identificación**](#_heading=h.jc3h5ypp2fxd) **3**

[Variables y clasificación](#_heading=h.li1vqjw8rb59) 3

[**Pre-procesamiento**](#_heading=h.k0pt64mng83s) **5**

[**Modelos**](#_heading=h.qg3gkvfful4h) **10**

[Support Vector Machines – SVM](#_heading=h.pwu66jn8bigi) 10

[Regresión Logística](#_heading=h.4f20rzlq5xmf) 11

[Random Forest](#_heading=h.iq4vn3jmi4df) 12

[**Conclusión**](#_heading=h.ex9mx1gyvoo3) **13**

# Identificación

Se eligió trabajar con el dataset de “Marketing” que contiene registros de encuestados con diferentes atributos de los mismos, como área de trabajo, edad, estado civil, educación, día de la semana y duración de la llamada, entre otros.

El objetivo es predecir si el encuestado va a aceptar la oferta de la campaña.

## Variables y clasificación

Tabla con el tipo de variable de cada uno de los campos del dataset.

| Campo | Tipo de Variable | Descripción |
| --- | --- | --- |
| age | Numérica discreta | Edad del encuestado |
| job | Categórica | Área de trabajo del encuestado |
| marital | Categórica | Estado civil |
| education | Categórica | Nivel educativo del encuestado |
| default | Categórica | “Defecto” – yes, no, unknown – |
| housing | Categórica | Alojamiento |
| loan | Categórica | Préstamo |
| contact | Categórica | Contacto |
| month | Categórica | Mes encuesta |
| day\_of\_week | Categórica | Día de la semana encuesta |
| duration | Numérica discreta | Duración de la comunicación |
| campaign | Numérica discreta | Label de la campaña de marketing |
| pdays | Numérica discreta | Cantidad de días |
| previous | Numérica discreta | “Previo” – cantidad de días |
| poutcome | Categórica | - |
| emp.var.rate | Numérica continua | Emp.var.rate |
| cons.price.idx | Numérica continua | Índice cons.price |
| cons.conf.idx | Numérica continua | Índice Cons.conf |
| euribor3m | Numérica continua | Tasa de interés |
| nr.employed | Numérica discreta | Número de empleado |
| y | Categórica | Resultado campaña |

# Pre-procesamiento

Durante el procesamiento se buscó entender el tipo de dataset analizado, así como la eliminación de registros duplicados y la detección de variables categóricas y numéricas

Imports

Text

Description automatically generated

Lectura de dataset

A screenshot of a computer

Description automatically generated with medium confidence

En este paso se descartan 252 registros duplicados

Graphical user interface, text, application

Description automatically generated

Aquí se observan los dtypes de los fields del dataset, esto es para detectar variables categóricas y numéricas. Aunque al ser todos object se hizo un relevamiento manual.

Graphical user interface, text

Description automatically generated

Aquí se visualizan histogramas de las variables numéricas:

Chart, waterfall chart

Description automatically generated

Tareas de preprocesamiento de preparación del dataset para el posterior procesamiento de los modelos. Se determina el tipo “category” para el field job y se obtienen los dummies de las variables categóricas para ajustar dichas variables a numéricas. Se toma como datos X e y, las dos variantes del campo “yes” – que determina si un encuestado aceptó la campaña o no. Se prepara como 1D array ya que los algoritmos posteriores lo requieren.

Se genera el train\_test\_split con el dataset, obteniendo los datos de entrenamiento y de testeo.

Text

Description automatically generated

Se escalan las variables con MinMaxScaler para obtener valores significativos entre 0-1.

Graphical user interface, text

Description automatically generated

Se plotea el dataset escalado junto al dataset original para ver que el escalamiento se realizó correctamente, podemos ver que el dataset escalado se observa entre rangos 0-1 correctamente sin perder la distribución de los elementos en el plot.

A picture containing text

Description automatically generated

# Modelos

## Support Vector Machines – SVM

Para el cálculo de SVM se utilizó sklearn.svm con un C=100, kernel fbs (por defecto en sklearn) en primera instancia, generando un accuracy del set de prueba del 0.90.

Graphical user interface, text

Description automatically generated

Al cambiar el tipo de Kernel a “linear”, se nota una pequeña mejora en el accuracy, pero no se mostraron mejoras en el caso de cambiar el resto de los parámetros, por ejemplo C=1000, C=100, C=0.1.

A screenshot of a computer

Description automatically generated with medium confidence

El parámetro C mide el error de clasificación del algoritmo en cada elemento de la muestra.

## Regresión Logística

En todos los casos se utilizó sklearn como librería para la regresión logística. En primera instancia se ejecutó con los parámetros por defecto generando el siguiente output:

Text

Description automatically generated

Para la regresión logística también utilizamos sklearn con un random\_state de 1 y solver saga, aunque la precisión del modelo se redujo, lo cual en este caso es favorable al alejarse del overfitting (<0.90).

Text

Description automatically generated

## Random Forest

Para Random Forest se ejecutaron pruebas y error para encontrar los mejores hiper parámetros para el Search Grid y comparar los resultados.

Se especificaron max\_features, max\_depth y criterions “entropy” y “gini” obteniendo los sig. resultados:

| Text  Description automatically generated | Graphical user interface, application  Description automatically generated |
| --- | --- |

Se observa que los mejores hiperparametros con max\_features=40, entropy como criterion y un max\_depth de 10.

| * **recall\_score**: se puede intuir como la habilidad del clasificador para encontrar muestras positivas. * **precision\_score**: se puede intuir como la habilidad del clasificador para no determinar como positiva una muestra que en realidad es negativa. * **f1\_score**: es el punto armónico mediano entre la precisión y el recall de un modelo. |
| --- |

Dados los valores de f1, recall, precisión y el mejor puntaje (0.91) se puede observar que el modelo está performando bien, pero podría estar cayendo en overfitting con el set de datos utilizado.

# Conclusión

Para la aplicación de modelos se eligieron modelos de relativa simpleza debido al conocimiento de machine learning que hay en el equipo, además de tener en cuenta que un nivel de complejidad superior pueden generar escenarios de overfitting. Es por ello que decidimos utilizar Support Vector Machines, Regresión Logística y Random Forests para tener variados algoritmos y obtener mayor generalización del modelo.

Random Forest es un método más complejo de los tres, lo cual significa un mayor riesgo de caer en overfitting, además de que la regresión logística y el SVM suelen performar mejor que el RF.

La regresión logística es óptima para la resolución de problemas de este tipo (e.g, detectar si una persona va a aceptar (1, yes) o rechazar (0, no) una oferta de cualquier tipo, pero tiene problemas de caer en overfitting. Mientras que SVM se utiliza tanto para clasificación como regresión y es menos propenso a sufrir overfitting, también se basa más en los aspectos geométricos mientras que la RL se basa más en aspectos estadísticos.

Comparando los tres modelos aplicados, sus resultados, y conociendo sus ventajas y desventajas se podría decir que el resultado final es que se debería mejorar el modelo debido a que está sufriendo de overfitting.