UNIVERSIDAD
COMPLUTENSE
MADRID

Grado en Matemáticas y Estadística

Machine Learning

Informe final maqueta de startup

Encuentra tu Carrera

Marina Pescador

Miguel Zabaleta

Contenido

1. Presentación del problema

Escoger qué carrera estudiar no es tarea fácil. Aunque siempre hay quien sabe cuál es su vocación desde pequeño, la realidad es que hay un gran número de estudiantes de secundaria y bachillerato que no saben cómo tomar esta decisión tan crucial para su futuro.
Con nuestro proyecto queremos crear un sistema para encontrar tu carrera ideal de forma interactiva. Existen muchos cuestionarios para recomendar carreras, pero, con solo una serie de preguntas básicas es difícil encontrar una verdadera vocación. Nosotros proponemos un servicio más personalizado que no se quede en esa propuesta original, sino que te guíe en los primeros estadios de descubrimiento de esa carrera y sea capaz de rectificar su propuesta inicial con el feedback que le irá proporcionando.

Para ello proponemos basarnos en un test inicial que evaluará tu personalidad y te recomendará diferentes recursos (vídeos, artículos, etc) introductorios de tu carrera elegida y te irá pidiendo feedback. Según los resultados que vayas proporcionando seguirá dando recursos de esa carrera (si tu respuesta hacia la información proporcionada es positiva) o cambiará de carrera (si es negativa), este cambio podría ser más o menos drástico dependiendo del nivel de descontento con la carrera en cuestión y los resultados del test original.
Para para conseguir un mayor nivel de *engagement* se implementa un sistema de puntos que a medida que vas consumiendo los recursos proporcionados te haría avanzar hasta confirmar tu carrera ideal. Con esto deseamos **gamificar** el programa y que así los usuarios sientan mayor deseo por continuar hasta llegar al "objetivo".

Además, dicho test inicial utilizará técnicas de **machine learning** para crear los resultados en función de una base de datos que consistirá en respuestas de estudiantes de grado reales, lo cual esperamos que nos dé información real sobre el tipo de aptitudes perfectas para cada carrera ya que muchos de los tests no parecen estar basados en datos reales de gente que haya estudiado dicha carrera sino en los estereotipos que se asocian con esta.

2. Datos a emplear y análisis exploratorio

Una vez decidida la estructura que iba a tener nuestro proyecto, el primer paso a dar fue conseguir los datos necesarios para crear el modelo predictivo de este. Para ello creamos un cuestionario Google que luego rellenaron estudiantes universitarios de grado.

2.1. Creación del cuestionario

Nuestro primer paso fue decidir qué datos íbamos a utilizar tanto para el cuestionario final como aquellos que solo necesitaremos para el procesamiento de los datos a recopilar. Tras una búsqueda de diferentes modelos de personalidad decidimos utilizar el modelo Big Five también conocido como OCEAN. Es un modelo común y aceptado en psicología que analiza los rasgos de la personalidad dividiéndola en los cinco grandes rasgos de personalidad:

- Apertura a la experiencia (inventivo/curioso vs. consistente/cauteloso) O
- Escrupulosidad (eficiente/organizado vs. extravagante/descuidado) C

- Extroversión (sociable/enérgico vs. solitario/reservado) E
- Amabilidad (amigable/compasivo vs. desafiante/insensible) A
- Neuroticismo (susceptible/nervioso vs. resistente/seguro) N

Estos factores fueron encontrados experimentalmente en una investigación sobre las descripciones de personalidad que unas personas hacían de otras (Goldberg, 1993). Usando análisis factorial, los investigadores pudieron observar las respuestas de las personas a cientos de elementos de personalidad y hacerse la pregunta ¿Qué es lo mejor para resumir a un individuo? Esto se ha hecho con muchas muestras de todo el mundo y el resultado general es que, si bien parece haber ilimitadas variables de personalidad, estas cinco destacan a la hora de explicar muchas incógnitas de una persona a preguntas sobre su personalidad. Los cinco grandes no están asociados con ninguna prueba en particular, se han desarrollado una variedad de medidas para medirlos. Nosotros seleccionamos el cuestionario de la página ya mencionada como base y tradujimos las preguntas intentando que conserven al máximo el estilo para evitar malentendidos. Estas fueron nuestras preguntas Big Five finales:

Soy el alma de la fiesta	Arruino las cosas	Comprendo nuevos conceptos con rapidez
Me preocupan poco los demás	No suelo estar triste	No me gusta llamar la atención
Siempre estoy preparado	No me interesan los conceptos abstractos	Saco tiempo para los demás
Me estreso fácilmente	Soy yo quien empieza las conversaciones	Eludo mis tareas
Tengo un vocabulario rico	No me interesan los problemas de los demás	Tengo cambios de ánimo frecuentes
No soy muy hablador	Hago mis tareas en el momento	Uso palabras complejas
Me interesan las personas	Soy interrumpida con facilidad	No me importa ser el centro de atención
Voy olvidando mis pertenencias a mi alrededor	Tengo buenas ideas	Siento las emociones de los demás
Suelo estar relajado	No tengo mucho que opinar	Sigo un horario
Me resulta difícil entender ideas abstractas	Tengo un corazón blando	Me enfado con facilidad
Me siento cómodo rodeado de personas	Se me suele olvidar guardar las cosas en su sitio	Paso tiempo reflexionando en las cosas
Hablo mal de la gente	Me enfado con facilidad	Soy callado alrededor de desconocidos
Soy detallista	No tengo mucha imaginación	La gente se siente tranquila a mi alrededor
Me preocupo por las cosas	Hablo con mucha gente distinta en fiestas	Sobresalgo en mis estudios
Tengo una gran imaginación	No me interesan mucho los demás	Me siento decaído a menudo
Me mantengo en segundo plano	Me gusta el orden	Estoy repleto de ideas
Simpatizo con los sentimientos de los demás	Cambio de estado de ánimo con facilidad	

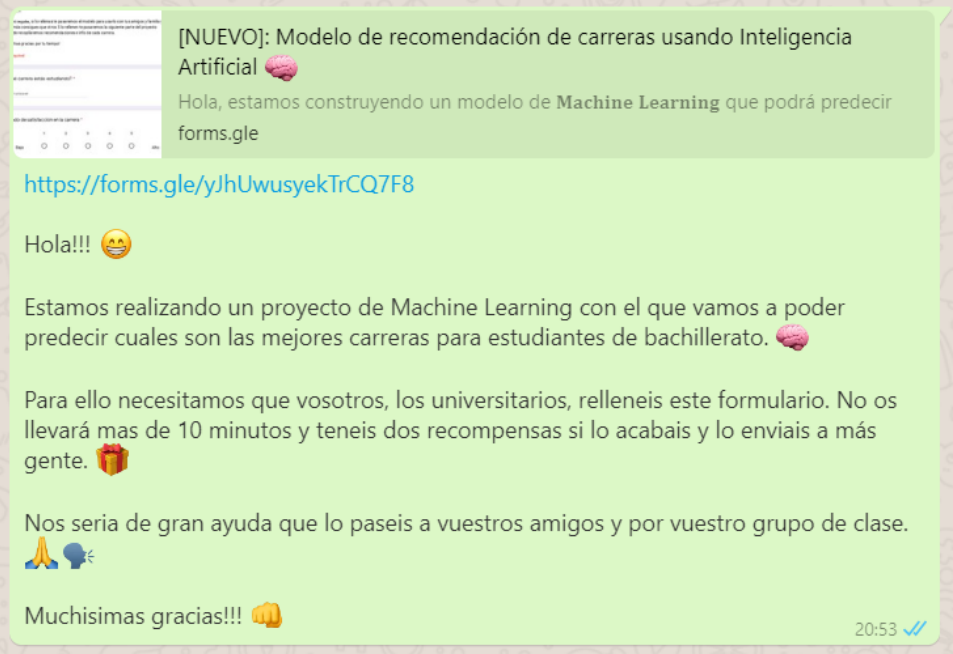
A estas preguntas decidimos añadir algunas más clásicas para la elección de carrera, en las que se podían elegir de cero a tres opciones:
¿Qué carrera escogerías si no estuvieses estudiando tu carrera actual?
¿Qué asignatura detestaban en el colegio/instituto?

Para el cuestionario del modelo predictivo también añadimos las siguientes preguntas:
¿Qué carrera estás estudiando?, claramente necesaria ya que es la variable a predecir.
Grado de satisfacción en la carrera, utilizada para poder cribar las respuestas descartando aquellas con un grado de satisfacción bajo.
"Qué carrera escogerías si no estuvieses estudiando tu carrera actual", esta no fue utilizada en el modelo final ya que con el número de datos con los que trabajamos es limitado y solo estábamos teniendo en cuenta un pequeño abanico de carreras; por lo que muchas de las respuestas obtenidas no entraban dentro de este. Pero a la hora de implementar el proyecto y conseguir una base de datos significativamente mayor creemos que sí que serían datos interesantes y no muy difíciles de añadir al modelo predictivo.

2.2. Recopilación de los datos

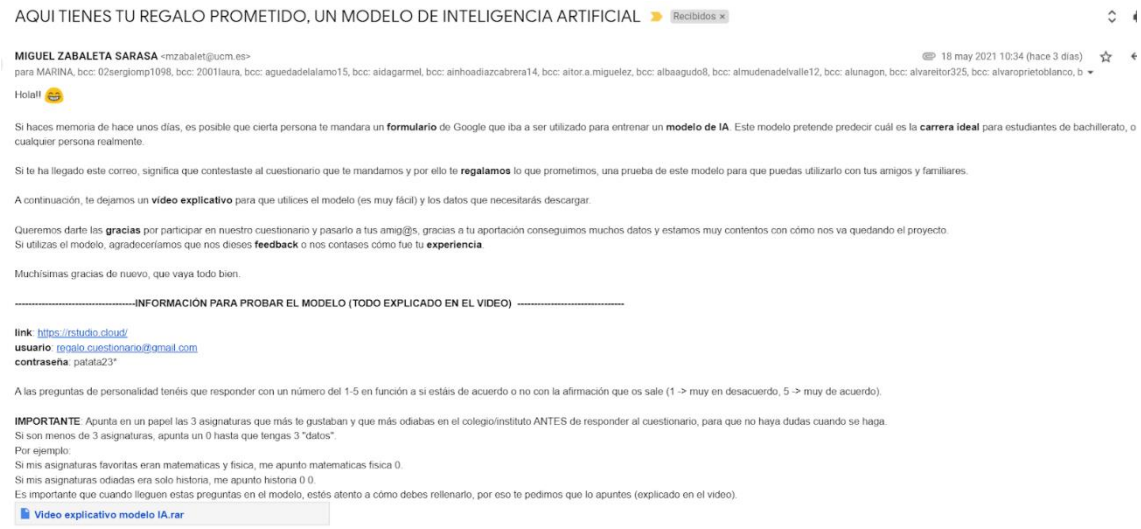
Elegimos hacer el cuestionario en Google forms ya que era una opción simple y gratuita y lo único que tuvimos que añadir a las preguntas ya mencionadas fue la opción (no obligatoria) de poner tu correo electrónico para recibir la recompensa, de la que hablaremos más adelante. Este es el cuestionario final.
Nos pareció importante que tanto la introducción al cuestionario como el mensaje de agradecimiento y recompensa fuesen amenos de leer y que incitasen a realizar el cuestionario.

Una vez hecho el cuestionario nuestro siguiente problema fue cómo conseguir animar a la gente a contestar. Decidimos compartirlo por grupos de clase y mandarlo a conocidos universitarios pidiendo si lo podían compartir por los suyos. Para animar a toda la gente a contestar escribimos el siguiente mensaje:



Intentamos que el mensaje fuese corto, fácil y agradable de leer. Esto lo conseguimos siendo **concisos** en el mensaje, **separando en párrafos** según las ideas e incluyendo **emoticonos** al final de cada párrafo. Además, nos pareció buena idea ofrecer cierta **recompensa** para fomentar más aún la involucración de las personas.

En cuanto a la recompensa, este es el correo que enviamos a más de 120 personas y esta es la rutina correspondiente:



AQUI TIENES TU REGALO PROMETIDO, UN MODELO DE INTELIGENCIA ARTIFICIAL Recibidos ×

MIGUEL ZABALETA SARASA <mzabalet@ucm.es> 18 may 2021 10:34 (hace 3 días)

para MARINA, bcc: 02sergiomp1098, bcc: 2001laura, bcc: aguedadelalamo15, bcc: aidagarmel, bcc: ainhoadiazcabrera14, bcc: aitor.a.miguelez, bcc: albaagudo8, bcc: almudenadelvalle12, bcc: alunagon, bcc: alvareitor325, bcc: alvaroprietoblanco, b

Hola!!

Si haces memoria de hace unos días, es posible que cierta persona te mandara un **formulario** de Google que iba a ser utilizado para entrenar un **modelo de IA**. Este modelo pretende predecir cuál es la **carrera ideal** para estudiantes de bachillerato, o cualquier persona realmente.

Si te ha llegado este correo, significa que contestaste al cuestionario que te mandamos y por ello te **regalamos** lo que prometimos, una prueba de este modelo para que puedas utilizarlo con tus amigos y familiares.

A continuación, te dejamos un **vídeo explicativo** para que utilices el modelo (es muy fácil) y los datos que necesitarás descargar.

Queremos darte las **gracias** por participar en nuestro cuestionario y pasarlo a tus amig@s, gracias a tu aportación conseguimos muchos datos y estamos muy contentos con cómo nos va quedando el proyecto.
Si utilizas el modelo, agradeceríamos que nos dieses **feedback** o nos contases cómo fue tu **experiencia**.

Muchísimas gracias de nuevo, que vaya todo bien.

----------**INFORMACIÓN PARA PROBAR EL MODELO (TODO EXPLICADO EN EL VIDEO)** ----------

link: https://rstudio.cloud/
usuario: regalo.cuestionario@gmail.com
contraseña: patata23*

A las preguntas de personalidad tenéis que responder con un número del 1-5 en función a si estáis de acuerdo o no con la afirmación que os sale (1 -> muy en desacuerdo, 5 -> muy de acuerdo).

IMPORTANTE: Apunta en un papel las 3 asignaturas que más te gustaban y que más odiabas en el colegio/instituto ANTES de responder al cuestionario, para que no haya dudas cuando se haga.
Si son menos de 3 asignaturas, apunta un 0 hasta que tengas 3 "datos".
Por ejemplo:
Si mis asignaturas favoritas eran matematicas y fisica, me apunto matematicas fisica 0.
Si mis asignaturas odiadas era solo historia, me apunto historia 0 0.
Es importante que cuando lleguen estas preguntas en el modelo, estés atento a cómo debes rellenarlo, por eso te pedimos que lo apuntes (explicado en el video).

Video explicativo modelo IA.rar

Nuestra idea fue diseñar una rutina de forma que los usuarios simplemente tuviesen que ejecutar el código entero y automáticamente fuesen apareciendo las preguntas del cuestionario y, una vez respondidas, seguidamente se mostrase el resultado de la predicción. De esta forma diseñamos el uso del modelo de la forma más sencilla posible para gente que no estuviese familiarizada con el machine learning o la programación.

Aunque habíamos planeado simular parte de los datos, ya que no esperábamos obtener los suficientes, el cuestionario funcionó muy bien en nuestra opinión, consiguiendo un **total de 212 respuestas**. De estas pudimos escoger, una vez hecho el procesamiento inicial (explicado en el siguiente apartado), un total de **132 respuestas** repartidas entre **4 carreras** con las que hacer el modelo predictivo. Por tanto, consideramos que nuestras pequeñas estrategias para tratar de atraer al máximo número de personas funcionaron muy bien y estamos muy satisfechos con los resultados obtenidos, sobre todo teniendo en cuenta que la información que obtuvimos es muy valiosa.

Cuando ya teníamos la mayor parte del proyecto redactado, conseguimos que una "*influencer*" con más de 600k seguidores en Instagram publicase en su historia nuestro cuestionario. En las 24 horas que dura la historia, conseguimos pasar de 212 respuestas a **667**. Por desgracia, era demasiado tarde para usar estos datos y volver a hacer todo el proyecto. Sin embargo, la conclusión que sacamos de esto es que a través de las redes sociales se podría publicitar la encuesta y obtener un gran número de respuestas. Por ejemplo, podríamos ponernos en contacto con distintos *influencers* que puedan tener interés en el ámbito universitario/académico, presentarles nuestro proyecto, y que publiquen la encuesta en sus cuentas.

A raíz de este gran número de nuevas respuestas, decidimos volver a enviar el correo con la recompensa, con el fin de obtener feedback y valoraciones positivas. Nos

llegaron **42** solicitudes para usar el modelo de prueba (de unas 200 enviadas) y tuvimos feedback positivo sobre la idea del proyecto.

2.3. Procesamiento de los datos

Tras observar los datos obtenidos, consideramos quedarnos con estas agrupaciones de carreras ya que son de las que más datos conseguimos.

Matemáticas: 75
Arquitectura: 15
Farmacia y Químicas: 17
Medicina y Ciencias de la Salud:25

Filtramos los datos, eliminando datos duplicados y registros en los que el grado de satisfacción con su carrera era de 1 o 2. Eliminamos la variable '*Marca temporal*' para que no nos estorbe, ya que no es de utilidad.

Las variables "*Asignaturas favoritas en el colegio*" y "*Asignaturas que detestabas en el colegio*" necesitaron ciertas transformaciones para poder ser usadas. Estas variables inicialmente tomaban los valores de las asignaturas que habíamos dado como opciones. Indicamos en el cuestionario que rellenasen 1, 2 o ninguna, pero había bastantes observaciones con más de 2 asignaturas. Nuestro planteamiento fue conseguir tener cada registro de estas variables como un **texto de 3 palabras** (por ejemplo: "Matematicas Biologia Ingles", o "Arte Fisica 0) para poder vectorizarlos como en el modelo *bag of words*.
Para ello, comenzamos sustituyendo los ' ; ' por ' , ' porque en un primer momento pensamos en leer los campos como un csv y así tener las variables deseadas directamente. Luego nos dimos cuenta que no podíamos obtener las variables directamente desde el csv, pero seguimos sustituyendo los ' ; ' por ' , ' de todas formas. Después, añadimos en cada registro el texto ' ,0 ' hasta que el texto tuviese 2 comas para rellenar los registros con menos de 3 asignaturas con un 0. Pasamos el *dataframe* resultante a un archivo excel para trabajar con este archivo en R.

Siguiente paso: **vectorizar el texto** de los campos.
Primero, sustituimos las comas por espacios para que se pudiesen leer correctamente. Segundo, creamos los objetos *document term matrix* correspondientes a cada variable. A continuación, quitamos las variables sobrantes, añadimos las columnas de los objetos *document term matrix* y renombramos las nuevas columnas.
Eliminamos las variables originales sobre las asignaturas, 2 variables inútiles para el estudio (datos de contacto de los participantes) y la variable "*qué carrera harías si no fuese la que estudias ahora*" ya que tampoco es de utilidad para este modelo de entrenamiento, en el que solo estamos considerando cuatro carreras.
En segundo lugar, realizamos dos bucles en los que añadimos cada columna de los *dtm* al *dataframe* y cambiamos su nombre.
Por último, comprobamos el número de valores *missing* (5 observaciones) e imputamos estos valores con el valor 3, ya que se corresponde con una respuesta neutra en el cuestionario.
Código

3. Soluciones y maqueta propuesta

3.1 Modelos

En cuanto a los modelos que desarrollamos, aunque al fin y al cabo estemos tratando con una maqueta de lo que sería el proyecto a gran escala, decidimos implementar los algoritmos **KNN**, **Árbol de decisión**, **Random Forest** y una **Red Neuronal** sencilla.

Consideramos que los dos primeros modelos son importantes ya que aportan un grado de **interpretabilidad** muy interesante a la hora de explicar el modelo (tanto a los estudiantes como a las posibles instituciones que adquieran el servicio).
Por ejemplo, en el modelo de árbol de decisión, un colegio sería capaz de obtener las variables (es decir, las facetas) que más diferencian los intereses de sus alumnos en función de la personalidad. De esta forma, podrían obtener un **mapa de la personalidad** de los alumnos, o por lo menos situarles en el espectro de las facetas de personalidad que más van a influir a la hora de decidir su carrera (y por ejemplo organizar actividades de diversos tipos en función de estas facetas). Esta información sería muy valiosa para los colegios ya que de otra forma sería difícil de conseguir y los colegios podrían por lo menos transmitirles a sus alumnos este "mapa" de la personalidad, que haría que los alumnos se conocieran mejor a sí mismos.

En cuanto a los modelos Random Forest y la Red Neuronal, se incluyen porque son modelos que generalmente **funcionan muy bien a la hora de predecir**. Aunque perdemos interpretabilidad, si la diferencia en predicción fuese muy grande con respecto a los otros modelos, se implementarían los modelos mencionados. Por esto decidimos incluirlos en nuestra maqueta.
Por último, es interesante hacer una comparación del acierto de los modelos, aunque en nuestro caso los resultados como tal no sean concordes con la realidad.

1. Modelo KNN

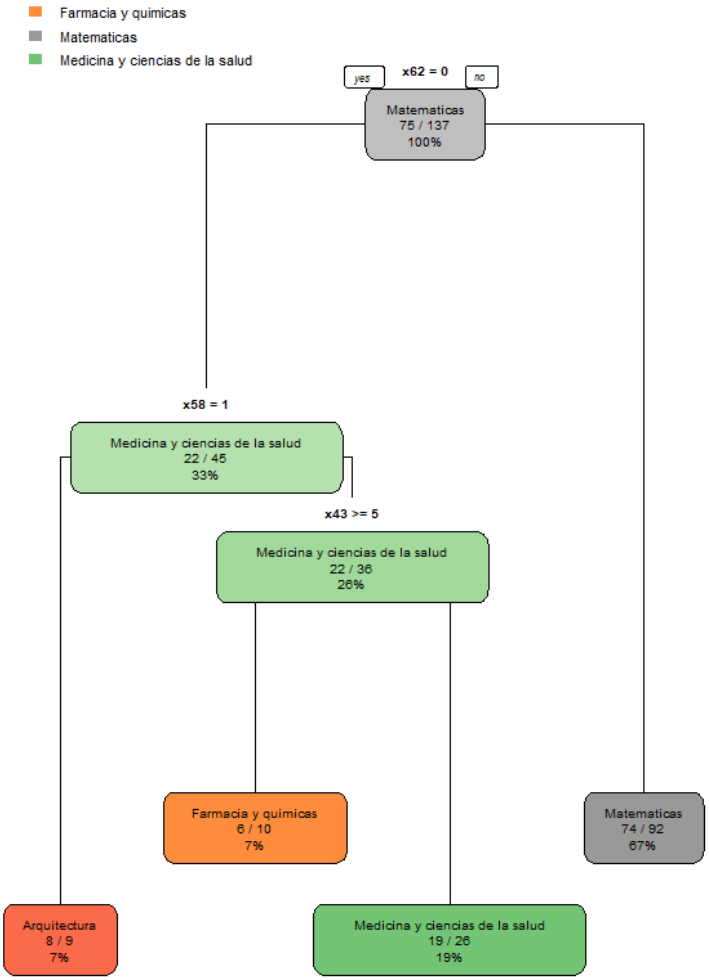
Comenzamos probando con el parámetro de número de vecinos, *k*, en *k*=3. Asignando un 75% de los datos como entrenamiento, en la predicción del resto de datos obtuvimos un 54.28% de acierto. No es un buen resultado, así que ajustamos este hiperparámetro por validación cruzada. El resultado fue que el mejor *k* se obtiene en *k*=5, y con un 80% de los datos de entrenamiento, obtuvimos un 60.71% de acierto.
En este caso se mejora con respecto al resultado anterior, pero realizando las predicciones 10 veces con distintos conjuntos de train y test, la media del acierto en este caso fue de 45.71%. Este valor es muy bajo, por tanto, en este caso descartaríamos este modelo.
Código

2. Árbol de decisión

Tanto en este modelo como en los siguientes, los nombres de las variables daban problemas a la hora de incluirlas en el modelo. Para solucionarlo, decidimos renombrar las variables como "*y*" para la variable a predecir (la carrera ideal) y "*x1*", "*x2*", ... para las variables explicativas (las variables del cuestionario relevantes). También

construimos una tabla con los nombres originales de las variables y los modificados, y así poder interpretar los árboles.
Código

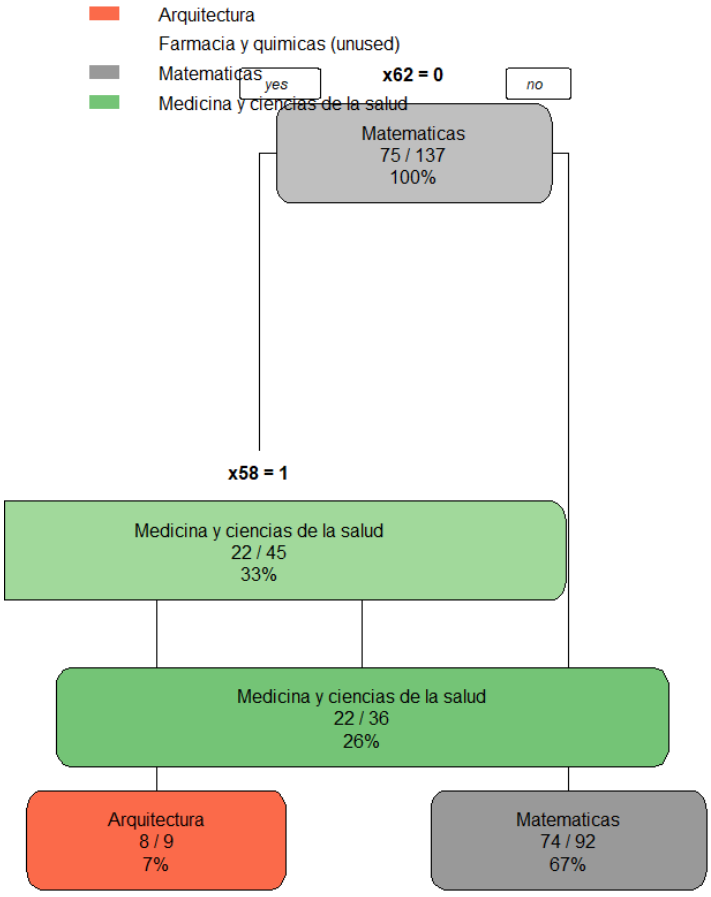
En un primer modelo inicial, las variables que más diferenciaron fueron "*matemáticas fav*", "*dibujo técnico fav*" y "*biología fav*". Prediciendo con un conjunto de entrenamiento del 75%, obtuvimos un 88.2% de acierto, un resultado bastante alto.
Después realizamos validación cruzada para determinar el número óptimo de nodos. Este número se obtuvo con 3 nodos. A continuación se muestra el árbol resultante de implementar el modelo con la función ***rpart()***, con la que obtuvimos un 70.5% de acierto.



La **interpretación** que se obtiene de este árbol es la siguiente:
Si matemáticas era de tus asignaturas favoritas, se recomienda matemáticas.
Si no lo era, y dibujo técnico era de tus asignaturas favoritas, se recomienda arquitectura.
Si ni matemáticas ni dibujo técnico eran de tus asignaturas favoritas, se ve si estás muy de acuerdo con la afirmación "sigo un horario". Si lo estás, se recomienda farmacia y químicas, si no lo estás, medicina y ciencias de la salud.

Por tanto, podemos afirmar que **existe cierta coherencia** en la clasificación con lo que uno esperaría.

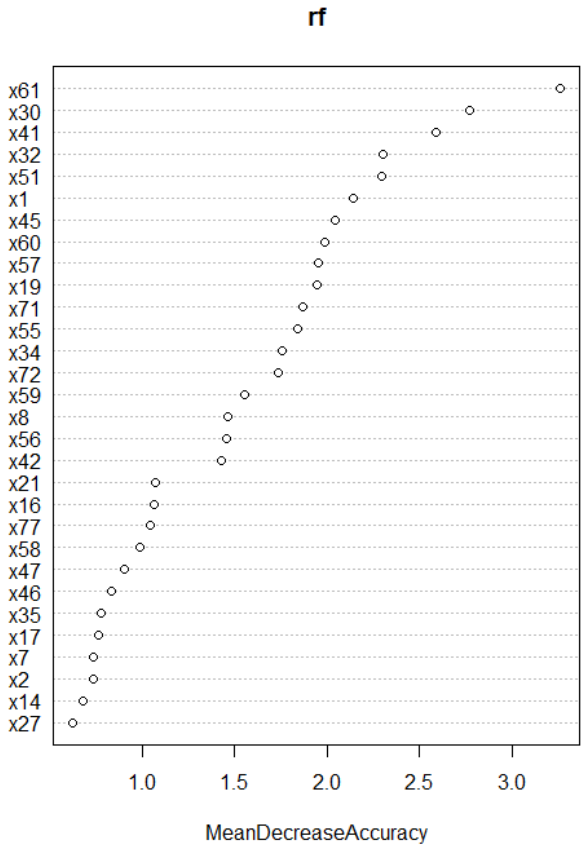
El siguiente paso fue buscar el mejor valor del hiperparámetro de complejidad (*cp*) por validación cruzada. Obtuvimos un acierto del 79.4% y este es el árbol correspondiente:



Podemos ver como es el mismo árbol que el anterior pero cortado en un nivel menos, es decir, agrupando los dos últimos nodos hoja de medicina y farmacia en medicina.
Por tanto, en este caso perderíamos la clase de farmacia y químicas. Este aspecto sería importante a tener en cuenta con unos datos reales y las instituciones deberían valorar si les merece la pena descartar alguna carrera con pocos participantes por mejorar la predicción en media.
Código

3. Random Forest:

En primer lugar, optimizamos el hiperparámetro del número de variables a considerar en cada corte (*mtry*). Se obtuvo que el mejor valor de *mtry* se daba con *mtry*=1. En el modelo correspondiente, este es el gráfico de importancia de las variables:



Las variables con más importancia son:
x61 ->biologia_fav
x30 -> No tengo mucha imaginación
x41 -> No me importa ser el centro de atención
x32 -> No me interesan mucho los demás
x51 ->literatura_fav

Lo que podemos destacar de este modelo con respecto al de Árbol de decisión es que el de Random Forest parece tener más en cuenta las **variables de personalidad**, en vez de centrarse tanto en las de las asignaturas.
Mirando más en detalle, podemos deducir asociaciones entre las variables de personalidad con los rasgos de OCEAN. Por ejemplo, podríamos asociar la variable "no tengo mucha imaginación" con ser la creatividad (fusión de *Openness* y *Conscientiousness*), "no me importa ser el centro de atención" con la extroversión, y "no me interesan mucho los demás" con una fusión entre amabilidad y extraversión.

Además, en psicología se ha demostrado que uno de los pocos rasgos de la personalidad que difieren entre hombres y mujeres es el interés por las cosas y por las personas, respectivamente. Por tanto, es claro que este es un aspecto de la personalidad relevante a la hora de decidir la carrera, y parece que este modelo lo tiene en cuenta. Así que se podría considerar como un **buen candidato a ser implementado** (independientemente de la tasa de acierto obtenida en esta maqueta).

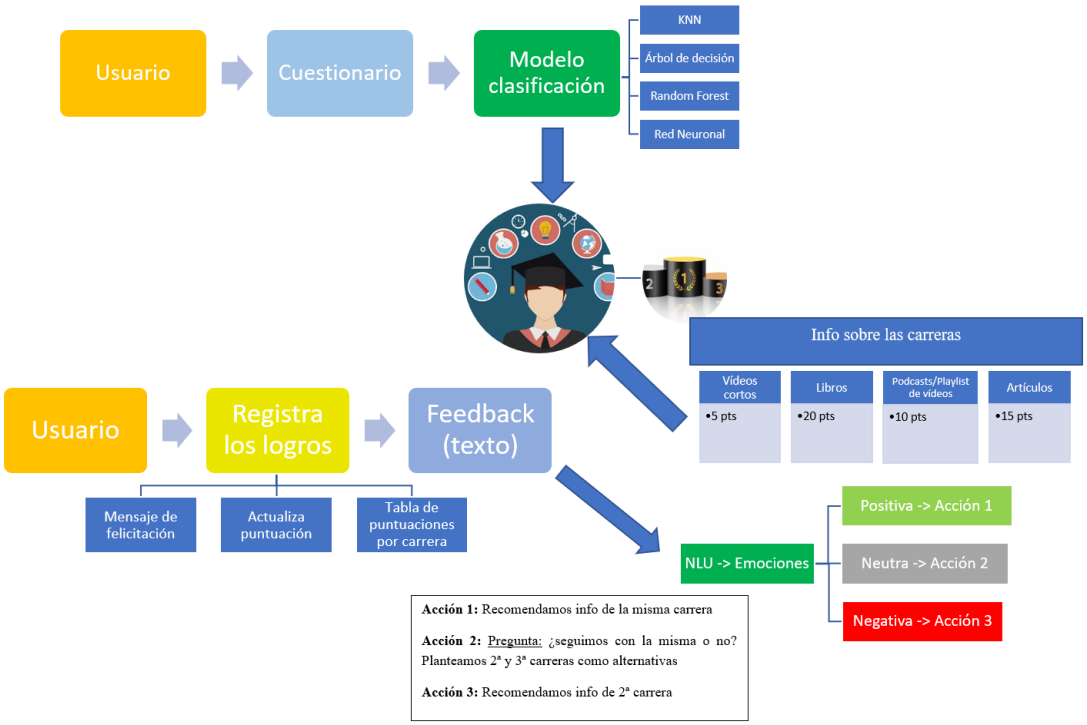
En este modelo obtuvimos una tasa de acierto del 62.8%. El modelo predecía siempre '*Matemáticas'* como carrera. Esto se debe al **desbalanceo** presente en los datos, ya que obtuvimos muchas más respuestas de Matemáticas que de las otras carreras.
Código

4. Red Neuronal:

Por último, implementamos una Red Neuronal sencilla: de una capa oculta con 5 nodos. Calculamos la tasa de acierto media al realizar este proceso 300 veces. El resultado final de *accuracy* es de un 41.4%.
Código

3.2 Esquemas de la maqueta

Esquema 1:



Como se puede observar, el proceso empieza con el usuario respondiendo al cuestionario. Las respuestas al cuestionario forman parte del nuevo input que recibe el modelo y con el que hace la predicción. Los modelos probados en esta maqueta han sido **KNN**, **Árbol de decisión**, **Random Forest** y **Red Neuronal**.

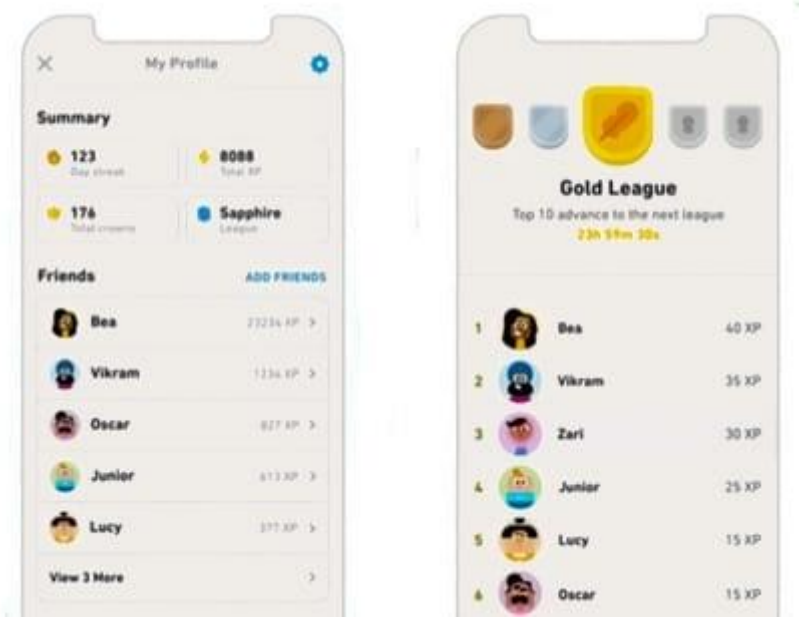
Una vez hecha la predicción de las tres carreras que tienen más probabilidad de ser las adecuadas para el usuario, el usuario recibe información sobre la carrera con más probabilidad. La información vendrá dada por vídeos cortos, libros, artículos y podcasts/playlist de videos.
Cada recurso tendrá asociada una puntuación que más tarde se registrará, se acumulará según se vaya realizando y se comparará con las puntuaciones de los demás usuarios de

la misma carrera. Además, se incluye un mensaje de felicitación y ánimo cada vez que se registre un logro de haber completado la tarea.
Estos recursos se dividirán por niveles de complejidad, podríamos pedir la edad del usuario y así poder saber sus conocimientos previos y tener los recursos divididos teniendo esto en cuenta, y por tiempo necesario para completarlos; así el usuario podrá comenzar con los recursos como los videos cortos que no necesitan tanto tiempo y a medida que vaya creciendo su interés por la carrera será más fácil que quiera invertir más tiempo en recursos que lleven más tiempo como las playlist o libros, que además en un principio tendrían que comprar o ir a hasta una biblioteca a por ellos.
Como ejemplo de estos recursos hemos seleccionado uno de cada tipo para la carrera de arquitectura:

- Libro: https://www.amazon.es/Entender-arquitectura-Leland-M-Roth/dp/8425217008/
- Articulo: https://noticias.arq.com.mx/Detalles/10624.html#.YIe3pxKxVNg
- Video: https://youtu.be/9npGxrKaXMo
- Playlist/podcast:https://youtube.com/playlist?list=PLTVPrD5CVmnjolGeokd8KnZ-KbKqZtrmA

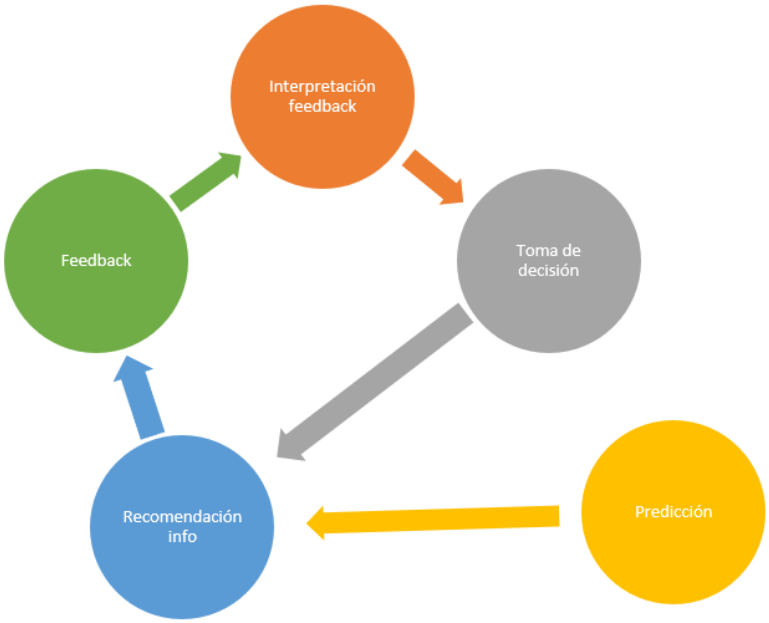
Consideramos que estas cualidades del proceso son muy importantes ya que consiguen **gamificar el aprendizaje** y por tanto involucran mucho más a los estudiantes.

Ejemplo de app gamificada:



Cuando se realizan las tareas el usuario nos envía feedback sobre su experiencia. Mediante un modelo de **NLU** (podría ser el servicio de IBM Watson o el de Google) extraemos la emoción asociada al texto (positiva, negativa o neutra) y en función de la emoción recibida, se actúa conforme indica el siguiente esquema:

Esquema 2:



Este segundo esquema muestra el proceso a un nivel más alto, representando cómo una vez recibimos el feedback y tomamos la decisión de recomendar info sobre la misma carrera o no, se obtiene nuevo feedback, que volvemos a interpretar y volvemos a tomar una decisión.
Este bucle se realizaría 2 o 3 veces, hasta que el usuario quedase satisfecho con la elección de la carrera.

4. Implementabilidad y business case del proyecto (impacto social y/o económico)

Como es de esperar, nuestros posibles usuarios serían estudiantes de entre catorce y dieciocho años, pero no los únicos; cualquier persona que no esté satisfecha con su trayectoria profesional y quiera hacer un cambio y volver a estudiar, lo cual es cada vez más común, podría beneficiarse de este servicio. Por ello utilizaremos dos vías paralelas de desarrollo del proyecto: **Aplicaciones A.M.** (aplicaciones multiplataforma) y **A.W.** (aplicaciones web).

La aplicación multiplataforma sería una aplicación de **acceso público** para dispositivos móviles (IOS, Android); con la que cualquier persona interesada en el servicio podría acceder a este. Como estaría orientada a **personas jóvenes** y que, por tanto, no tendrán

presupuesto para una aplicación de pago, la mejor opción de negocio sería a través de **anuncios**.
Creemos que es una oportunidad muy buena para los **patrocinadores** ya que como los usuarios de la aplicación serían pertenecientes a un grupo concreto (estudiantes de bachillerato y secundaria), esto sería muy interesante para posibles patrocinadores cuyos clientes objetivo sean personas de este grupo. Podemos imaginarnos empresas relacionadas con la educación, el deporte, tecnología, viajes,... como potenciales patrocinadores.

La aplicación web sería de una web de acceso restringido a miembros de entidades que compren el acceso a ella y por tanto no tendría publicidad.
Esta opción sería ideal para ser utilizada por **centros educativos** (en un principio de secundaria y bachillerato, pero se podría estudiar una opción para centros de educación primaria), ya que así la podrán utilizar dentro del instituto.

Creemos que sería interesante ofrecer a estos centros educativos un paquete por el que se pudiese dar acceso al servicio desde el aula virtual, que es un servicio de *moodle* del que disponen todos los institutos o por lo menos los públicos y como cada alumno tiene acceso a su propio espacio. Sería una gran opción plantearnos la posibilidad de integrar nuestro servicio a este para que así los alumnos dispusieran de todos los recursos fácilmente y así fuesen más propensos a utilizarlos. Los alumnos disponen en la mayoría de cursos de una hora a la semana de tutorías, aunque dependiendo de la comunidad autónoma en bachillerato no todos la tienen, sí que todos tienen tutores. Por lo que sería sencillo destinar algunas de estas horas a que los alumnos utilicen la aplicación.

Otros posibles usuarios serían **universidades** que podrían ofertar el servicio como parte del contenido promocional que suelen ofrecer cuando ofertan su universidad a posibles nuevos alumnos. Esto sería posible ofreciendo un paquete para un número determinado de usuarios que las mismas universidades podrían repartir con los medios usuales; ferias tipo Aula, conferencias, zona “nuevos alumnos” de sus páginas web, etc.

Desde un **punto de vista social,** creemos que es una herramienta realmente útil, ya que los centros educativos no suelen disponer de un servicio de este estilo y eso deja a los estudiantes teniendo que tomar decisiones realmente importantes sin ningún tipo de asesoramiento, lo cual no solo puede llevar a decisiones de última hora y mal informadas, sino que acarrea un nivel adicional de estrés en un momento ya de por sí especialmente estresante como es bachillerato y más aún selectividad, que es cuando todo alumno que desee continuar sus estudios deberá tomar las decisiones finales sobre qué grados cursar.

Además, una vez tomada la decisión de que la carrera que va a comenzar es la ideal, el alumno se sentiría **motivado**, habríamos conseguido despertar su **vocación** por una materia en la que de otra forma no se habría adentrado.
Esto aporta una sensación de **seguridad** en ti mismo, de que estás yendo por el camino correcto, que sin duda juega un papel vital en tener una carrera profesional de éxito. Además, conocería más en detalle y profundidad la materia y los conocimientos que va a adquirir en los próximos años de universidad. Es muy común entre jóvenes que

comienzan un grado universitario que, al empezar las asignaturas en cuestión, no son lo que se esperaban y por tanto se sienten muy desmotivados.

Más aún, estos conocimientos adquiridos no solo corresponderían a la materia que fuesen a estudiar, sino también de las **salidas** y el **perfil profesional** que adquirirían al terminar el grado. Por tanto, habría personas que llegarían a la universidad, quizás las materias iniciales no les gustan mucho, pero saben que sí quieren llegar a ser los profesionales en los que se convertirían al terminar la carrera, y así seguirían motivados y con un espíritu ambicioso, que de otra forma no tendrían.

Por todo esto creemos que si se desarrollase la aplicación a gran escala, tendría un **impacto** realmente significativo y extremadamente **positivo** en las carreras profesionales de muchas personas. Pensando más a lo grande, si la aplicación tuviese mucho éxito y fuese conocida, los estudiantes confiarían a priori en que esta aplicación les fuese a ayudar a tomar esta importante decisión, a que tengan un futuro próspero y, por tanto, se sentirían mucho más motivados a la hora de usar la aplicación como tal y profundizar en los materiales que les ofreciésemos.
En los desarrollos a futuro explicaremos más en detalle de qué formas la aplicación podría obtener este éxito y una buena reputación.

Hemos descrito cómo la consecuencia primordial del uso de esta aplicación sería despertar la vocación y motivar a los estudiantes a perseguir la carrera profesional que ellos quieren. Un corolario de esto es que estas personas se convertirían en profesionales con vocación por su carrera y altas aspiraciones, o por lo menos el nivel medio en los usuarios sería más alto que si estos usuarios no hubiesen utilizado la aplicación.
En consecuencia, podemos esperar un **aumento en la calidad de formación** de estos usuarios, que sin duda sería un aspecto muy positivo a tener en cuenta por parte de las empresas e instituciones que busquen este talento. Por tanto, el **impacto económico** de la aplicación vendría dado por el aumento en ingresos y crecimiento de las empresas e instituciones influenciado por estos profesionales con una alta ambición y formación. Por ejemplo, los colegios que adquiriesen esta aplicación mejorarían sus resultados académicos y por tanto su reputación, aumentando por tanto su demanda e ingresos.

Por todo esto, creemos que nuestra aplicación tiene un **gran potencial** y no sería complicado encontrar empresas dispuestas a escalar esta maqueta y desarrollar los servicios que hemos descrito. Para ello sería esencial transmitir las ideas del potencial impacto social y económico que tendrían nuestras aplicaciones.

5. Desarrollos a futuro

Hemos decidido describir los desarrollos a futuro según los siguientes puntos:

5.1. Escalabilidad

El primer paso si quisiéramos de verdad crear esta *startup* sería pasar de una maqueta a una aplicación propiamente dicha. Para ello tendríamos que encontrar una empresa dispuesta a diseñar la arquitectura tanto de la aplicación multiplataforma como de la aplicación web.
Una vez hecho esto, trataríamos de recopilar más observaciones similares a las que obtuvimos al pasar el cuestionario, añadiendo por tanto más carreras a predecir y mejorando la validación de los modelos.
Por supuesto, habría que hacer un estudio más exhaustivo a la hora de comparar los diferentes modelos y decidir cuál es el mejor. Seguramente sería buena idea que en la arquitectura de la aplicación se incluyese que periódicamente se compruebe cuál es el mejor modelo para predecir, y se actualice debidamente.

5.2. Marketing

Tanto a la hora de encontrar posibles diseñadores de apps multiplataforma y aplicaciones web, como a la hora de captar usuarios, es muy importante atraer al máximo número de potenciales interesados en nuestra app, y por ello el marketing asociado a la empresa sería de gran importancia.

Para atraer a desarrolladores, sería esencial definir y comunicar claramente los objetivos de la empresa y nuestra visión con este producto, así conseguiríamos **credibilidad** por su parte. Es posible que los desarrolladores duden de la capacidad de que estos modelos realmente predigan qué carrera es la ideal (sobre todo si nunca han trabajado con modelos de ML), por tanto, también podría ser interesante hacerles ver de lo que son capaz y el valor de los modelos de Machine Learning en general y cómo son utilizados hoy en día en empresas altamente exitosas por todo el mundo (recomendador Netflix, …), y así despertaríamos **esperanza en el éxito** de nuestra aplicación.

Por otra parte, una vez desarrollada la app multiplataforma, generaríamos **publicidad** para atraer al máximo número de usuarios. Hoy en día el marketing digital es muy potente, sobre todo en redes sociales y especialmente si está dirigido a adolescentes. Así que, por ejemplo, crearíamos una cuenta de Instagram sobre la app y publicaríamos anuncios en forma de posts e historias. En estos anuncios podríamos incluir promociones o regalos si adquieren la aplicación en cierto periodo de tiempo, proporcionando por ejemplo ventajas exclusivas en la aplicación. También, conforme pasase el tiempo incluiríamos las buenas valoraciones de la gente que fuese usando la app.
También sería interesante intentar colaborar con universidades para que ellas ofreciesen la app como opción para encontrar una futura carrera ya que esto nos daría fiabilidad por parte de los posibles usuarios.
Un elemento clave también a la hora de definir nuestra *startup* sería definir un **nombre atractivo** de la empresa/aplicación y un buen **eslogan**, posiblemente.

5.3. Futuras versiones de la app

Una vez desarrolladas las aplicaciones, podemos imaginarnos cómo podrían desarrollarse las futuras versiones.

Un elemento que podría ser interesante sería **captar profesionales** de las diferentes carreras y que los usuarios pudiesen comunicarse con ellos para resolver dudas más concretas y conocer las carreras más de primera mano, a través de un chat o por correo.
También podríamos incluir ***reviews*** acerca de las experiencias de los usuarios que usaron la aplicación y que contasen qué impresión les dio acerca de la carrera que eligieron y si fue la acertada, y también cómo fue el camino que siguieron después.
Esta **información** sería **muy valiosa** para todos los estudiantes y podría cubrir casos más específicos sobre los intereses y vocación de las personas, o sobre cómo son impartidas las clases en cierta universidad, además de la experiencia general de estudiar en otra ciudad.
Otra característica relacionada con la **gamificación** de la aplicación que podríamos ampliar podría ser que según van avanzando en los niveles, se desbloqueen conferencias online exclusivas presentadas por expertos o estudiantes en las distintas carreras. Sería un incentivo más para que los estudiantes realicen las tareas que les ofrezcamos.

5.4. Consecuencias debidas a la reputación

Supongamos que desarrollamos la aplicación y tiene mucho éxito: mucha gente la utiliza y los resultados y la satisfacción parecen ser prometedores. Para crear una buena reputación podríamos hacer un **seguimiento** de las personas que más se beneficiaron de usar nuestra aplicación y de esta forma adquirir hechos remarcables de cara al público.
Unos ejemplos de estos hechos podrían ser:
El 95% de las personas satisfechas con nuestra aplicación consigue trabajo al terminar los estudios.
Un 80% de las personas satisfechas con nuestra aplicación afirma que:

han conseguido el trabajo que tienen gracias al uso de la aplicación
son personas más ambiciosas y vocacionales gracias al uso de la aplicación
la aplicación consiguió despertar su vocación por la carrera
la aplicación le ayudó a encontrar su carrera ideal

Con estos hechos y con una buena publicidad de estos, conseguiríamos una **buena reputación** que podría desembocar en que los departamentos de RRHH que buscan talento valoren positivamente que el candidato haya decidido hacer su carrera por usar nuestra aplicación. Por supuesto esto sería un gran logro para la empresa y en esencia podría hacer que la gente viese a la empresa no solo como una empresa que te orienta en la elección de elegir tu carrera ideal, sino como una empresa que fundamentalmente te ayuda a conseguir tus objetivos vitales y, posiblemente, a ser más feliz.

6. Bibliografía

- Big Five Personality Test - Open Psychometrics
- Preguntas Big Five
- Samuel A. Stein, Yiwein Chen, Gary M. Weiss, Daniel D. Leeds, A College Major Recommendation System

7. Apéndices

Código preprocesamiento de los datos

```
cont = 0
for i in df['Cual era tu asignatura favorita en el colegio/instituto  (Escoge una, dos o ninguna)']:
    while i.count(',') < 2:
        i = i + ',0'
    df.loc[cont,'Cual era tu asignatura favorita en el colegio/instituto  (Escoge una, dos o ninguna)'] = i
    cont = cont +1


cont = 0
for i in df['Qué asignatura detestabas en el colegio/instituto  (Escoge una, dos o ninguna)']:
    while i.count(',') < 2:
        i = i + ',0'
    df.loc[cont,'Qué asignatura detestabas en el colegio/instituto  (Escoge una, dos o ninguna)'] = i
    cont = cont +1


df.to_excel (r'C:\Users\mzaba\Documents\Matemáticas\4º\2º Semestre\Aprendizaje automático. Machine learning\Espacio pr
```

```
datos = read_excel("C:/Users/mzaba/Documents/Matemáticas/4º/2º Semestre/Aprendiz
df = data.frame(datos)



df[,4] = gsub(",", " ", df[,4])
df[,5] = gsub(",", " ", df[,5])

it_train_4 = itoken(df[,4],
                    preprocessor = tolower,
                    tokenizer = word_tokenizer,
                    progressbar = TRUE)
vocab_4 = create_vocabulary(it_train_4)


vectorizer_4 = vocab_vectorizer(vocab_4)

dtm_train_4 = create_dtm(it_train_4, vectorizer_4)



it_train_5 = itoken(df[,5],
                     preprocessor = tolower,
                     tokenizer = word_tokenizer,
                     progressbar = TRUE)
vocab_5 = create_vocabulary(it_train_5)


vectorizer_5 = vocab_vectorizer(vocab_5)

dtm_train_5 = create_dtm(it_train_5, vectorizer_5)
```

```
# 2. Quitamos variables originales y variables inútiles. Renombramos variables asignaturas
df_final = df[-c(3:5,56:57)]
dim(df_final)

cont = 53
for(i in colnames(dtm_train_4)){
  df_final = cbind(df_final,dtm_train_4[,i])
  colnames(df_final)[cont] = paste(i,'_fav', sep='')
  cont = cont + 1
}

cont = 66
for(i in colnames(dtm_train_5)){
  df_final = cbind(df_final,dtm_train_5[,i])
  colnames(df_final)[cont] = paste(i,'_hate', sep='')
  cont = cont + 1
}

# 3. Sustituimos valores NA por 3 (respuesta neutra)
df_final[is.na(df_final)] = 3
```

Esta parte del código hay que ejecutarla antes de cada modelo para volver a tener el data frame inicial:

```
# 4. Renombramos variables y creamos tabla de nombres originales y modificados
names(df_final)[1] = 'y'
nombres_var = matrix(NA, ncol=2, nrow = 78)
nombres_var[,1] = names(df_final)[3:80]

lista_renombres = numeric()
for(i in 1:78){
  lista_renombres[i] = paste("x",i, sep='')
}
nombres_var[,2] = lista_renombres


# 5. Quitamos variable grado de satisfacción
df_final = df_final[-c(2)]

# 6. Cambiamos todas las variables directamente
for(i in 2:79){
  names(df_final)[i] <- paste("x",i-1,sep='')
}

# 7. Convertimos variable a predecir en factor
df_final[,1] = as.factor(df_final[,1])
```

Código KNN

```
#KNN

library(class)
n <- nrow(df_final)
idx <- sample(n, n*0.75)
train <- df_final[idx, ]
test <- df_final[-idx, ]
y_train <- train[,  1]
X_train <- train[, -1]
y_test <- test[,  1]
X_test <- test[, -1]
y_pred <- knn(X_train, X_test, y_train, k=3)
mean(y_test == y_pred)*100
```

```
yy<-matrix(0,1,20)
for (i in 2:20)
{
  idx <- sample(n, n*0.8)
  train <- df_final[idx, ]
  test <- df_final[-idx, ]
  y_train <- train[,  1]
  X_train <- train[, -1]
  y_test <- test[,  1]
  X_test <- test[, -1]
  y_pred <- knn(X_train, X_test, y_train, k=i)
  yy[i]<-mean(y_test == y_pred)*100
}
plot(yy[1,2:20])
idx <- sample(n, n*0.8)
train <- df_final[idx, ]
test <- df_final[-idx, ]


y_train <- train[,  1]
X_train <- train[, -1]
y_test <- test[,  1]
X_test <- test[, -1]
y_pred <- knn(X_train, X_test, y_train, k=5)

mean(y_test == y_pred)*100
 xx<-matrix(0,1,10)
 yy<-matrix(0,1,10)

 for (i in 1:10)
 { idx <- sample(n, n*0.8)
 train <- df_final[idx, ]
 test <- df_final[-idx, ]
 y_train <- train[,  1]
 X_train <- train[, -1]
 y_test <- test[,  1]
 X_test <- test[, -1]
 y_pred <- knn(X_train, X_test, y_train, k=5)
 yy[i]<-mean(y_test == y_pred)*100
 xx[i]<-i
 }
 plot(xx[1,],yy[1,])
 mean(yy[1,])
 sd(yy[1,])
```

Código Árbol de decisión

```
# 1. Modelo inicial
tree.inicial=tree(y~.,df_final)
summary(tree.inicial)

# Vemos cuales son las primeras variables que diferencian

View(nombres_var)

"
62 -> matematicas fav
58 -> dibujo tecnico fav
61 -> biologia fav

"

# Predecimos en test
set.seed(1)
indices = sample(1:nrow(df_final), floor(0.25*nrow(df_final)))

tree.pred=predict(tree.inicial,df_final[indices,],type="class")

confu<-table(tree.pred,df_final[indices,1])
(confu[1,1]+confu[2,2]+confu[3,3]+confu[4,4])/34 # accuracy=0.882

# 2. Validación cruzada
cv.model=cv.tree(tree.inicial)

plot(cv.model$size,cv.model$dev,type='b') # el minimo está en 3

# Buscamos el mejor árbol con 3 nodos:
prune.boston=prune.tree(tree.inicial,best=3)

fit <- rpart(y ~ ., df_final, method = "class", cp=0)

# Vemos la exactitud en el conjunto test y dibujamos el árbol con rpart.plot
test = df_final[indices,]
preds = predict(fit, test, type = "class")
sum(preds == test$y)/nrow(test) # 0.705
rpart.plot(fit, type=1, extra = 102)
```

```
# Podamos el árbol usando el mejor valor de cp obtenido usando validación cruzada y pintamos el árbol.
pfit<- prune(fit, cp = fit$cptable[which.min(fit$cptable[,"xerror"]),"CP"])
rpart.plot(pfit, type=1, extra = 102)
preds = predict(pfit, test, type = "class")
sum(preds == test$y)/nrow(test) # acc = 0.794
```

Código Random Forest

```
set.seed(1)
indices = sample(1:nrow(df_final), floor(0.75*nrow(df_final)))

train = df_final[indices,]
test = df_final[-indices,]
MSEi = numeric()

for(i in 1:dim(df_final)[2]){
  rf = randomForest(y~., data = train, ntree = 100, mtry=i)
  rf.pred = predict(rf, test, type="class")
  realidad = df_final[-indices,1]
  MSE = mean(rf.pred==realidad)
  MSEi[i]=MSE
}
which.min(MSEi) # el menor MSE de test se obtiene con 1 variable utilizada en cada corte

rf = randomForest(y ~ ., data = train, ntree = 100, mtry=1, importance=TRUE)
varImpPlot(rf, type=1)

# Las variables con más importancia son:

'
x61 -> biologia_fav
x30 -> No.tengo.mucha.imaginación
x41 -> No.me.importa.ser.el.centro.de.atención
x32 -> No.me.interesan.mucho.los.demás
x51 -> literatura_fav
'
rf_pred = predict(rf, test, type="class")
confu = table(rf_pred,df_final[-indices,1])

# El modelo predice siempre Matemáticas. Esto se debe a la falta de datos (desbalanceo en Matemáticas)
sum(diag(confu))/sum(confu) # acc=0.628
```

Código Red Neuronal

```
accb = numeric()
for(b in 1:300){
  indices = sample(1:nrow(df_final), floor(0.75*nrow(df_final)))

  train = df_final[indices,]
  test = df_final[-indices,]

  nn <- nnet(y~ ., data=train, size=5, maxit=100, rang=0.1, decay=5e-4)
  pred <- predict(nn, test, type="class")
  cm_nn <- table(pred=pred, true=df_final[-indices,1])
  acc = sum(diag(cm_nn))/sum(cm_nn)
  accb[b]=acc
}
mean(accb) # 0.414
```

Código recompensa

```
remove.packages("stringr")
remove.packages("readxl")
remove.packages("text2vec")
remove.packages("tree")

install.packages("stringr")
install.packages("readxl")
install.packages("text2vec")
install.packages("tree")
library(stringr)
library(readxl)
library(text2vec)
library(tree)


datos = read_excel("export_dataframe.xlsx", col_names = TRUE)
df = data.frame(datos)
```

```
df[,4] = gsub(",", " ", df[,4])
df[,5] = gsub(",", " ", df[,5])

it_train_4 = itoken(df[,4],
                    preprocessor = tolower,
                    tokenizer = word_tokenizer,
                    progressbar = TRUE)
vocab_4 = create_vocabulary(it_train_4)
vectorizer_4 = vocab_vectorizer(vocab_4)
dtm_train_4 = create_dtm(it_train_4, vectorizer_4)

it_train_5 = itoken(df[,5],
                    preprocessor = tolower,
                    tokenizer = word_tokenizer,
                    progressbar = TRUE)
vocab_5 = create_vocabulary(it_train_5)
vectorizer_5 = vocab_vectorizer(vocab_5)
dtm_train_5 = create_dtm(it_train_5, vectorizer_5)

df_final = df[-c(3:5,56:57)]


cont = 53
for(i in colnames(dtm_train_4)){
  df_final = cbind(df_final,dtm_train_4[,i])
  colnames(df_final)[cont] = paste(i,'_fav', sep='')
  cont = cont + 1
}

cont = 66
for(i in colnames(dtm_train_5)){
  df_final = cbind(df_final,dtm_train_5[,i])
  colnames(df_final)[cont] = paste(i,'_hate', sep='')
  cont = cont + 1
}

df_final[is.na(df_final)] = 3

attach(df_final)

names(df_final)[1] = 'y'
nombres_var = matrix(NA, ncol=2, nrow = 78)
nombres_var[,1] = names(df_final)[3:80]

lista_renombres = numeric()
for(i in 1:78){
  lista_renombres[i] = paste("x",i, sep='')
}
nombres_var[,2] = lista_renombres

df_final = df_final[-c(2)]


for(i in 2:79){
  names(df_final)[i] <- paste("x",i-1,sep='')
}

df_final[,1] = as.factor(df_final[,1])

tree.inicial=tree(y~.,df_final)

preguntasv1 = names(df)[4:55]
preguntasv2 = gsub("[.]", " ", preguntas)
preguntasv3 = preguntasv2
preguntasv3[46] = "Me trastorno/preocupo con facilidad"
preguntasv3[31] = "Me irrito con facilidad"
preguntasv3[1] = "Cual era tu asignatura favorita en el colegio/instituto. Escoge una, d
preguntasv3[2] = "Qué asignatura detestabas en el colegio instituto. Escoge una, dos o 0
preguntasv4=preguntasv3[-c(1,2)]
preguntasv5 = c(preguntasv4,nombres_var[51:78,1])
preguntasv5[29] = "Me enfado con facilidad"
```

```
print("Comienza el cuestionario:")

vector_regalo = numeric()
for(i in preguntasv5){
  respuesta = readline(prompt = str(i))
  vector_regalo = append(vector_regalo,respuesta)
}
m1 <- matrix(vector_regalo, ncol=78, byrow=TRUE)
df <- as.data.frame(m1, stringsAsFactors=FALSE)
colnames(df) = names(df_final)[-1]
i = 1:length(df)
df[,i] = apply(df[,i], 2,function(x) as.numeric(as.character(x)))


tree.pred=predict(tree.inicial,
                  newdata = df,
                  type="class")

paste('Tu carrera ideal es ', tree.pred,'. Enhorabuena! :D', sep='')
```