GIVE ME SOME CREDIT:

INTRODUCCIÓN

Para que los mercados funcionen, es necesario que, tanto las personas físicas como las empresas, tengan acceso al crédito, es decir, a financiarse. Por este motivo, las entidades financieras juegan un papel crucial en nuestra economía, ya que son las que deciden quién o quienes pueden obtener financiación y en qué términos van a adquirirla.

Las entidades financieras, como no son adivinas, intentan desarrollar algoritmos de credit scoring, con el fin de poder determinar a base de estadística, matemáticas y una serie de técnicas, la probabilidad de default (impago) o no default de los créditos.

Este caso, nos plantea el problema de llevar a cabo algoritmos para predecir de la mejor manera posible si un crédito va a ser impagado o no.

Para ello, utilizamos una base de datos compuesta por, en este caso, 150000 créditos (muestra de training) indicándose características que se entienden como significativas y si en cada uno de los casos se ha impagado el crédito o no.

Con estos 150000 se entrenarán una serie de logaritmos con el fin de determinar si en nuestra muestra de test (compuesta por 101503 casos), estos algoritmos han sido capaces de acertar el máximo posible de créditos con default y con no default.

Previo a los análisis y realización de algoritmos debemos conocer las variables de las que se compone el dataset:

1- **SeriousDlqin2yrs**: persona con 90 días de morosidad o peor, es decir, "default" (1) y "no default" (0).
2- **RevolvingUtilizationOfUnsecuredLines**: Saldo total en tarjetas de crédito y líneas personales de crédito, excepto bienes inmuebles y deuda a plazos (como préstamos para automóviles) dividido entre la suma de los límites de crédito. Es decir, porcentaje de líneas de circulante en uso.
3- **Age**: Edad del prestatario en años.
4- **NumberOfTime30-59DaysPastDueNotWorse**: Número de veces que el prestatario ha estado de 30 a 59 días vencido, pero sin haber empeorado en los últimos 2 años.
5- **DebtRatio**: Pagos mensuales de deudas, pensión alimenticia...etc, es decir, costes de vida mensuales divididos entre los ingresos brutos mensuales (porcentaje).
6- **MonthlyIncome**: Ingresos mensuales del prestatario.
7- **NumberOfOpenCreditLinesAndLoans**: Número de préstamos en curso (cuotas como préstamos o hipotecas para automóviles) y líneas de crédito (por ejemplo, tarjetas de crédito).
8- **NumberOfTimes90DaysLate**: Número de veces que el prestatario se ha retrasado en el pago de la cuota del préstamo 90 días o más.
9- **NumberRealEstateLoansOrLines**: Número de préstamos hipotecarios e inmobiliarios, incluidas líneas de crédito con garantía hipotecaria.

10- **NumberOfTime60-89DaysPastDueNotWorse**: Número de veces que el prestatario ha estado de 60 a 89 días vencido, pero sin haber empeorado en los últimos 2 años.
11- **NumberOfDependents**: Número de personas dependientes del prestatario en la familia excluyéndose él mismo (cónyuge, hijos, etc.).

Veremos a lo largo del ejercicio, cada uno de los análisis realizados para el mismo los cuales son:

- **ALEATORIA:** estableceremos la probabilidad de default y no default indicando de forma aleatoria un 1 en caso de default y un 0 en caso de no default.

- **REGRESIÓN LOGÍSTICA:** llevaremos a cabo un modelo de regresión logística para predecir la probabilidad de default (1) y no default (0).

- **SUPPORT VECTOR MACHINE (SVM):** en este caso, llevaremos un análisis utilizando soportes y a través del empleo de kernels.

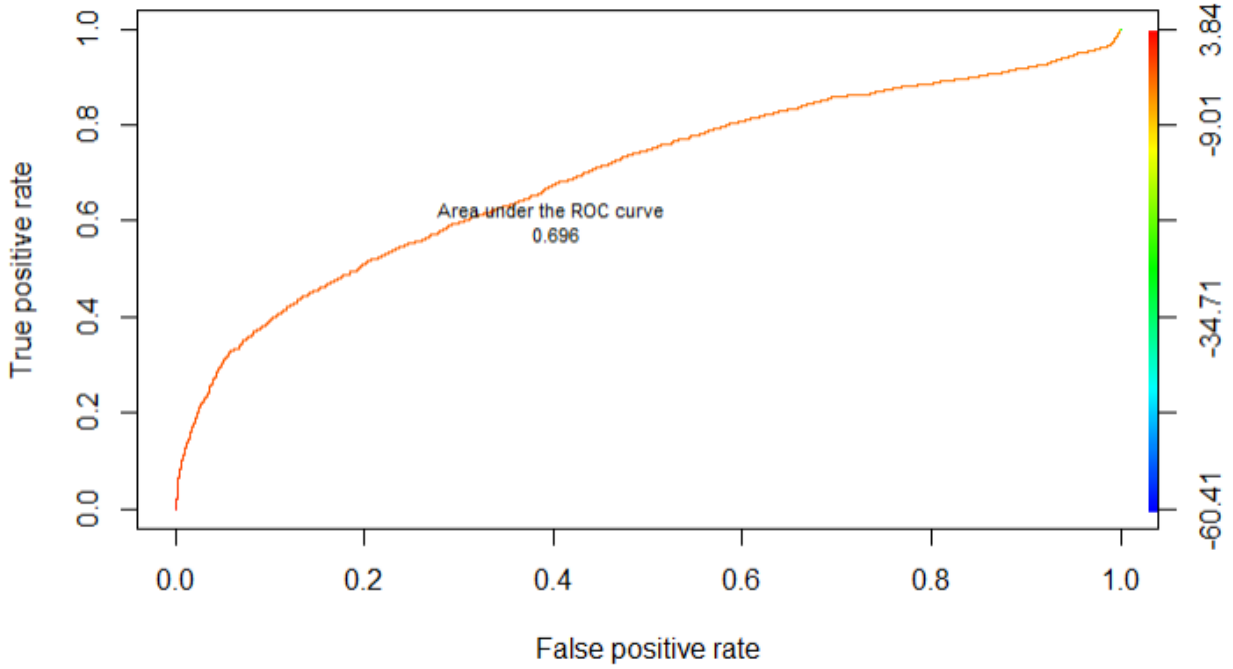
- **RANDOM FOREST:** algoritmo de tipo bagging (para un mismo predictor se establecen muestras de entrenamiento aleatorias, de la misma base de datos, varias veces). Con la realización de un bosque de árboles de decisión nuestro modelo llegará al que entiende como mejor resultado para estudiar en nuestra muestra de test (credit_test) si se impaga o no el crédito.

- **XGBOOST:** en este caso, se llevan a cabo distintos modelos estableciendo un peso diferente para cada uno, es parecido al random forest, aunque este algoritmo es de tipo boosting.

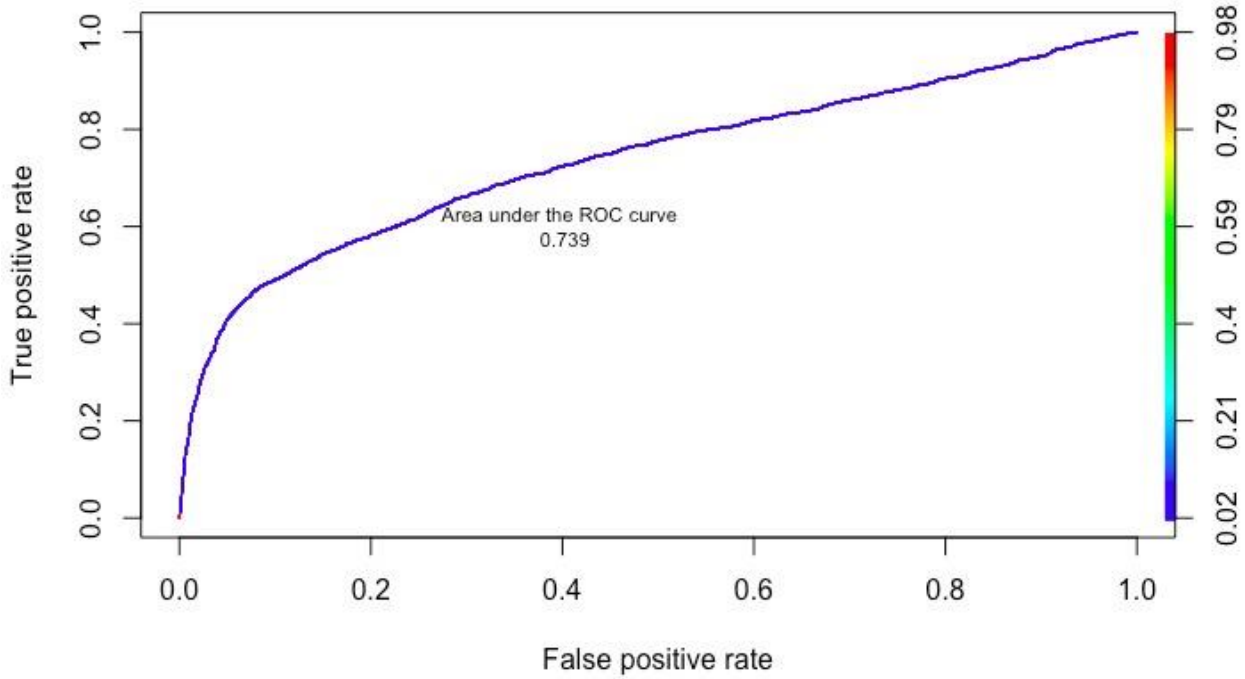
Por último y a modo de conclusión, estimaremos y visualizaremos las curvas ROC resultantes de cada uno de los modelos estimados, que posteriormente pintaremos todas juntas en la misma gráfica para una mejor comparación visual, y podremos observar qué modelo clasifica mejor una observación aleatoria, que será aquel con mayor área bajo su curva.

CURVAS ROC

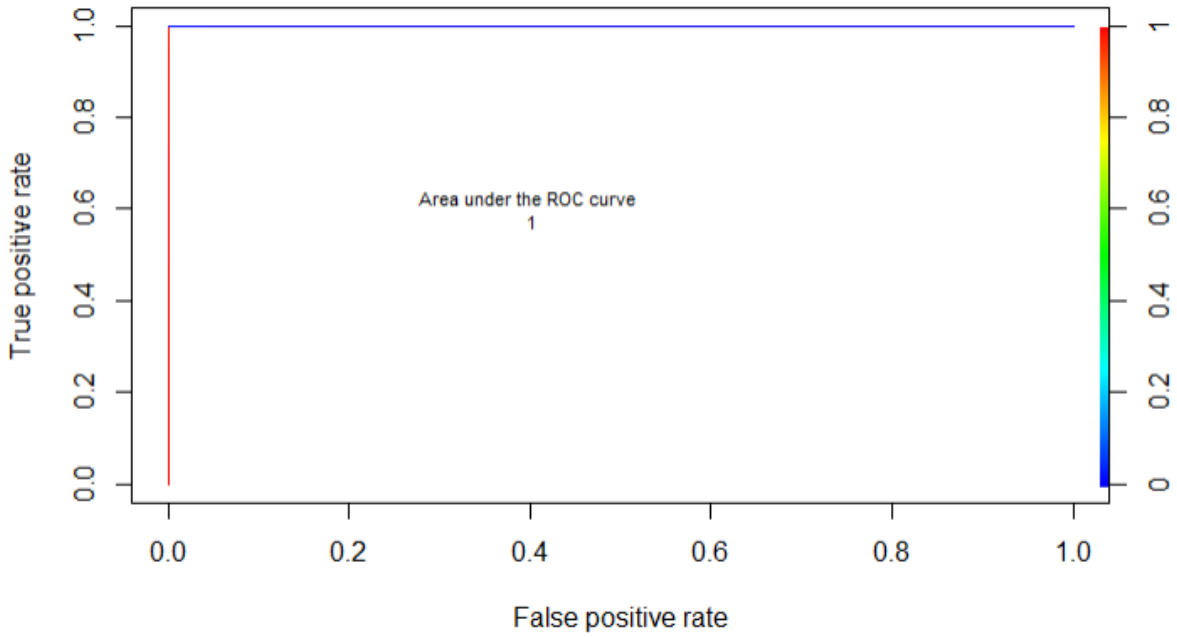
1- Regresión logística:



2- Support Vector Machine:

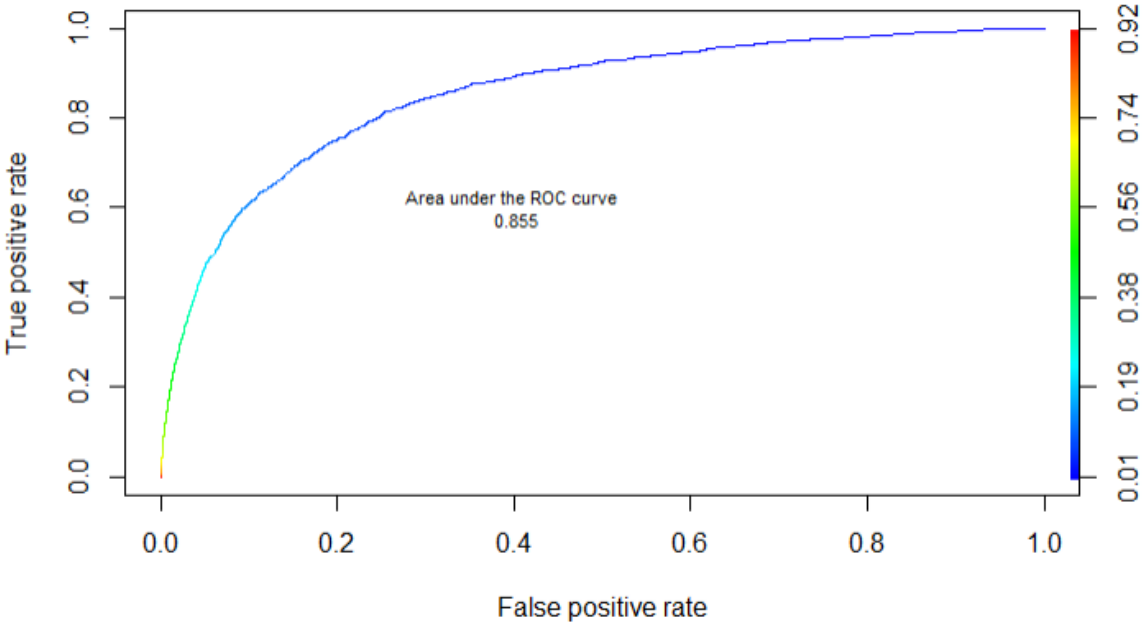


3- Random Forest: *

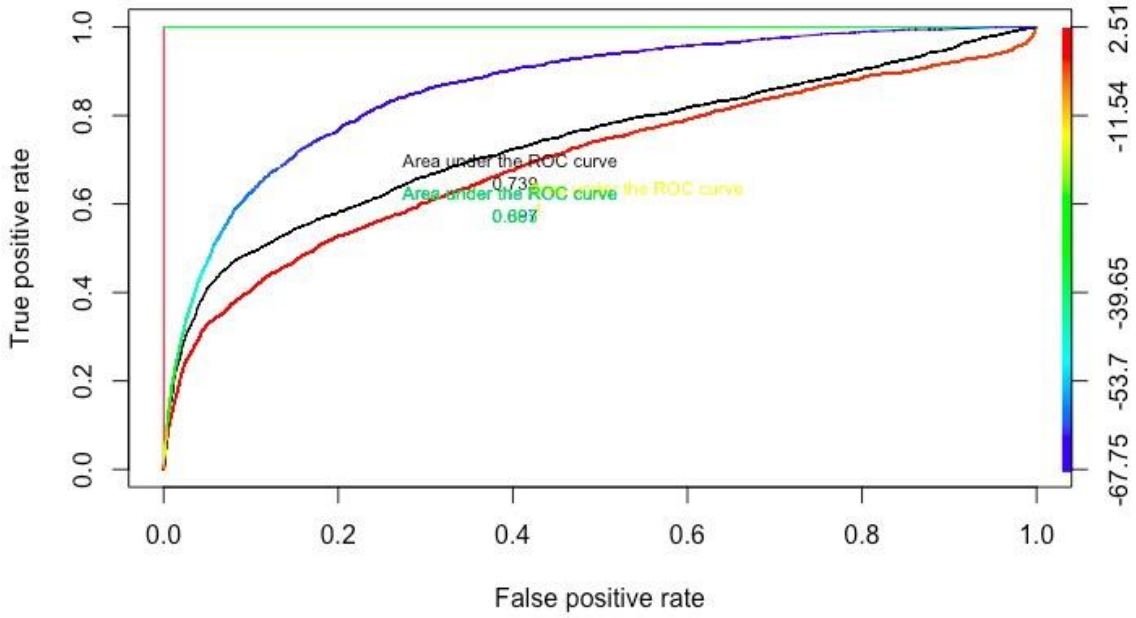


*Posteriormente se detectó un error de cálculo. Esto queda explicado en las conclusiones.

4- XGBoost:



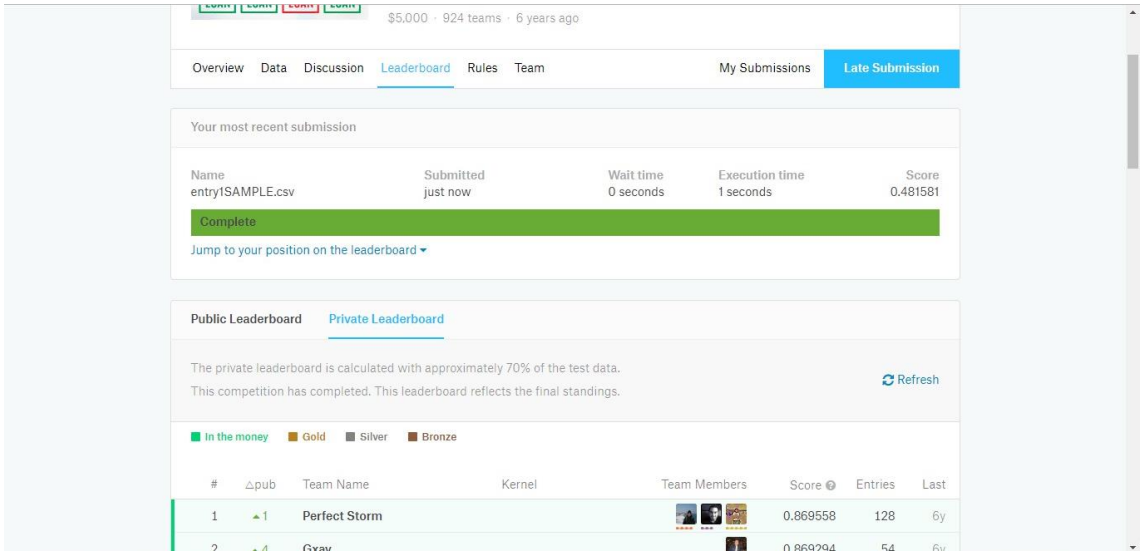
5- RL – SVM – RF – XGB:



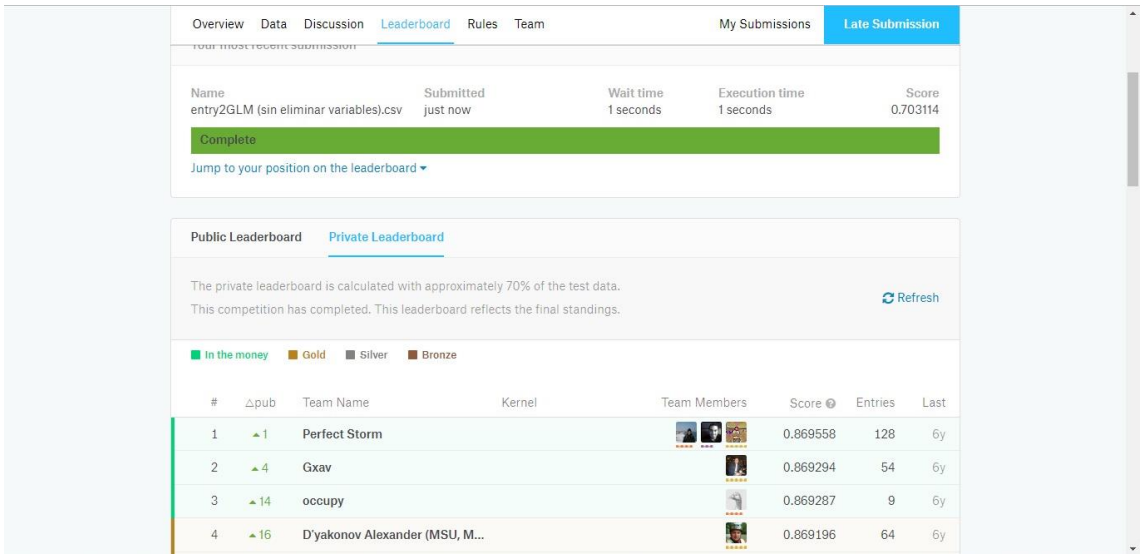
Conclusiones del conjunto de curvas ROC podemos encontrarlas en la tabla final del documento.

RESULTADOS KAGGLE

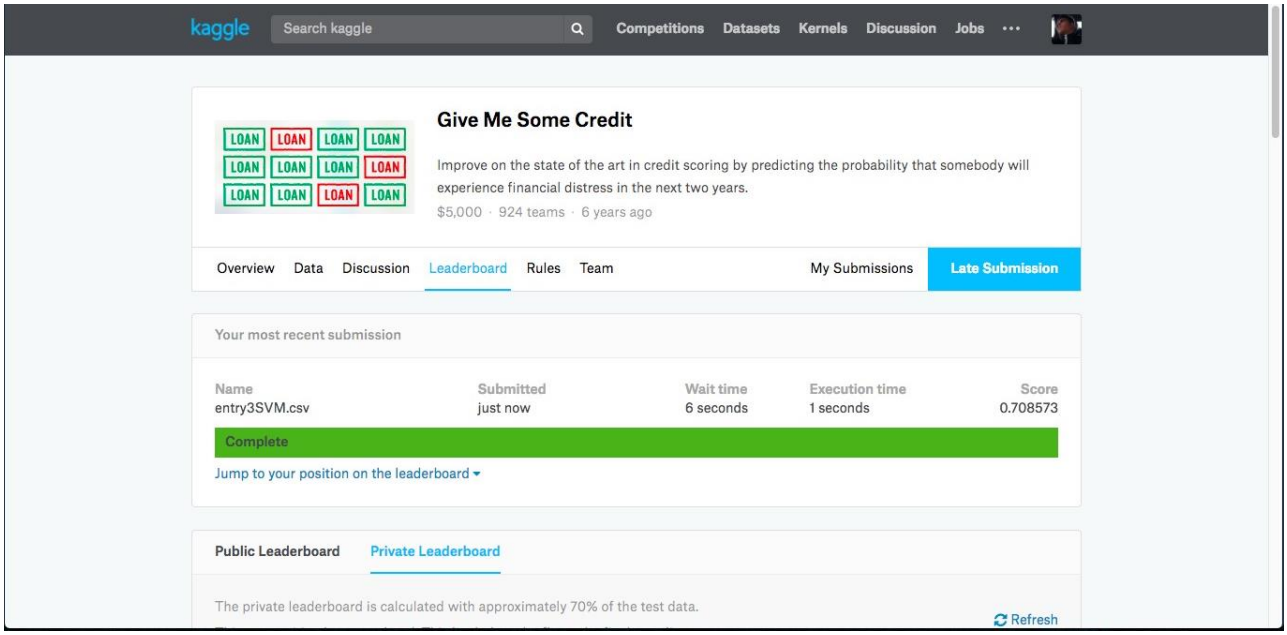
1- Muestra aleatoria:



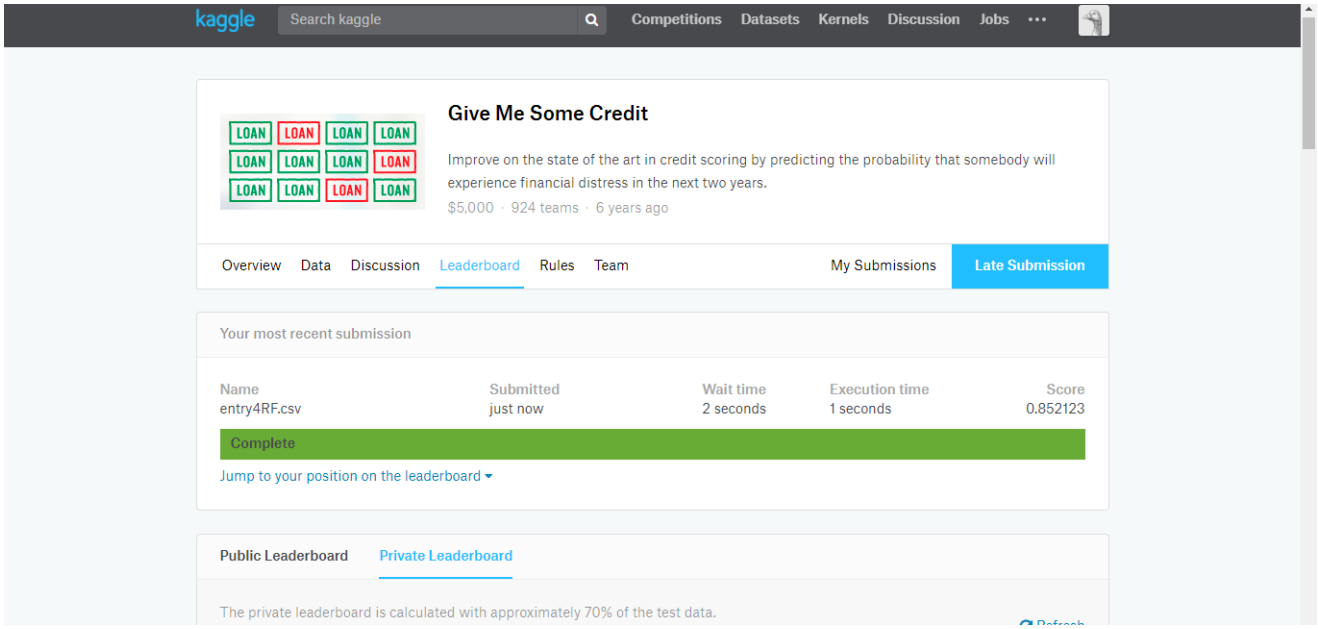
2- Regresión logística:



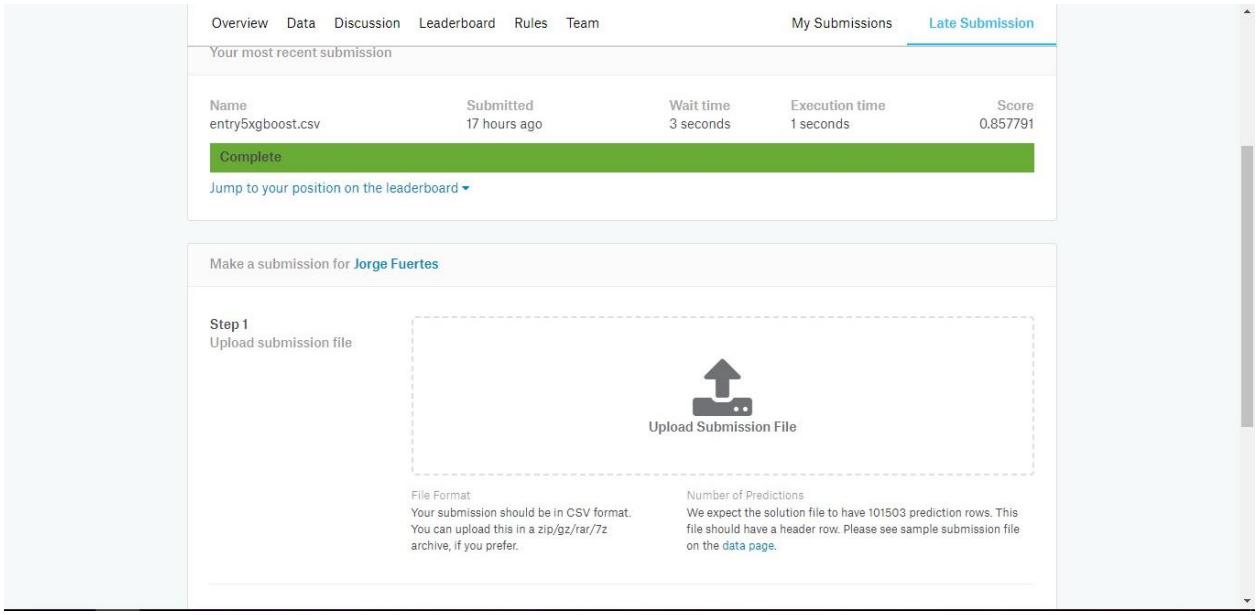
3- Support Vector Machine:



4- Random Forest:



5- XGBoost:


Overview
Data
Discussion
Leaderboard
Rules
Team
My Submissions
Late Submission
Your most recent submission
Name
entry5xgboost.csv
Submitted
17 hours ago
Wait time
3 seconds
Execution time
1 seconds
Score
0.857791
Complete
Jump to your position on the leaderboard
Make a submission for Jorge Fuertes
Step 1
Upload submission file
Upload Submission File
File Format
Your submission should be in CSV format. You can upload this in a zip/gz/rar/7z archive, if you prefer.
Number of Predictions
We expect the solution file to have 101503 prediction rows. This file should have a header row. Please see sample submission file on the data page.

CONCLUSIÓN:

Metodología llevada a cabo:

A lo largo de este trabajo se han llevado a cabo 4 modelos de Machine Learning distintos:

- XGBoost
- Random Forest
- SVM
- Regresión

En todos ellos la forma de proceder ha sido dividiendo el training set obtenido de Kaggle (el cual tiene los datos de la variable a predecir) en una muestra de training y otra de Test, de forma que pudiéramos comprobar de forma local la calidad de los modelos antes de llevar a cabo la predicción sobre el dataset de test de Kaggle, el cual no facilita los datos de la variable a predecir.

Una vez se realizaba el modelo sobre el conjunto de training, hemos procedido a realizar la predicción sobre el test de training, llevando a cabo para ello la curva ROC.

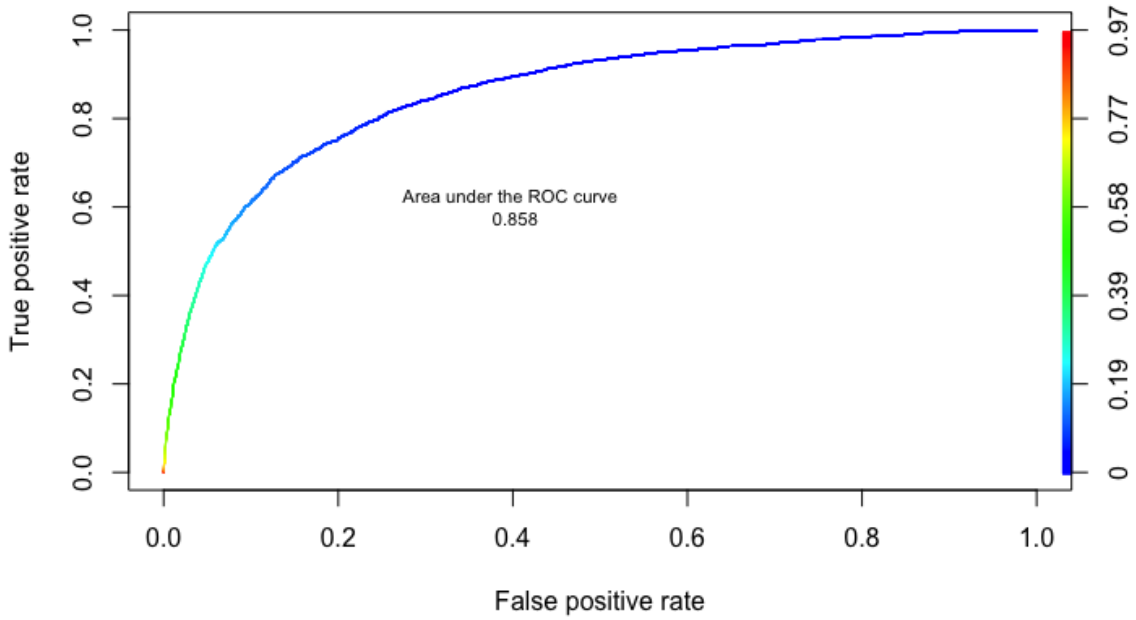
Finalmente, tras validar nuestro modelo con los datos de training, hemos procedido a ampliar los datos del modelo a todo el dataset de training obtenido de kaggle, para posteriormente realizar la predicción sobre el dataset de test del que desconocíamos los resultados. Finalmente hemos subido los resultados a Kaggle

- Análisis de los resultados:

Los resultados obtenidos varían con los distintos algoritmos, pero en general son bastante precisos, llegando a obtener un 85.78% de precisión en el modelo de XGBoost, siendo este el más preciso.

En cuanto al modelo de Random Forest, inicialmente obtuvimos una precisión del 100% sobre el conjunto de training, por lo que la curva ROC era la óptima. Lógicamente este resultado, aunque bueno, resulta muy improbable, por lo que investigamos las causas que podían estar ocasionando esta predicción perfecta. Finalmente descubrimos que en la función de Random Forest habíamos incluido el dataset con todas las columnas, por lo que la predicción había concluido que utilizando la variable a predecir en la propia predicción los resultados tendrían el mínimo error, lógicamente. Por lo tanto, para subsanar este error lo único que tuvimos que hacer fue volver a ejecutar el Random Forest excluyendo la columna a predecir del dataset.

Obteniendo finalmente la siguiente curva ROC:



En cuanto a la predicción del modelo subido a Kaggle, no hizo falta modificar nada pues en este caso si se había excluido dicha columna desde el principio y los resultados son correctos.

Resumen de los resultados:

Modelo	AIC (sobre training)	Puntuación Kaggle	Diferencia
Regresión logística	0,696	0,70311	-0,00711
Random Forest	0,858	0,852123	0,005877
SVM	0,739	0,708573	0,030427
SXBoost	**0,855**	**0,857791**	**-0,002791**

Por lo tanto, y a la vista de los resultados obtenidos, obtenemos que **el modelo XGBoost nos proporciona la predicción óptima con respecto al resto de modelos.**