Privatización del espacio

Natalia Benitez
Tomas Romero Mancinelli
Pablo Parlatore Siritto

link a presentación

Tabla de Contenidos

Descripción del caso de negocio

El espacio exterior junto con sus misterios siempre ha ocupado un importante campo de investigación en la historia de la humanidad. Estudiar los astros, entender que materiales los conforman, cómo afectan el resto de los planetas y cuerpos celestes a nuestro Globo, todas incógnitas que poco a poco se fueron respondiendo, pero siempre a cientos de miles de kilómetros de distancia.

Con el avance de la tecnología la sociedad se fue acercando cada vez más a la posibilidad de viajar al espacio exterior para seguir investigando lo desconocido y hasta viendo la posibilidad de explorar nuevos planetas para vivir en ellos. Es por esto, que a inicios de la década de 1960 los dos países más poderosos de ese entonces, Estados Unidos de América y la Unión Soviética, comenzaron una carrera por ser los primeros en enviar al hombre en conquista de lo desconocido.

Por décadas estas investigaciones bajaron su intensidad, empezando a crecer el número de misiones financiadas por fondos privados interesados en la temática y la posibilidad de iniciar una nueva vida en el espacio exterior ante la posibilidad de una catástrofe que impida la supervivencia del ser humano en la Tierra.

Hoy en día, los dos hombres más ricos del mundo están enfrentados en una reñida carrera espacial mundial, compitiendo por contratos con agencias gubernamentales y empresas privadas.

Ante esta gran cantidad de exploraciones que se están llevando a cabo, surgió la necesidad de comprender el desarrollo en la historia de esta temática, como así también cuales pueden ser las **causales de éxito de una misión.**

Tabla de versionado

Nombre	Version	Fecha	Comentarios
• Space_E1	1.0	28-2-2022	Recopilación de data, análisis primarios de dataset y algoritmos de clasificación supervisados.
• Space_E2	2.0	8-3-2022	Se agregan las métricas de los diversos algoritmos utilizados.
• Space_E3	3.0	24-3-2022	Mejora de los hiperparametros para KNN y Random Forest y suma de modelos de ensamble tipo boosting
• Space_E4	4.0	15-4-2022	Aplicación de hyperparameter tuning mediante el utilizo de GridSearchCV a todos los modelos.

Objetivos del modelo

El objetivo del modelo es lograr **predecir el resultado de una misión** conforme a las variables de entradas: País, PBI del país, % del PBI destinado a militarización, Fondos Privados o Públicos y Compañía que lleva a cabo la producción del cohete.

Para poder realizar esto vamos a utilizar un algoritmo de clasificación, el cual dirá si nuestra variable Target, ‘Status Mission’, es ‘Success’ o ‘Failure’.

Descripción de los datos

- Los datasets elegidos fueron dos: los lanzamientos espaciales globales y un dataset con datos de los países (PBI, gasto militar, etc.) adecuados al enfoque financiero del proyecto.
- Dataset 1 : contiene detalle de los lanzamientos realizados desde 1957 a 2019, empresas que participaron y sus respectivos países de origen, lugar de lanzamiento, resultado de la misión, estado actual de actividad de los cohetes.
- Dataset 2: contiene detalle del PBI anual de cada país, su población y del porcentaje del PBI destinado a gastos militares.

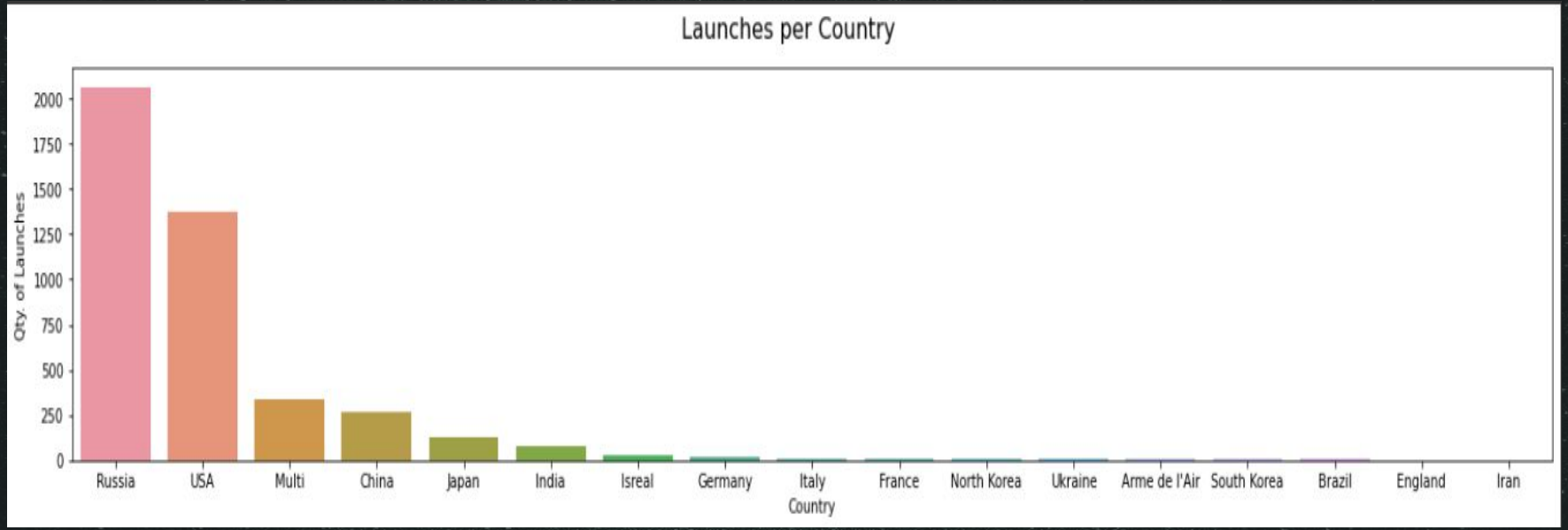
Descripción de los datos

Company name	Nombre de la compañía a cargo de la misión
Location	Lugar en donde se realizó el despegue
Detail	Detalle del nombre de la cohete lanzado
Status Rocket	Status actual del cohete, está dividido entre ‘StatusActive’ y ‘StatusRetired’
Status Mission	Resultado de la misión, el resultado es ‘Success’ o ‘Failure’
Country of Launch	País desde el cual se realizó el lanzamiento de la misión
Country	País de orígen de la empresa a cargo de la misión
Private or State Run	Tipo de empresa a cargo de la misión, estatal ‘S’ o privada ‘P’
Year	Año en que se realizó el despegue
Military expenditure (% of GDP)	% del PBI destinado a la milicia
GDP (current US$)	Producto Bruto Interno del país
Population, total	Población total del País

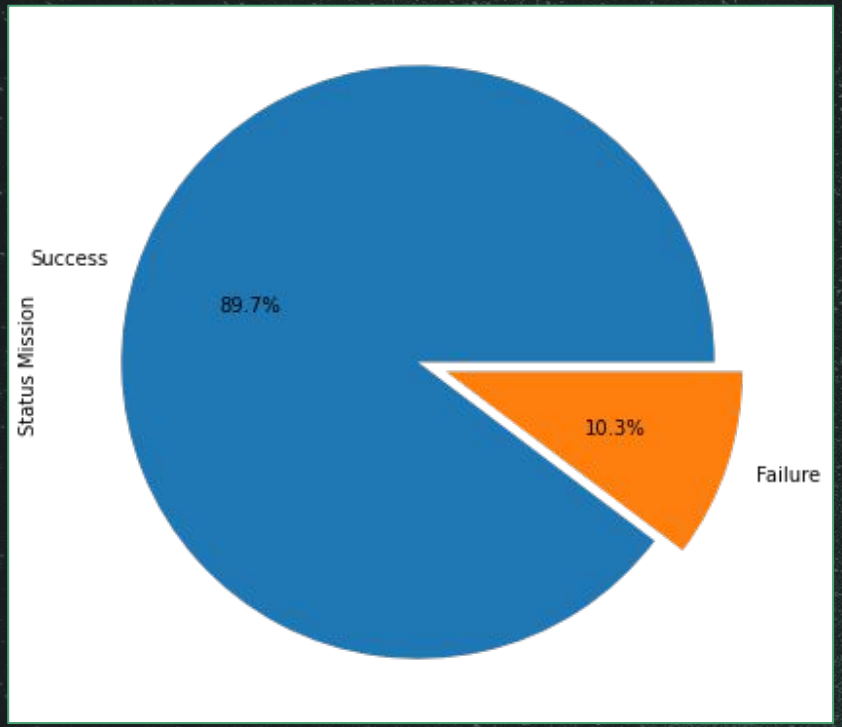
Hallazgos encontrados por el EDA

Cantidad de misiones por País:

A simple vista se puede notar la que la participación de Rusia y USA en comparación con el resto de los países es ampliamente mayor, esto está relacionado con la carrera al espacio que se llevó a cabo en la guerra fría, en la que ambos países luchaban por ser los primeros en llegar a la luna.



Status Mission (Variable Target): Cantidad de misiones fallidas y exitosas

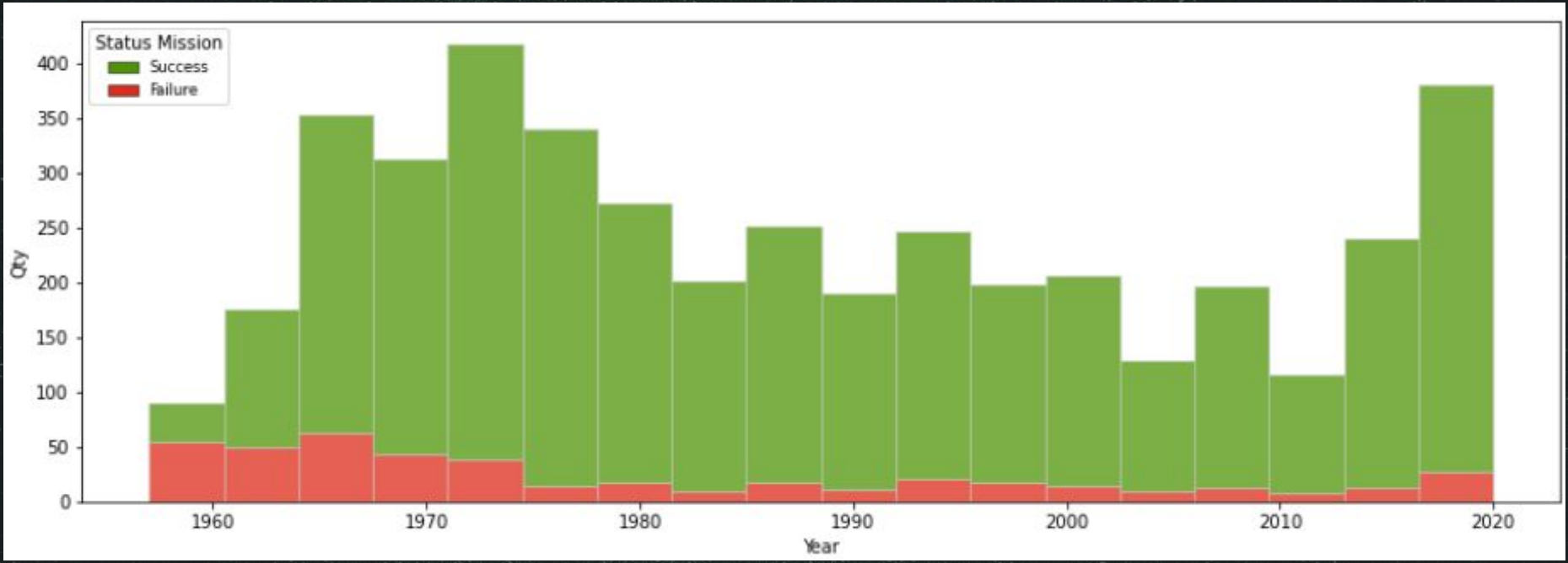


- Como se puede observar en el gráfico representado a la izquierda el porcentaje de misiones exitosas es ampliamente mayor al de las misiones fallidas.
- Así también se puede ver esta diferencia en los número absolutos de misiones exitosas y misiones fallidas.

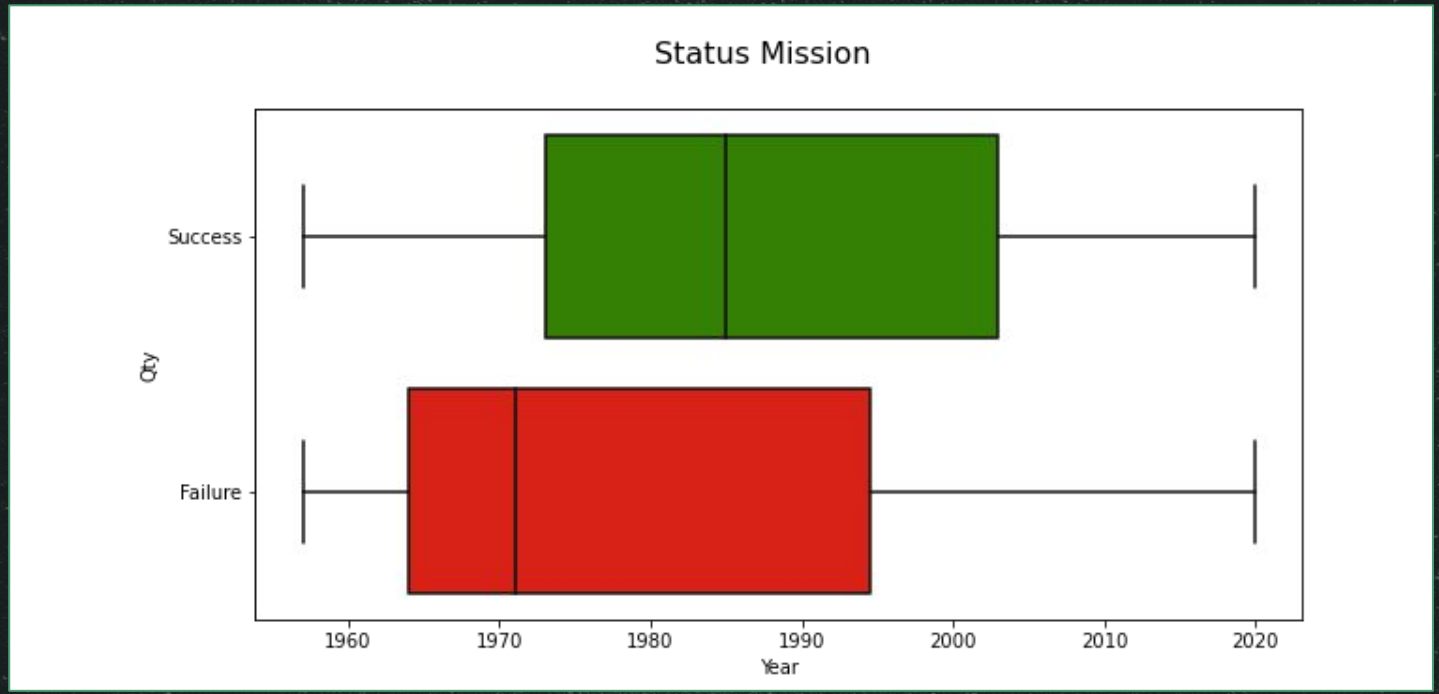
Success	3879
Failure	445

Con los años hay una menor cantidad de misiones fallidas lo cual está relacionado con una curva de aprendizaje y mejora de la tecnología.

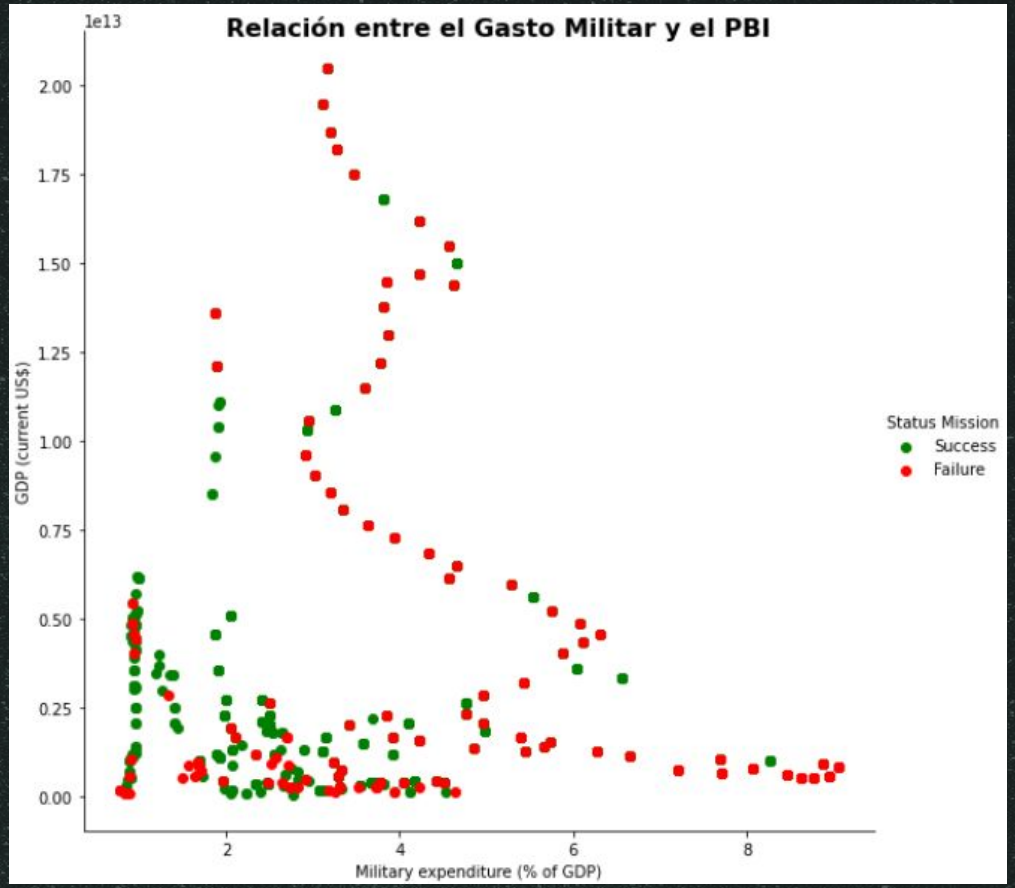
En el año 2020 se puede ver un aumento en la cantidad de misiones fallidas y esto tiene que ver con las nuevas tecnologías de naves sin tripulación que están siendo probadas actualmente.



Realizamos un Boxplot para poder analizar la media del éxito o fracaso en los años. En la medida del avance del tiempo la media de resultados exitosos fue creciendo.



No existe relación directa del éxito de una misión con el PBI o el gasto militar de cada país.



Elección de algoritmo.

Como se ha dicho anteriormente, **el principal objetivo de nuestra investigación es poder determinar,** mediante el uso de ciertas variables económicas y geopolíticas, **si la misión tendrá éxito o no.**

Al tener nuestra variable target tan clara y contar con información de ella en nuestro Dataset (Satus Mission), nos enfocamos en el uso de algoritmos supervisados dejando de lado los algoritmos de clustering no supervisados.

Ante la poca cantidad de variables numéricas con las que contamos no fue necesario realizar un PCA, para disminuir las variables y hacer un análisis de las variables principales.

En esta primer etapa se llevó a cabo un análisis de diversos tipos de algoritmos supervisados. Fueron analizados mediante el uso de diversas Métricas que detallaremos y explicaremos en breve.

Métricas

Cuando necesitamos evaluar el rendimiento en clasificación, podemos usar las métricas de precisión, recall, F1, accuracy y la matriz de confusión. Vamos a explicar cada uno de ellos.

Accuracy: La exactitud (accuracy) mide el porcentaje de casos que el modelo ha acertado.

Precisión: Con esta métrica podemos medir la calidad del modelo de machine learning en tareas de clasificación. ¿Qué porcentaje de los que hemos dicho que son la clase positiva, en realidad lo son?

Recall: La métrica de exhaustividad nos va a informar sobre la cantidad que el modelo de machine learning es capaz de identificar. ¿Qué porcentaje de la clase positiva hemos sido capaces de identificar?

F1: El valor F1 se utiliza para combinar las medidas de precision y recall en un sólo valor. Esto es práctico porque hace más fácil el poder comparar el rendimiento combinado de la precisión y la exhaustividad entre varias soluciones.

Matriz de confusión: Una matriz de confusión, es una tabla resumida que se utiliza para evaluar el rendimiento de un modelo de clasificación. El número de predicciones correctas e incorrectas se resumen con los valores de conteo y se desglosan por cada clase.

ARBOL DE DECISIONES

Los árboles de decisión son una técnica de aprendizaje automático supervisado. Como su nombre indica, esta técnica de machine learning toma una serie de decisiones en forma de árbol. Los nodos intermedios (las ramas) representan soluciones. Los nodos finales (las hojas) nos dan la predicción que vamos buscando

Se realizó el **entrenamiento** del mismo con un **75% de los datos**, dejando el restante **25% para llevar a cabo los testeos.**

Métricas de Desempeño del Modelo:

Accuracy:

- % de aciertos sobre el set de evaluación: 89.92%

Precisión:

- La precisión del algoritmo es de: 89.92%

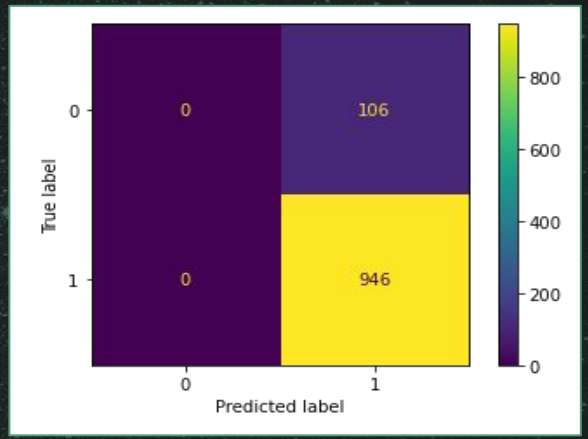
Recall:

- El resultado del Recall del algoritmo es de: 100%

F1 Score:

- El resultado del F1 del algoritmo es de: 94.69%

Matriz de confusión:



RANDOM FOREST

Es un modelo de ensamble de tipo Bagging que ayuda a mejorar los resultados de las métricas obtenidos por el algoritmo de Árbol de decisión.

El algoritmo de Random Forest es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.

Métricas de Desempeño del Modelo:

Accuracy:

- % de aciertos sobre el set de evaluación: 90,11%

Precisión:

- La precisión del algoritmo es de: 90.71%

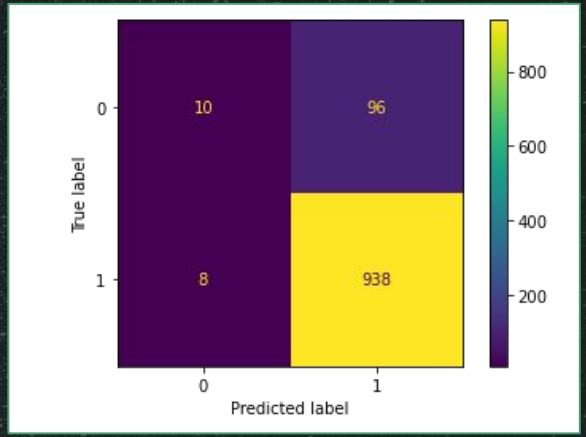
Recall:

- El resultado del Recall del algoritmo es de: 99.15%

F1 Score:

- El resultado del F1 del algoritmo es de: 94.75%

Matriz de confusión:



KNN

El método de los k vecinos más cercanos es un método de clasificación supervisada que estima el valor de que un elemento x, pertenezca a la clase C, a partir de la información proporcionada por el conjunto de prototipos.

Para poder obtener los mejores resultado en este algoritmo utilizamos GridSearchCV, que es una clase disponible en scikit-learn que permite evaluar y seleccionar de forma sistemática los parámetros de un modelo. Indicando un modelo y los parámetros a probar, puede evaluar el rendimiento del primero en función de los segundos mediante validación cruzada.

Métricas de Desempeño del Modelo:

Accuracy:

- % de aciertos sobre el set de evaluación: 89.16%

Precisión:

- La precisión del algoritmo es de: 90.38%

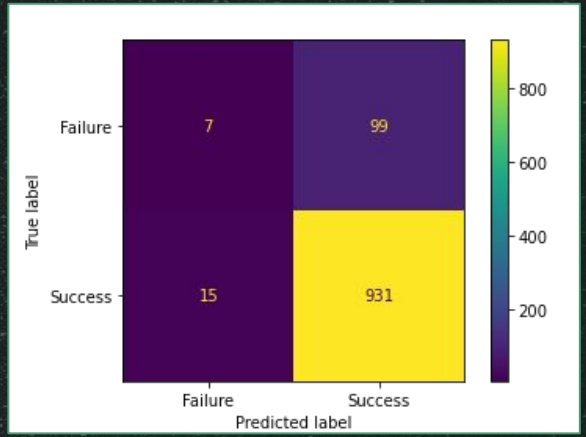
Recall:

- El resultado del Recall del algoritmo es de: 98.41%

F1 Score:

- El resultado del F1 del algoritmo es de: 94.23%

Matriz de confusión:



ADABOOST

El modelo de Adaboost, consiste en crear varios predictores sencillos en secuencia, de tal manera que el segundo ajuste bien lo que el primero no ajustó, que el tercero ajuste un poco mejor lo que el segundo no pudo ajustar y así sucesivamente

Métricas de Desempeño del Modelo:

Accuracy:

- % de aciertos sobre el set de evaluación: 88.59%

Precisión:

- La precisión del algoritmo es de: 90.25%

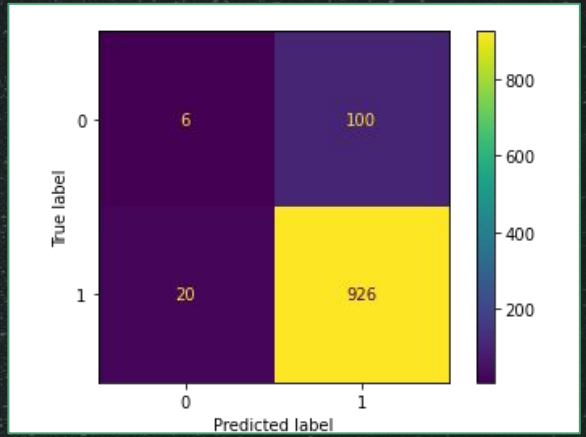
Recall:

- El resultado del Recall del algoritmo es de: 97.88%

F1 Score:

- El resultado del F1 del algoritmo es de: 93.91%

Matriz de confusión:



XGBOOST

Extreme Gradient Boosting, es uno de los algoritmos de machine learning de tipo supervisado más usados en la actualidad. Este algoritmo se caracteriza por obtener buenos resultados de predicción con relativamente poco esfuerzo, en muchos casos equiparables o mejores que los devueltos por modelos más complejos computacionalmente, en particular para problemas con datos heterogéneos.

Métricas de Desempeño del Modelo:

Accuracy:

- % de aciertos sobre el set de evaluación: 89.54%

Precisión:

- La precisión del algoritmo es de: 90.34%

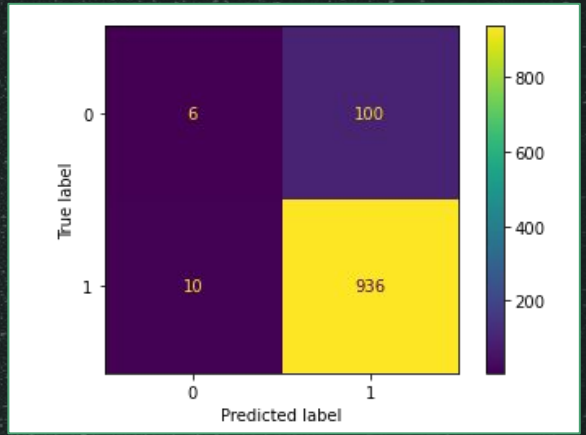
Recall:

- El resultado del Recall del algoritmo es de: 98.94%

F1 Score:

- El resultado del F1 del algoritmo es de: 94.45%

Matriz de confusión:



Conclusiones de primer análisis

Al analizar las métricas de los diversos algoritmos pudimos notar que el más preciso de ellos es Random Forest, presentando un 90.11% de aciertos en la etapa de testeos, y un valor de F1 del 94.75%

Es por eso que como equipo tomamos la decisión de utilizar dicho algoritmo para nuestro proyecto.

	Accuracy	Precisión	Recall	F1
Árbol de decisión	89.92%	89.92%	100%	94.69%
Random Forest ✓	90.11%	90.71%	99.15%	94.75%
Adaboost	88.59%	90.25%	97.88%	93.91%
KNN	89.16%	90.38%	98.41%	94.23%
XGboost	89.54%	90.34%	98.94%	94.45%

Conclusiones de segundo análisis

Luego de aplicar Hyperparameter Tuning, mediante GridSearchCV, comparamos los mejores resultados obtenidos por cada modelo y con qué hiperparámetros, podemos llegar a la conclusión de que el mejor modelo para nuestra data es el XGboost .

Modelo	Mejor Score	Hyper parametros
Árbol de decisión		Los mejores parámetros son:
Random Forest	90.67%	Los mejores parámetros son: {'max_features': 'log2', 'n_estimators': 1000}
Adaboost	90.61%	Los mejores parámetros son: {'learning_rate': 4, 'n_estimators': 5}
KNN	90.64%	Los mejores parámetros son: {'algorithm': 'auto', 'leaf_size': 10, 'n_neighbors': 8, 'weights': 'uniform'}
XGboost	90.86%	Los mejores parámetros son: {'colsample_bytree': 0.8, 'gamma': 2, 'max_depth': 4, 'min_child_weight': 1, 'subsample': 0.6}

Referencias

- https://www.kaggle.com/davidroberts13/one-small-step-for-data
- https://www.kaggle.com/greeshmagirish/worldbank-data-on-gdp-population-and-military?select=API_MS.MIL.TOTL.P1_DS2_en_csv_v2_513199.csv
- https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/
- https://www.analyticslane.com/2018/07/02/gridsearchcv/#:~:text=GridSearchCV%20es%20una%20clase%20disponible,los%20segundos%20mediante%20validaci%C3%B3n%20cruzada.
- https://blog.escueladedatosvivos.ai/como-hacer-optimizacion-parametros-python/