Estudio sobre la empleabilidad de los estudiantes

Sergio Quijano Rey
sergioquijano@correo.ugr.es

17 de enero de 2023

Índice

Índice de figuras

Índice de cuadros

1. Abstract

En este trabajo, estudiaremos una base de datos consistente en métricas recogidas durante entrevistas de prueba, junto a si los candidatos son o no escogidos para el hipotético puesto de trabajo.

Con **dos objetivos** en mente:

- Construir un clasificador eficiente para predecir la empleabilidad
- Usar la base de datos para realizar un estudio sobre lo meritocrático del proceso. Dicho experimento consiste en entrenar dos modelos, uno usando las variables que consideramos meritocráticas, y otro usando las variables que consideramos no meritocráticas. Si el modelo que usa variables no meritocráticas funciona mejor, podemos pensar que entonces el proceso de selección se basa más en estos aspectos, que hemos considerado no meritocráticos

Para ello, realizaremos:

- Un estudio univariante de la base de datos, destacando el tratamiento de *outliers*, el estudio de la normalidad univariante, y un análisis descriptivo clásico
- Un estudio multivariante de la base de datos, destacando el estudio de las correlaciones entre variables, tratamiento multivariante de *outliers*, estudio de normalidad multivariante y reducción de la dimensionalidad con *PCA* y *FA*
- Ajuste de los modelos, destacando una pequeña exploración de hiperparámetros, entrenamiento y validación, comparando los resultados
- El experimento adicional, previamente mencionado

Al final, conseguimos obtener un modelo muy robusto a la hora de realizar predicciones, y todo el análisis realizado en el cuaderno, más el experimento adicional, confirman de forma contundente la falta de meritocracia en el proceso de selección.

2. Introducción

La base de datos usada se compone de métricas recogidas durante entrevistas de trabajo de pruebas a alumnos universitarios en las Filipinas. Además, en dicha base de datos, se recoge si el candidato es elegido o no para el hipotético puesto de trabajo. Dicha base de datos se encuentra en *Kaggle*, en [1]. Comentaremos más sobre la base de datos en *"3. Materiales y métodos"*.

Como ya se ha comentado, los objetivos del trabajo son dos:

1. Construir un modelo de clasificación robusto para predecir la empleabilidad de los candidatos
2. Estudiar la posible falta de meritocracia en el proceso de selección, reflejado en la base de datos. Esto mediante el análisis exploratorio de los datos, el análisis del comportamiento de los modelos obtenidos y finalmante mediante el experimento adicional

Para ello hemos realizado todas las tareas especificadas en el guión de prácticas, junto al experimento adicional. En este, realizamos las siguientes tareas:

1. Dividimos la base de datos en dos, una con variables meritocráticas y otra con las variables no meritocráticas [1]
2. Entrenamos dos modelos, uno en cada base de datos
3. Explicamos qué **implicaciones** tiene que el **modelo no meritocrático obtenga los mejores resultados**

De todas las tareas, las más relevantes a la hora de estudiar la meritocracia son:

- El análisis descriptivo de las variables (univariante y multivariante), donde vemos la poca relevancia del rendimiento académico
- Reducción de dimensionalidad. Obtenemos dos componentes principales y dos variables latentes. En ambos casos, obtenemos que la segunda componente o variable latente, se compone únicamente del rendimiento académico, con muy poca relevancia
- Análisis de la importancia de las variables a la hora de construir los modelos de discriminante lineal y *XGBOOST*. Con esto, podemos ver cómo ciertas variables no meritocráticas tienen demasiado peso respecto a las que consideramos meritocráticas
- El experimento deja claro que el proceso de selección no es meritocrático al obtener mejores resultados

Sobre el **estado del arte**, debemos comentar dos aspectos. El primero, referente a los **modelos de clasificación** que hemos usado. El análisis discriminante lineal y cuadrático da muy buenos resultados cuando los supuestos sobre los que se sustenta se cumplen. En esta

[1] Notar que esta elección es totalmente subjetiva. El lector puede opinar que cierta variable es meritocrática o no. No tenemos un criterio objetivo para realizar esta distinción. Sin embargo, el código está planteado para que se pueda modificar la subdivisión de la base de datos, y que el resto siga ejecutándose

base de datos no se cumplen, así que no podemos esperar resultados muy buenos, aunque como veremos, son decentes. El otro modelo que usamos, *XGBOOST*, es considerado uno de los mejores modelos para todo tipo de datos tabulares. El segundo aspecto que debemos comentar es sobre **los mejores resultados sobre esta base de datos**. En el repositorio de *Kaggle*, a día de 16 de Enero de 2023, tenemos 10 cuadernos de otras personas [2]. Los mejores resultados no pasan del 90 % de *accuracy* en *test*. Nuestros resultados son del 89 % de *accuracy* en *test*, sin haber hecho una búsqueda de hiperparámetros potente. Por tanto, podemos pensar que en ese aspecto hemos alcanzado prácticamente el estado del arte.

Por último, destacar que todo el desarrollo se puede ver en el repositorio de *Github*, que se encuentra en [3].

[2] El código de otras personas que comentamos se puede encontrar en [2]. De nuevo, notar que cuando escribimos esto, solo 10 personas han subido su código

3. Materiales y métodos

3.1. Materiales

En primer lugar, la **base de datos se puede obtener** del repositorio de *Kaggle* [1].

Como ya se ha comentado previamente, la base de datos se compone de registros con métricas de candidatos en entrevistas de trabajo de pruebas (para entrenar a los candidatos en estas entrevistas), junto con si el candidato es escogido o no para el hipotético puesto de trabajo.

El conjunto de datos se compone originalmente de 10 columnas (variables) y 2983 filas (registros). Las variables que contiene la base de datos son las siguientes:

1. Identificador numérico del estudiante. Lo llaman nombre del estudiante, pero en verdad se corresponde con una etiqueta numérica del tipo *"Student <id >"*
2. Apariencia general
3. Formas de hablar
4. Condición física
5. Agilidad mental
6. Confianza en si mismo
7. Habilidad para presentar ideas
8. Habilidades de comunicación
9. Rendimiento académico
10. Empleabilidad

Todas las variables, salvo el identificador y la empleabilidad, son variables discretas tomando valores del 1 al 5. La estructura del identificador ya la hemos comentado. Y la empleabilidad toma dos valores de tipo *string*: empleable o no empleable.

En [1], se comenta que la base de datos no ha sido tratada de ningún modo. Sin embargo, podemos inspeccionar la base de datos manualmente. El identificador llega hasta un valor de 3000, pero solo tenemos 2983 filas. Como vemos más adelante, no tenemos valores faltantes, así que asumimos con bastante seguridad que las 17 filas que faltan tenían valores faltantes, y se han borrado.

No usamos para nada el identificador del estudiante, así que de ahora en adelante, nos podemos olvidar de esa variable. Mostramos ahora algunos estadísticos básicos sobre el conjunto de datos sin tratar:

Variable	Mínimo	Máximo	1º Cuartil	Media	Mediana	3º Cuartil	Desv. Típica
Agilidad mental	2.0	5.0	3.0	3.963	4.0	5.0	0.7814509
Rendimiento académico	3.0	5.0	4.0	4.611	5.0	5.0	0.6895586
Formas de hablar	2.0	5.0	3.0	3.885	4.0	4.0	0.7527855
Condición física	2.0	5.0	3.0	3.972	4.0	5.0	0.7466378
Apariencia	2.0	5.0	4.0	4.247	4.0	5.0	0.6824864
Confianza	2.0	5.0	3.0	3.911	4.0	5.0	0.8074833
Habilidad para presentar ideas	2.0	5.0	3.0	3.814	4.0	4.0	0.7339429
Habilidades comunicativas	2.0	5.0	3.0	3.525	3.0	4.0	0.7371541

Cuadro 1: Resumen del conjunto de datos original

Con esto vemos que aunque las métricas van, en un diseño inicial, del valor 1 al 5, muchas de ellas tienen un rango más acotado (por debajo), puesto que no hubo participantes que obtuviesen los valores mínimos.

Al estar trabajando con variables discretas, el siguiente gráfico es muy útil a la hora de visualizar la distribución de los datos:

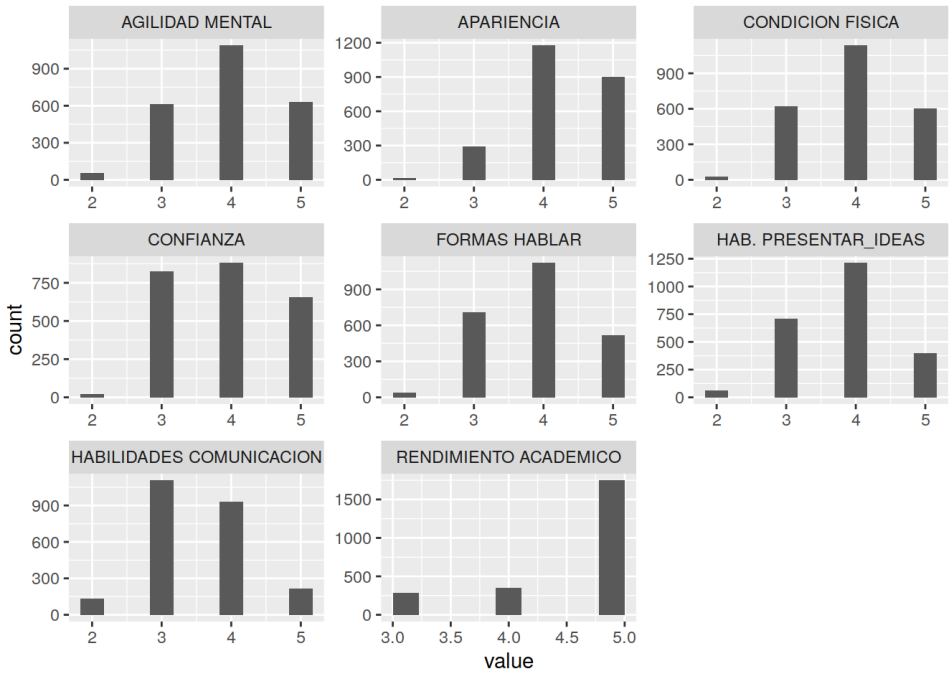


Figura 1: Histograma de las variables de entrada

Mostramos gráficamente la distribución de la variable de salida:

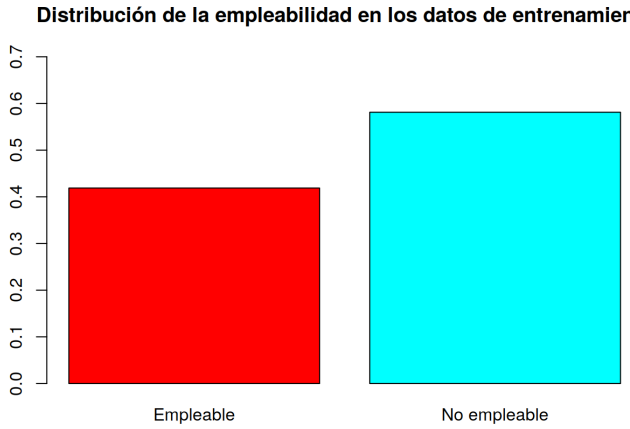


Figura 2: Distribución de la variable de salida, con la que podemos estudiar el balanceo de las clases

Hay cierto desbalanceo hacia la no empleabilidad, aproximadamente un $40-60\,\%$. En general, este desbalanceo no es demasiado grave. Además, considerando el desbalanceo esperable dado el contexto en el que estamos (un porcentaje mayoritario no debería ser aceptado para un trabajo), consideramos que no necesitamos aplicar alguna técnica para tratar este desbalanceo (i.e. podría aplicarse *SMOTE*).

3.2. Métodos estadísticos

Para el **análisis univariante**, usamos los siguientes métodos:

- Separación en entrenamiento y *test* para una futura validación cruzada ($80\,\%$ - $20\,\%$)
- Recodificación del *string* de la variable de salida, en valores $0, 1$ y con un *factor* de R
- Gráficos de cajas
- Análisis univariante de *outliers*, usando rangos intercuartílicos
- Test de normalidad univariante: test de *Shapiro-Wilk*

Para el **análisis multivariante**, usamos:

- Matriz de correlaciones, test de correlación: test de esfericidad de *Bartlett*
- Análisis multivariante de *outliers* usando la distancia de *Mahalanobis*
- Estandarización de las variables
- Componentes Principales. Para elegir el número de componentes principales:
  - Regla de *Abdi*
  - Mínimo de Varianza explicada

    - Método del codo
    - Análisis paralelo
- Análisis Factorial. Para elegir el número de variables latentes:
    - Método del codo
    - Análisis paralelo
- Test de normalidad multivariante: test de *Mardia* y test de *Henze-Zirkler*
- Test de la homogeneidad de la varianza: test *Box M*

Para la **exploración de los hiperparámetros**, necesarios para la futura clasificación, usamos [3]:

- Los tres modelos clasificadores: discriminante lineal, discriminante cuadrático y *XGBOOST*
- *k-Fold Cross Validation* como método estadístico para realizar la búsqueda de hiperparámetros. Usamos 10 *folds*.

Para la **clasificación**, aunque escogemos la mejor combinación de modelo-conjunto de datos, entrenamos sobre los tres modelos, en cada uno usando el mejor conjunto de datos para ese modelo.

Además, estudiamos las importancias de las variables a la hora de construir el modelo (en *LDA* y *XGBOOST*).

Para la **validación de los modelos**, usamos:

- *Accuracy* sobre entrenamiento y *test*
- Matriz de confusión

Y para finalizar, en el **experimento adicional**, usamos:

- Separación de la base de datos en función de si tenemos variables meritocráticas o no
- Ajuste de hiperparámetros de *XGBOOST* usando *k-Fold Cross Validation* y *Grid Search*
- Validación de los resultados obtenidos con el *accuracy* y matriz de confusión

[3] Tenemos tres conjuntos de datos con los que podemos construir un clasificador. También estamos trabajando con tres modelos distintos. Así que realizamos esta exploración de hiperparámetros para escoger entre las nueve combinaciones posibles.

4. Resultados

4.1. Análisis Univariante y Multivariante

En *"1. Histograma de las variables de entrada"* ya hemos visto los histogramas de las variables. Podemos complementar esto con los diagramas de cajas:

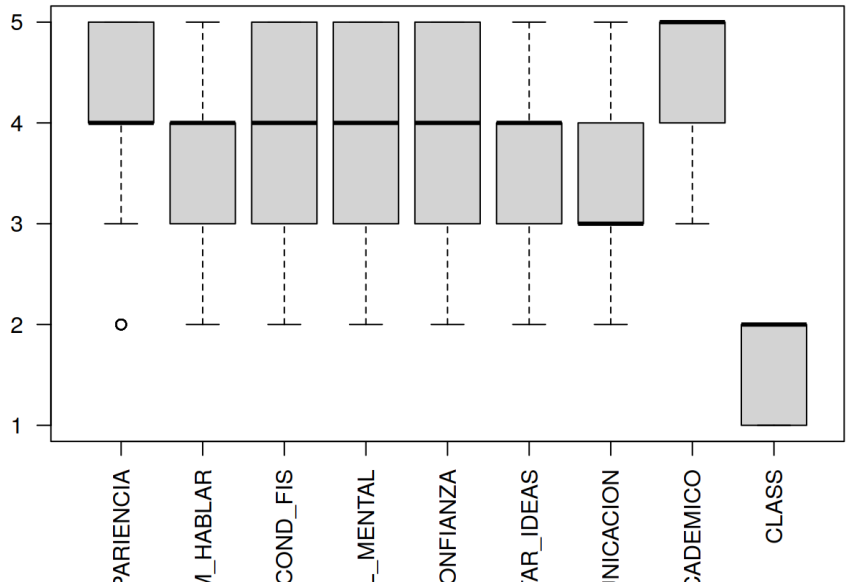


Figura 3: Gráficos de cajas de las variables de entrada

A la hora de borrar *outliers*, de forma univariante, solo la variable *Apariencia* los presenta. Concretamente, presenta 14 registros, lo que supone un 0.587 % del total de los registros, por que lo borramos dichos registros. Aplicando la misma transformación de datos, para no caer en *data snooping*, borramos 2 filas en test, lo que supone un 0.335 %.

A la hora de borrar *outliers*, de forma multivariante, borramos 13 filas del conjunto de entrenamiento, lo que supone un 0.084 %. Para no caer en *data snooping*, no aplicamos este borrado usando la distancia *Mahalanobis* sobre el conjunto de *test*.

En *"2. Distribución de la variable de salida, con la que podemos estudiar el balanceo de las clases"* ya hemos visto el balanceo de las clases, con aproximadamente un 60 % de no empleables, 40 % de empleables.

El estudio de la normalidad univariante, apoyado en los gráficos *qqplot* y el contraste de hipótesis nos muestra que **no tenemos normalidad univariante**, de forma muy contundente. Los gráficos *qqplot* obtenidos son los siguientes:

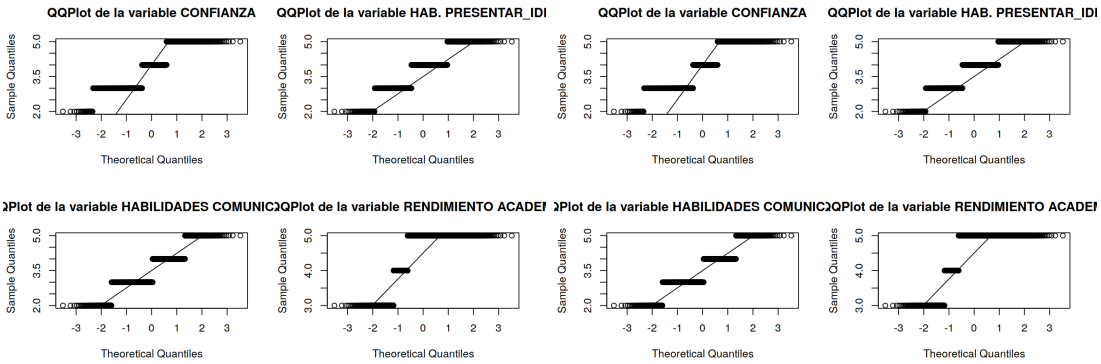


Figura 4: Gráficos *qqplot* de las variables de entrada

La visualización de la matriz de correlaciones, junto con el *test de esfericidad de Bartlett* nos dejan claro que las variables están correlacionadas. La matriz de correlaciones se muestra como la siguiente visualización:

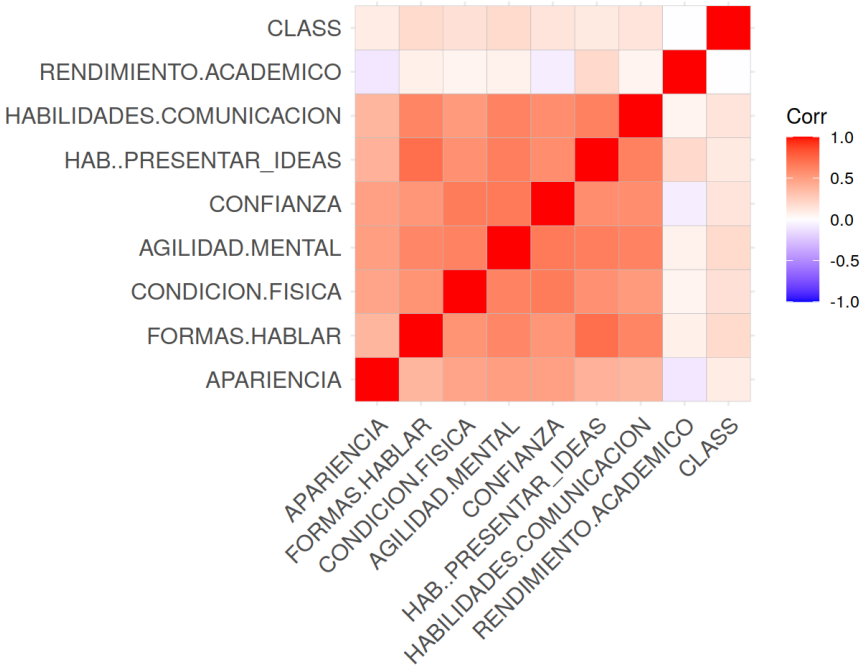


Figura 5: Matriz de correlaciones

Podemos visualizar, en otro formato, los pares de variables que más correladas están entre sí:

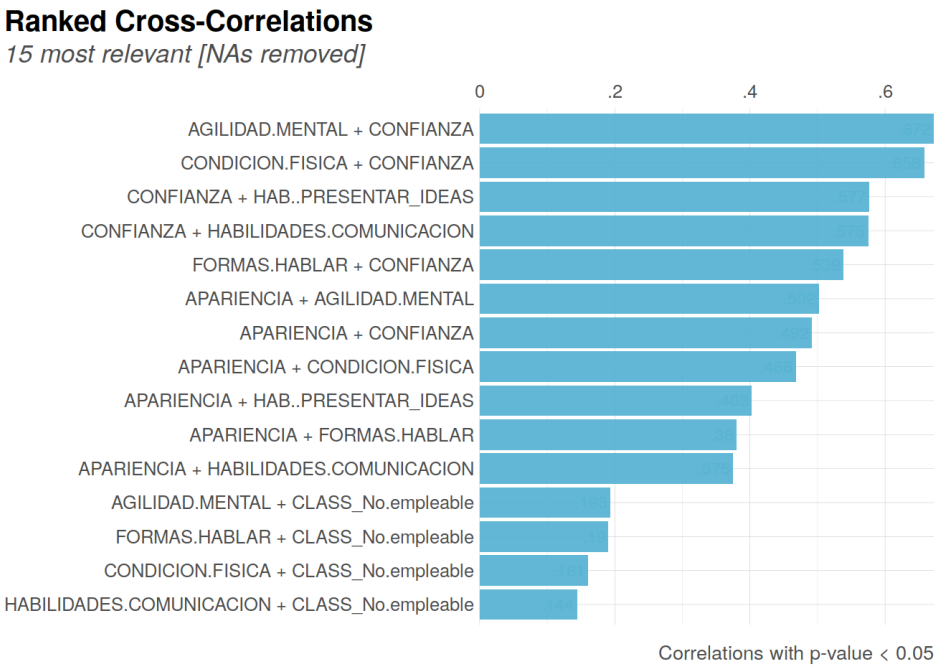
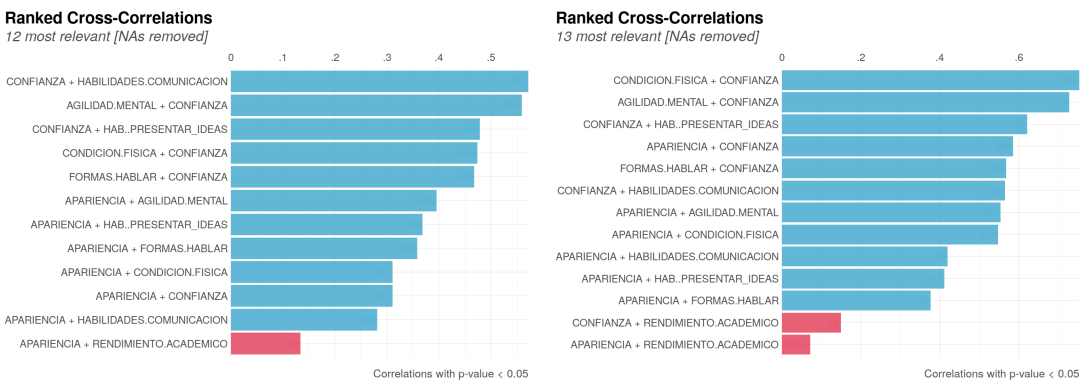


Figura 6: Pares de variables más correladas entre sí

Y también, podemos ver esto mismo en dos subconjuntos de datos, el de registros asociados a personas empleables y el de no empleables:



(a) Candidatos empleables

(b) Candidatos no empleables

Figura 7: Pares de variables más correladas entre sí, según la empleabilidad

Los contrastes de hipótesis, que ya hemos comentado, nos indican que **NO tenemos normalidad multivariante**, como era de esperar al fallar la normalidad univariante.

Otro contraste de hipótesis nos indica que **NO tenemos homogeneidad de la varianza**. Aunque al fallar la normalidad multivariante, este resultado no es fiable. No es del todo relevante que el *test* no sea fiable. Puesto que estamos comprobando los supuestos para los modelos discriminante, y la falta de normalidad ya hace que, independientemente de la homogeneidad de la varianza, no vayan a funcionar de forma óptima.

Por tanto, los dos **supuestos sobre los que se sustentan el discriminante lineal y cuadrático fallan**.

4.2. Reducción de la dimensionalidad

Esta parte es muy relevante en nuestro estudio. Porque producimos dos nuevos conjuntos de datos, que proporcionarán uno de los mejores resultados a la hora de clasificar, y porque además serán muy relevantes a la hora de estudiar la posible falta de meritocracia.

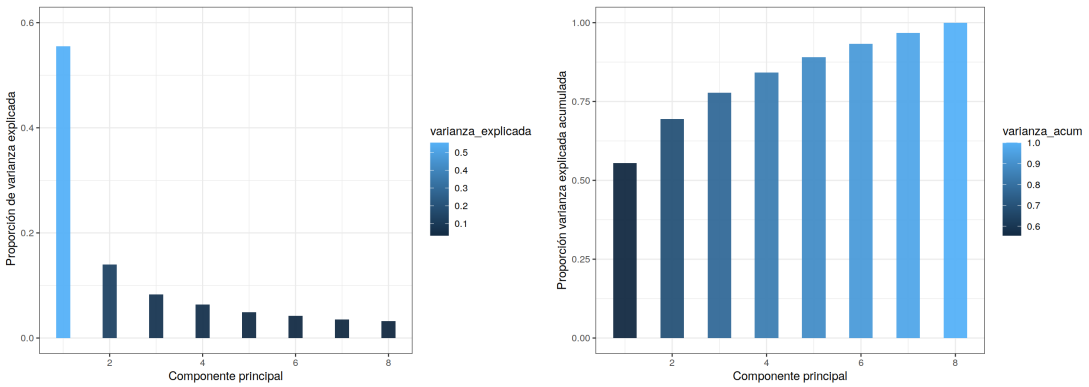
Además, mantenemos a partir de ahora tres conjuntos de datos:

1. El conjunto de datos original, sobre el que realizamos el tratamiento ya mencionado
2. El conjunto de datos original al que aplicamos *PCA*
3. El conjunto de datos original al que aplicamos *FA*

Esto porque a la hora de clasificar, tenemos que elegir todavía cuál es la mejor combinación de modelo-conjunto de datos. Y no todos los modelos necesariamente tengan los mejores resultados sobre el mismo conjunto de datos.

4.2.1. Componentes Principales

Veamos la varianza explicada, y la varianza explicada acumulada, por las componentes principales:



(a) Proporción de la varianza explicada por las componentes principales

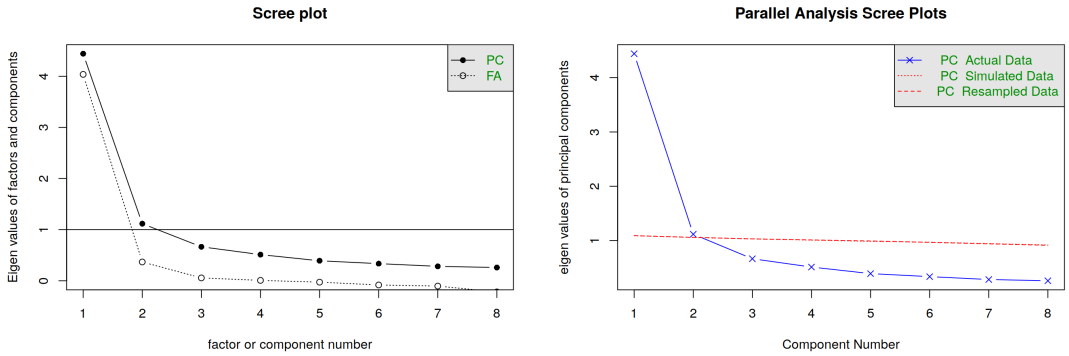
(b) Proporción de la varianza explicada acumulada por las componentes principales

Figura 8: Información sobre la proporción de la varianza explicada por cada una de las componentes principales

Los resultados de los métodos para elegir el número de componentes principales son:

- Regla de *Abdi*: dos componentes
- Mínimo de Varianza explicada: cuatro componentes, buscando explicar al menos el 80 % de la varianza
- Método del codo: dos componentes
- Análisis paralelo: dos componentes

El método del codo y análisis paralelo se fundamentan en los siguientes gráficos:



(a) Gráfica del método del codo

(b) Gráfica para el método paralelo

Figura 9: Gráficas sobre las que se fundamentan el método del codo y método del análisis paralelo

Por tanto, elegimos usar **dos componentes principales**

Tras realizar la transformación de los datos, podemos mostrar el siguiente gráfico:

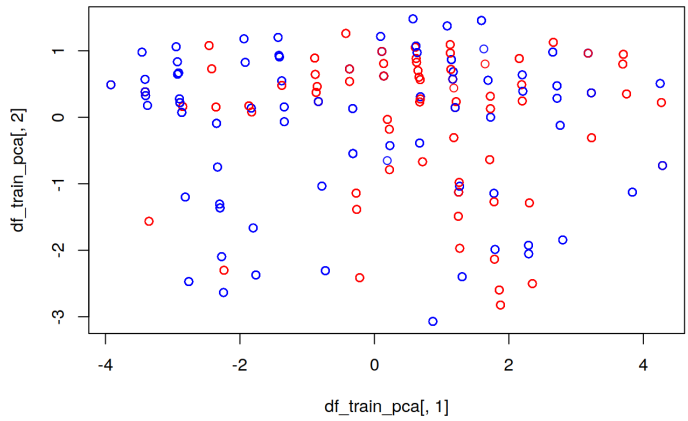


Figura 10: Dos primeras componentes principales, coloreadas según la empleabilidad

Además, podemos visualizar la relevancia de las variables de entrada a la hora de generar las dos componentes principales:

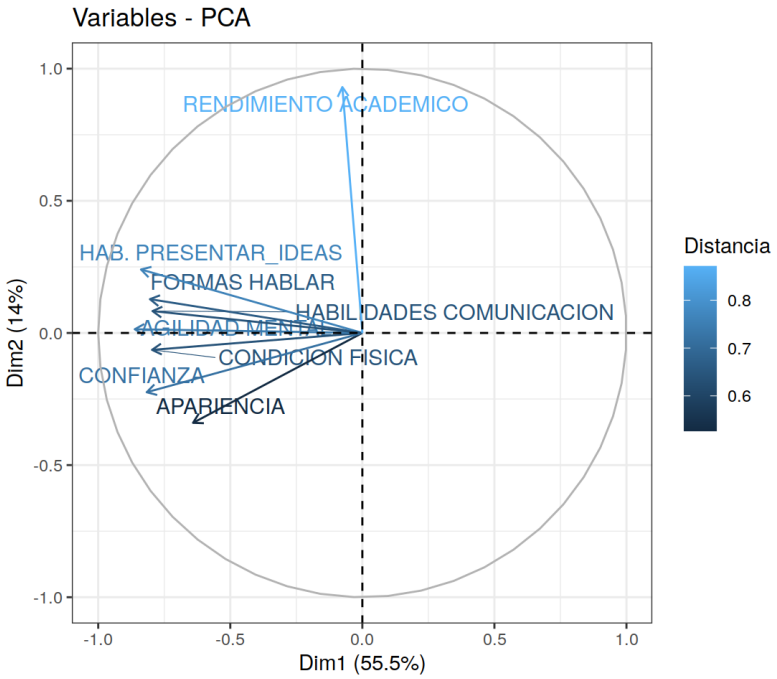


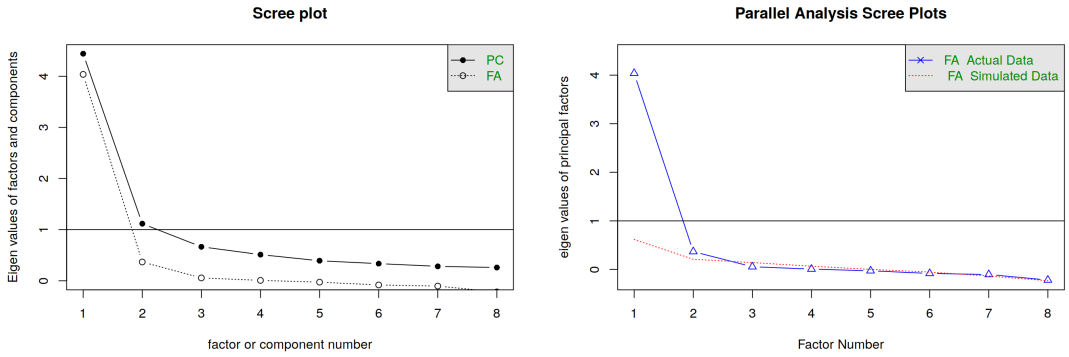
Figura 11: Importancia de las variables de entrada a la hora de calcular las dos primeras componentes principales

4.2.2. Análisis Factorial

Los resultados de los métodos para elegir el número de variables latentes son:

- Método del codo: un factor latente
- Análisis paralelo: dos factores latentes

Esto se fundamenta en los siguientes gráficos:

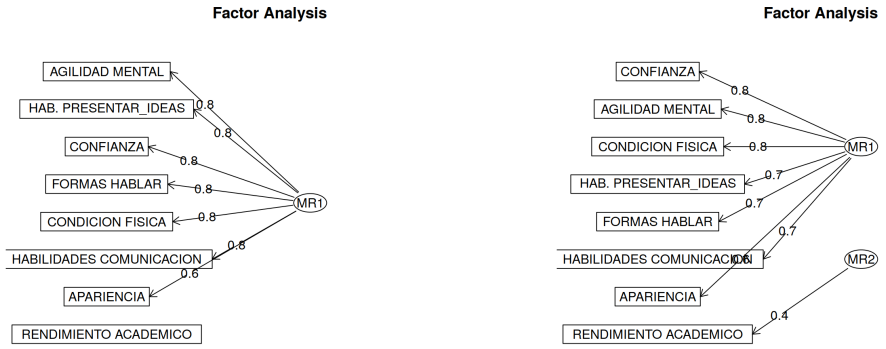


(a) Gráfica del método del codo

(b) Gráfica para el método paralelo

Figura 12: Gráficas sobre las que se fundamentan el método del codo y método del análisis paralelo

Con un contraste de hipótesis, vemos que es suficiente con un factor latente, y por tanto, también con dos factores latentes. Podemos visualizar cómo se construyen las variables latentes, tanto en el caso de una o dos variables latentes:



(a) Combinación de las variables de entrada, para un solo factor latente

(b) Combinación de las variables de entrada, para dos factores latentes

Figura 13: Gráfica que muestra cómo se combinan las variables para calcular los factores latentes, para un solo factor latente y para dos factores latentes

Por los motivos que comentamos en *"5. Discusión"*, al final decidimos quedarnos con dos variables latentes, aunque sea suficiente con una variable latente.

4.3. Exploración de hiperparámetros

XGBOOST tiene hiperparámetros que podemos optimizar usando *k-Fold Cross Validation*. Sin embargo, por la potencia del modelo, basta con que usemos unos parámetros que consideramos razonables, y los modificamos ligeramente para obtener el resultado final. Usamos los siguientes hiperparámetros en este modelo:

- `max_depth` = 10
- `eta` = 0.3
- `nrounds` = 100

Los resultados se resumen en la siguiente tabla:

Modelo	Dataset	Accuracy
Discriminante Lineal	Original	0.5915605
Discriminante Lineal	PCA	0.5334985
Discriminante Lineal	FA	0.5284333
Discriminante Cuadrático	Original	0.6891568
Discriminante Cuadrático	PCA	0.5615132
Discriminante Cuadrático	FA	0.587775
XGBOOST	Original	0.9014924
XGBOOST	**PCA**	**0.9046639**
XGBOOST	FA	0.9036134

Cuadro 2: Resultados de *k-Fold Cross Validation*

Vemos que los mejores resultados se obtienen con *XGBOOST* sobre el conjunto de *PCA*. El discriminante cuadrático y el discriminante lineal obtienen sus mejores resultados sobre el conjunto de datos original.

4.4. Clasificación

Aunque **ya hemos decidido que el mejor modelo es *XGBOOST*** sobre los datos a los que aplicamos *PCA*, entrenamos y validamos los tres modelos, cada uno con el conjunto de datos sobre el que obtiene mejores resultados. Con esto, **podemos realizar un estudio sobre el comportamiento de los tres modelos**.

Los coeficientes (en valor absoluto) del discriminante lineal, que discutiremos en *"5. Discusión"*, se visualizan en la siguiente figura:

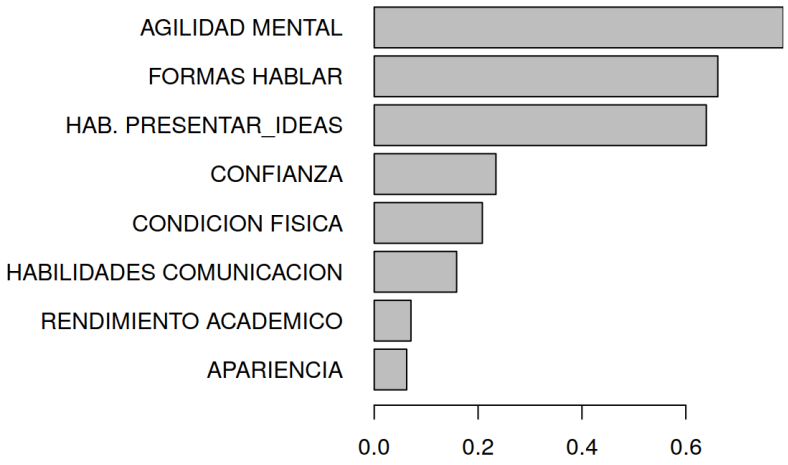
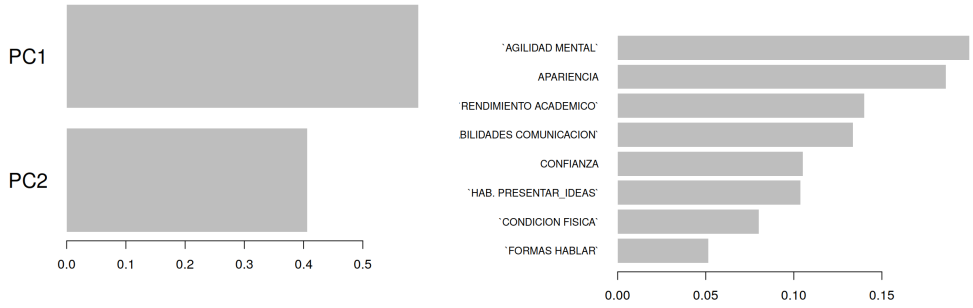


Figura 14: Visualización de los valores absolutos de los coeficientes del modelo discriminante lineal

El paquete de *XGBOOST* nos otorga una función para mostrar la importancia de cada variable. Mostramos dicha importancia, tanto para el modelo sobre *PCA* (que es el que escogemos por lo visto en *"2. Resultados de k-Fold Cross Validation"*) como para el modelo sobre el conjunto de datos original (que solo usamos para visualizar dicha relevancia):



(a) Modelo sobre el conjunto de datos al que aplicamos *PCA*

(b) Modelo sobre el conjunto de datos original

Figura 15: Importancia de las variables a la hora de construir el modelo *XGBOOST*

4.5. Validación

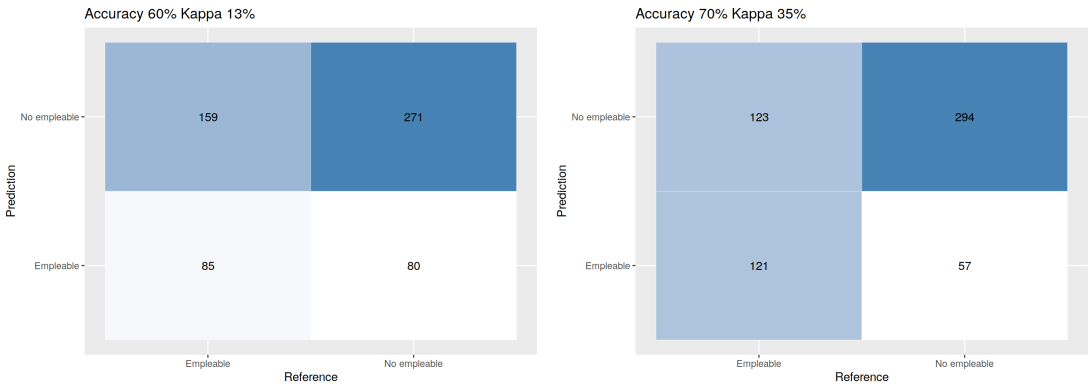
Empezamos mostrando el *accuracy* de los modelos, tanto en entrenamiento como en *test* [4]:

Modelo	*Train acc*	*Test acc*
Discriminante lineal	0.591178965224767	0.598319327731092
Discriminante cuadrático	0.706106870229008	0.697478991596639
XGBOOST	0.913910093299406	0.890756302521008

Cuadro 3: *Accuracy* de los tres modelos entrenados

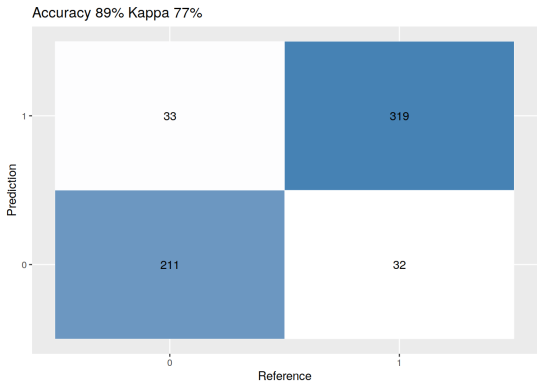
Mostramos ahora las matrices de confusión, de los tres modelos, de forma visual, sobre el conjunto de *test*:

[4] De esta forma, podemos diagnosticar ciertos problemas, principalmente, el *overfitting*



(a) *LDA*

(b) *QDA*



(c) *XGBOOST*

Figura 16: Matrices de confusión, sobre el conjunto de *test*, de los tres modelos. Notar que los ejes de *XGBOOST* están cambiados, pero la representación sigue siendo correcta fijándonos en la leyenda

4.6. Experimento adicional

Como hemos visto que el mejor modelo es *XGBOOST*, y como nos da algunas facilidades para interpretar los modelos obtenidos, usaremos este modelo. Además, usaremos el conjunto de datos original para simplificar el desarrollo del experimento. Esto porque, para aplicar *PCA* o *FA*, habría que aplicarlo a los dos conjuntos por separado, y ver si se obtienen mejores resultados que sin aplicar la transformación, usando *k-fold Cross Validation*.

Realizamos la separación del conjunto de datos original en dos, uno conteniendo las variables que consideramos meritocráticas, y otro con las variables que consideramos no meritocráticas.

Como variables meritocráticas consideramos:

- Rendimiento académico
- Agilidad mental
- Habilidad para presentar ideas
- Habilidades de comunicación

Como variables no meritocráticas consideramos:

- Condición físicas
- Apariencia
- Formas de hablar
- Confianza

Realizamos un ajuste de hiperparámetros de los dos modelos *XGBOOST*. Esto lo realizamos usando *k-Fold Cross Validation* con *Grid Search*. Los mejores parámetros para cada modelo son:

- Modelo meritocrático: `nrounds` = 480, `max_depth` = 3, `eta` = 0.3
- Modelo no meritocrático: `nrounds` = 120, `max_depth` = 5, `eta` = 0.1
- Para el resto de parámetros, usamos el valor por defecto

Tras entrenar los modelos, los resultados obtenidos son:

Conjunto de datos	Train Acc	Test Acc
Meritocrático	0.733248515691264	0.694117647058824
No meritocrático	0.719677692960136	**0.715966386554622**

Cuadro 4: Resultados del experimento

5. Discusión

En el cuaderno que se entrega junto a esta memoria, se discuten prácticamente todos los resultado que obtenemos más o menos en profundidad. Por tanto, aquí solo comentamos los resultados más relevantes. Si el lector tiene curiosidad por profundizar en alguno de los resultados presentados previamente, en dicho cuaderno seguramente encuentre dichos resultados discutidos.

El **primero de nuestros objetivos era construir un clasificador robusto**. Podemos considerar que hemos logrado este objetivo. En 4.3 elegimos *XGBOOST* sobre el conjunto de datos al que aplicamos *PCA* como el modelo más robusto. En 4.5 vemos que:

- Efectivamente, *XGBOOST* es el modelo que mejor funciona para esta base de datos
- El modelo generaliza bien, puesto que en el conjunto de *test* se comporta prácticamente igual de bien

Era de esperar que *XGBOOST* fuera el mejor modelo. Los supuestos sobre los que se fundamentan los dos modelos de discrimante fallan, y *XGBOOST* es de los modelos más potentes para datos tabulares.

Nuestro **segundo objetivo era comprobar si el proceso de selección no es meritocrático**. Esta pregunta es más complicada de responder, y nos tenemos que fundamentar en muchos más recursos.

En primer lugar, vemos en 5 que todas las variables de entrada, salvo el **rendimiento académico**, están bastante correlacionadas entre sí. El rendimiento académico apenas está correlacionada con ninguna otra variable. Esto nos hace sospechar en un primer momento que **dicha variable no va a ser relevante** a la hora de determinar la empleabilidad. De hecho, es la variable menos correlacionada con la variable de salida. Es más, en 6 no aparece. En dicho gráfico, y en 7, podemos ver que la apariencia y la confianza son de los conceptos más relevantes, considerando que ambos son (según nuestro criterio) no meritocráticos.

Los dos métodos de reducción de dimensionalidad dejan claro que el rendimiento es muy poco relevante. En ambos casos, el primer elemento (componente principal o variable latente) se compone de una combinación de todas las variables menos el rendimiento. El segundo elemento deja al rendimiento en solitario, teniendo mucha menos importancia dicho segundo elemento. Esto se fundamenta en 11 y 13

Con 14 obtenemos un modelo más justo de lo que esperábamos: la agilidad mental, formas de hablar y habilidad para presentar ideas son las tres variables más importantes, con una gran diferencia respecto a la cuarta (la confianza). Tenemos dos variables meritocráticas (agilidad mental y habilidad para presentar ideas) como las más relevantes. Aunque hay que comentar que este modelo no obtiene resultados satisfactorios. En 15 las dos variables más importantes son la agilidad mental y la apariencia. Nos sorprende que **la segunda variable más importante a la hora de determinar la empleabilidad**, según este modelo, sea la **apariencia**, que quizás pueda considerarse la **variable menos meritocrática de todas**, solo por debajo de la condición física. En tercer lugar tenemos el rendimiento académico. Nos sorprende porque ya hemos visto la poca relevancia que parece tener. Sin embargo, seguimos pensando que tiene poca relevancia. Al trabajar con árboles de decisión, puede ocurrir que usemos esta variable como desempate cuando la agilidad mental y apariencia no sea suficientes para determinar la empleabilidad.

El experimento deja claro que el proceso no es meritocrático, por lo que vemos en 4. Aunque la diferencia en *test* sea pequeña, estamos superando el rendimiento usando **únicamente variables no meritocráticas**. Con solo conocer las formas de hablar, condición física, apariencia y confianza obtenemos mejores resultados que conociendo el rendimiento académico, las habilidades de comunicación y de presentar ideas y la agilidad mental, que deberían ser mucho más útiles.

6. Conclusión

Consideramos que **hemos logrado alcanzar los dos objetivos planteados**: construir un clasificador robusto para la base de datos y estudiar la posible falta de meritocracia en el proceso de selección de los candidatos.

Del trabajo realizado, **destacamos**:

- El análisis exploratorio de la base de datos, que ya nos da bastante información sobre la falta de meritocracia que reflejan los datos
- El uso de varios clasificadores, que escogemos usando *k-fold Cross Validation*, y que una vez entrenados son interpretables, proporcionando información muy valiosa para nuestro estudio de la meritocracia. Además, este proceso genera un clasificador muy robusto
- El experimento adicional que deja más clara la falta de meritocracia

Como **puntos a mejorar en un futuro trabajo**, consideramos principalmente:

- En el experimento adicional, escogemos las variables de forma manual y completamente subjetiva. En un futuro trabajo, se puede usar una metodología *wrapper feature selector* para seleccionar automáticamente las variables, como se describe en [4] o en [5]
- Los dos modelos de discriminante no han funcionado bien por la falta de los supuestos. La base de datos, con variables discretas con rangos muy pequeños, puede no darnos tanto juego como nos gustaría. Así que un buen futuro trabajo sería repetir la metodología de este trabajo sobre un mejor conjunto de datos

7. Referencias

[1] A. Hamoutni, "Students' employability dataset - philippines." https://www.kaggle.com/datasets/anashamoutni/students-employability-dataset, 2022.

[2] A. Hamoutni, "Students' employability dataset - philippines, notebooks." https://www.kaggle.com/datasets/anashamoutni/students-employability-dataset/code, 2022.

[3] S. Q. Rey, "Sergioquijanorey/estadisticamultivariantepracticafinal: Repositorio para la práctica final de la asignatura "estadística multivariante", de la universidad de granada." https://github.com/SergioQuijanoRey/EstadisticaMultivariantePracticaFinal, 2023.

[4] W. Saelens, "Wrapping in r :: dynverse." https://dynverse.org/developers/creating-ti-method/create_ti_method_r/, 2023.

[5] K. Szutu, "Feature selection using wrapper methods in r - analytics vidhya - medium." https://medium.com/analytics-vidhya/feature-selection-in-r-9bfae551d22b, Apr 2020.