



Investigación de experiencia usuaria para DashAI, app. web de Machine Learning

Diseño Centrado en Usuarios - INF337
Proyecto 3 - 10 de Diciembre del 2022
UTFSM - CC

Tabla de contenidos

Página

1. Definición del problema
2. Usuarios y audiencia
 - a. User Persona
3. Roles y responsabilidades
4. Alcance y restricciones
5. Proceso
 - Tests de usuario
 - Análisis, definición de problema y prototipo
6. Resultados
 - a. Test de Usuario
 - i. Definición
 - Tareas
 - Evaluación completitud tareas
 - Test de deseabilidad
 - ii. Piloto
 - iii. Evaluación Experiencia de Usuario (4 personas).
 1. Attrak diff
 2. Test de deseabilidad
 - b. Hallazgos
 - c. Análisis específico y mejora propuesta
 - i. Storyboard
 - ii. Scenario mapping
 - iii. Schematic diagram
 - iv. Prototipo
 - d. Discusión
7. Anexo
 - Tabla de tiempos

1. Definición del problema

Machine Learning (ML) se puede definir como “una forma de la Inteligencia Artificial (IA) que permite a un sistema aprender de los datos en lugar de aprender mediante programación explícita”¹. Se utiliza para crear **modelos a partir de ciertos datos, con los cuales se pueden hacer predicciones**. Para lograr esto, es importante que quien esté programando en este ámbito sepa de los modelos que puede implementar en un conjunto de datos, conozca los datos más apropiados y use los algoritmos correctos. *Sin embargo, no es un proceso fácil llegar, sobre todo para alguien que no esté familiarizado en este tipo de programación.*

Para hacer el trabajo anterior menos tedioso, el **Centro Nacional de Inteligencia Artificial** (CENIA) está construyendo una **interfaz gráfica interactiva** que permita realizar de forma más rápida las configuraciones necesarias de los modelos en ML y así encontrar el algoritmo adecuado para cada caso. No obstante, esta plataforma **no ha seguido un diseño centrado en personas**, por lo que es incierto si esta herramienta puede lograr su objetivo de hacer más simple el trabajo de una persona que está experimentando con ML.

Dado lo anterior y la primera parte de esta investigación: “*Investigación sobre el proceso de trabajo con Machine Learning*”², donde se realizaron principalmente investigaciones contextuales para responder a:

*¿Cómo podríamos nosotros crear una herramienta
que facilite el proceso de experimentación en Machine Learning?*

Se continúa ahora con la producción, ejecución y análisis de un **test de usuario** elaborado para la aplicación en cuestión, DashAI, con ello se plantean problemas de amplio espectro encontrados respaldándose en la evidencia, para proponer mejoras con técnicas como **storyboard, mapeo de escenarios, diagrama esquemático** y finalmente **proponer interfaces** elaboradas en figma para así mejorar la experiencia de usuario presente actualmente.

¹ <https://www.ibm.com/cl-es/analytics/machine-learning>

² [Investigación sobre el proceso de trabajo con Machine Learning, 12.'22 - Aspe, Cayazaya, Morales, Torres](#)

2. Usuarios y audiencia

Al resolver el problema de crear una herramienta que facilite el proceso de experimentación en Machine Learning, los usuarios afectados **directamente** serían los **estudiantes** que utilizarían esta herramienta para aprender de manera experimental los procesos que se realizan al momento de crear y entrenar (además de todos sus pasos indirectos) un modelo de Machine Learning, esto incluyendo modelos de machine learning puros, como también sus derivados (redes neuronales, deep learning, etc). Además, existe otro grupo de usuarios que se podrían beneficiar **directamente** de esta herramienta, estos son los **usuarios experimentados** en Machine Learning que realizan modelos de manera regular, ya sea debido a su trabajo o proyecto personal, estos usuarios utilizarían esta herramienta para facilitar las modificaciones que constantemente deben realizar en sus datos y modelos para así obtener los mejores resultados.

Por otro lado, los usuarios afectados **indirectamente** serían los **profesores**, los cuales tendrían una forma más visual de demostrar a sus alumnos los efectos que generan los cambios de datos o parámetros dentro de un modelo de Machine Learning.

Recordatorio

a. User Persona

Tomando lo elaborado en la primera parte de la investigación [[Anexo i](#)], se recuerda la User Persona, a la cual se le llamó Ricardo Jara, correspondiente a un estudiante universitario chileno de alrededor de 24 años con experiencia sobre la media en temas de Machine Learning. Junto a ello posee alta experiencia en programación y afinidad tecnológica, sin embargo su mala memoria ocasiona que frecuentemente olvide funciones necesarias para hacer su trabajo. A este usuario le gustaría poder trabajar de manera rápida y cómoda sin tener que preocuparse de manejar la limpieza de datos manualmente.

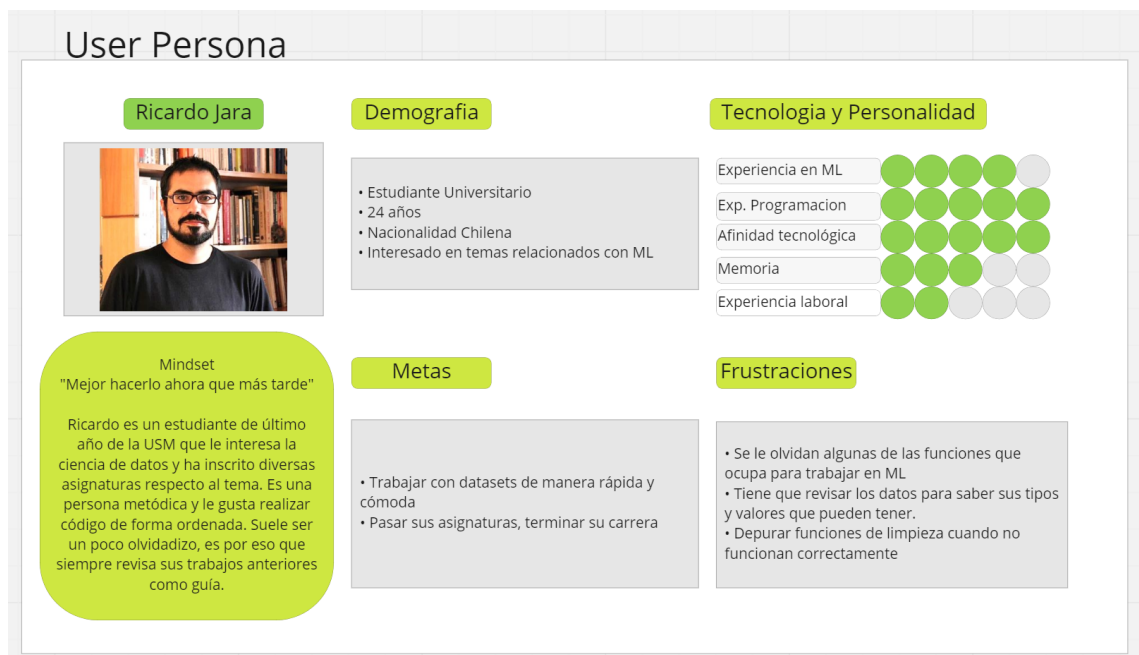


Imagen 1 - User persona desarrollado en la primera parte de la investigación

3. Roles y responsabilidades

El equipo es compuesto por cuatro integrantes, desarrollando las siguientes acciones:

- **Francisco Aspe:** Mediador en diagrama de afinidad
- **Rodrigo Cayazaya:** Mediador en Tests de usuarios
- **Javier Torres:** Mediador en scenario mapping
- **Víctor Morales:** Encargado de Storyboard

Se destaca que cada integrante participó en la elaboración del test de usuario, además de participar en la redacción del presente informe.

4. Alcance y restricciones

Alcance:

El equipo elaboró un test de usuario, el cual se ajustó con la realización de un piloto para aplicarlo con otros cuatro usuarios, todos los mencionados son estudiantes que tienen experiencia con machine learning. Cabe destacar que no se observó a la misma persona dos veces, incluyendo a la del piloto. El presupuesto para realizar los tests de usuario fue nulo, ya que fueron realizados de forma online a través de videollamada, compartiendo la aplicación localmente (versión detallada en [anexo ii](#)) mediante Visual Code Studio y junto con la generosidad de los entrevistados, permitieron que no se incurriera en gastos extras.

En cuanto al tiempo dedicado aproximado en horas por método, invertimos (como equipo) en conjunto: [N] con la elaboración del test e instalación del entorno del aplicativo, [N] en ejecución de tanto el piloto como las ejecuciones posteriores, 1³/₄ referente al Diagrama de afinidad, 1 hora con la Storyboard, ³/₄ Escenario Mapping, al igual que el schematic diagram y también [N] para el prototipado en figma . Para más detalles revisar [tabla de tiempos](#) en anexos.

Restricciones:

- Tiempo limitado para conseguir usuarios para los test de usuario, lo que produjo que no existiera mucha diversidad en el perfil de individuo, aunque se realizó una continuación con los trabajados en la primera parte de la investigación [\[Anexo i\]](#).
- Realización del test de usuario vía online con pantalla compartida, pudiendo existir distracciones en los usuarios por temas externos, como también omitir posibles detalles del ambiente e interacción no visibles por esta vía.

5. Proceso

Tests de usuario

Debido a la naturaleza del proyecto, las personas debían tener un conocimiento mínimo de Machine Learning para pedirles que realicen las tareas organizadas, además debido al tiempo y facilidad se optó por trabajar con las mismas personas observadas en la primera parte de la investigación [[Anexo i](#)], sumando una nueva adicional para el piloto.

En más detalle, todos los usuarios son o fueron estudiantes de Ingeniería Civil Informática de la Universidad Técnica Federico Santa María del campus San Joaquín (Santiago). La estructura de la evaluación se realizó el martes 6 de diciembre, para luego el piloto realizarlo el día siguiente (7 de diciembre) por la mañana y los test mejorados ese mismo día en el transcurso de la tarde.

Para poder visualizar y tomar nota de las interacciones usuarias, se le pidió a cada usuario su permiso para grabar la sesión y voluntariamente compartir su cámara y su pantalla, a manera de simular una interacción de observación presencial, observando el navegador y sus expresiones faciales. Como aspecto más técnico, cabe mencionar que los usuarios no instalaron DashAI, sino que la app fue compartida mediante la extensión LiveShare de Visual Code Studio que con una cuenta de github permite compartir servidores locales, ahorrando gran cantidad de tiempo previo.

Análisis, definición de problema y prototipo

Durante los test de usuarios se tomaron **notas de cada tarea definida** asociadas a las interacciones con la interfaz, tiempo en completarla (si lo hizo), complejidades presentadas, comentarios y mejoras explícitas informadas por el usuario; esto con ayuda de la técnica de **think aloud**. Estas notas se escribieron en una plantilla Google Sheets [[Anexo iv](#)].

El día 9 de diciembre se discutieron y organizaron, juntando adicionalmente los resultados obtenidos automáticamente de los **test complementarios** de attrak diff y deseabilidad, creando un **diagrama de afinidad** en donde se presentan los hallazgos más importantes clasificados en categorías, teniendo una visión más amplia de la plataforma y lo que proyecta.

Una vez realizado el análisis de lo nombrado, se desarrolló el problema que encontramos genera una mayor oportunidad de mejora y facilidad de resolución: "*Pruebas del modelo, inputs*". Así como primer acercamiento se generó un **Storyboard** evidenciando el contexto y el problema específico.

Posteriormente el 10 de diciembre, se bajó en abstracción el escenario planteado, generando un **scenari mapping** donde se exploraron soluciones del problema en concreto, así como posibles mejoras en ámbitos cercanos que pueden discutirse en una siguiente instancia. Teniendo ya soluciones específicas al problema, se continuó experimentando con **schematic diagram** los posibles flujos para presentar estas ideas, junto a la identificación de dos tipos de usos destacables, básico (input manual), avanzado (input por lotes).

Por último, en los días posteriores y con los anteriores pasos dispuestos se agrupó la información creada, para así, generar un **prototipo** de alta fidelidad en Figma. Concluyendo esto se evaluó el trabajo elaborado y se discutieron mejoras para una próxima iteración.

6. Resultados

a. Test de Usuario

i. Definición

Inicialmente se separó el funcionamiento de la aplicación en las diferentes vistas que conlleva, subida de archivo, selección del modelo, modificación de parámetros, vista de resultados y prueba mediante inputs; con ello se pensó en diferentes formas de realizar tareas incrementales teniendo en cuenta nuestro user persona y el objetivo de evaluar generalmente a DashAI, llegando a la siguiente clasificación:

Tareas

1. *Subir el dataset "iris" y realizar modelo de vecino más cercano con parámetros por defecto para encontrar el % de asertividad.*

Se comienza teniendo en cuenta la menor interacción del usuario (camino por defecto), para evaluar si el sujeto identifica las partes importantes de la aplicación como el resultado del modelo.

2. *Colocar un input para observar la predicción. Puede revisar el dataset si lo encuentra necesario.*

Luego Esta tarea se añadió, ya que la vista de pruebas de modelo no tiene ninguna información extra, por lo que se pensó podría dificultar al usuario, posteriormente se añadió la revisión del dataset, esto se explica en la sección [Piloto](#).

3. *Sin cambiar de modelo, cambiar los parámetros a:*
 - Número de vecinos: 10 vecinos
 - Pesos: Distancia
 - Algoritmo: automático*volver a correr el modelo.*

Esto para entender si el usuario reconoce el flujo de la aplicación e identifica el sector específico a modificar (sin ser necesario el conocimiento del significado de parámetros).

4. *Subir el dataset twitter, realizar un modelo NumericalWrapperForText, con parámetros:*
 - Clasificador numérico: RandomForest
 - Número de estimadores: 100
 - Profundidad máxima: 3
 - División mínima de muestras: 2
 - Muestra mínima de hojas: 1
 - Muestra máxima de hojas: 5
 - Random: 2
 - Cantidad mínima de ngramas: 1
 - Cantidad máxima de ngramas: 3*y realizar una predicción mediante un input libremente ingresado.*

Esta tarea se pensó originalmente como la final debido a que abarca el flujo completo, aunque con modificaciones debido al diferente tipo de modelo requerido, si bien se dan instrucciones, es necesario el entendimiento en casi completitud del usuario sobre la aplicación.

5. *Pregunta Bonus: ¿cómo lo mejorarías?*

Esta pregunta se añadió luego del [piloto](#), debido al nivel posible (avanzado) del usuario en cuestión, para más detalles revisar la sección correspondiente.

Evaluación completitud tareas

Con las tareas ya creadas se discutió la evaluación de estas y luego de pasar por varias alternativas, se llegó a la conclusión de utilizar, el cuestionario AttrakDiff, que evalúa tanto la interfaz visual como el desempeño del software en estudio, al finalizar todas las tareas. En particular, se escogió esta evaluación frente a otros tipos de cuestionarios por los siguientes motivos:

- Es menos subjetivo que otros cuestionarios del mismo tipo (como por ejemplo SUS)
- Preferimos medir la satisfacción del test completo para obtener mejores impresiones de usabilidad y la experiencia de las personas con DashAI, por este motivo se realizó el cuestionario al terminar todas las tareas.
- Para no gastar tanto tiempo del usuario se realizó sólo este cuestionario en vez de realizar un cuestionario por cada tarea evaluada (por ejemplo NASA-TLX) . Por este mismo motivo se escogió la versión corta de AttrakDiff para no gastar tanto tiempo del usuario y porque el cuestionario era suficiente para el estudio con las 10 categorías que tiene.

Se adjunta en la sección de [Anexo. n°v](#), el enlace a la versión del cuestionario utilizada (short version).

Test de deseabilidad

Para evaluar la facilidad de uso del sistema se ocupó un cuestionario de 25 palabras seleccionadas de Microsoft Desirability Toolkit³ que en su totalidad presenta 118, se eligieron equilibrando términos tanto positivos como negativos. Así se propuso la siguiente lista

1. Accessible / Accesible
2. Appealing / Atractivo
3. Complex / Complejo
4. Comfortable / Cómodo
5. Confusing / Confuso
6. Convenient / Conveniente
7. Dull / Aburrido

³ Microsoft Desirability Toolkit, desde <https://www.nngroup.com/articles/desirability-reaction-words/>

8. Easy to use / Fácil de usar
9. Efficient / Eficiente
10. Friendly / Amigable
11. Frustrating / Frustante
12. Hard to use / Difícil de usar
13. Helpful / Útil
14. Intimidating / Intimidante
15. Intuitive / Intuitivo
16. Organized / Organizado
17. Overwhelming / Abrumador
18. Professional / Profesional
19. Reliable / Novedoso
20. Rigid / Rígido
21. Simplistic / Simple
22. Slow / Lento
23. Stable / Estable
24. Time-consuming / Consume mucho tiempo
25. Trustworthy / Confiable

Las palabras fueron expuestas en primera instancia en inglés para luego traducirlas, el usuario las respondió en un formulario de Google.

Se adjunta link a test de deseabilidad: <https://forms.gle/B34QvhCvZx2Qmngo6>

ii. Piloto

En primera instancia el test de usuario elaborado se probó en un usuario, siendo el añadido a los ocupados en primera parte de la investigación [Anexo i], así se encontraron algunas falencias/problemáticas que fueron mejoradas, tales como:

- La adición de la **posibilidad para revisar el dataset con anterioridad**, si el usuario lo desea, esto para que conozca del contexto del problema y pueda intentar deducir la entrada válida para la tarea correspondiente (tarea 2)
- Para **usuarios con más fluidez en ML** (tiempo de desarrollo tareas 1-4 sin problemas, implicando un entendimiento rápido de la herramienta), se **añadió una pregunta/tarea extra N°5: “¿Cómo mejoraría el modelo mediante los parámetros?”** esto para observar si sabría cómo manipular adecuadamente la aplicación para mejorar el rendimiento de su modelo, algo comúnmente realizado al desarrollar actividades de ML.
- La **entrega de tareas por medio escrito**, esto ya que se documentó consultas directas para continuar las tareas casi paso a paso, debido por ejemplo a la dificultad para retener los valores de los parámetros requeridos para completar la tarea 5.
- **Traducción y cambio de ciertas palabras similares en test de deseabilidad**, esto debido a que el usuario presentó confusión con la diferencia entre algunas ellas, lo cual fue tomado en cuenta y corroborado, junto con la traducción para mayor facilidad de respuesta.

iii. **Evaluación Experiencia de Usuario (4 personas).**

Como se mencionó anteriormente, para realizar una objetiva y más exacta medición de la experiencia, se ejecutaron al final de cada sesión, el test Attrak Diff relacionado a experiencia usuaria y un test de deseabilidad asociado a la facilidad de uso; esto como complemento de las notas analizadas en la siguiente sección, [Hallazgos](#). Los resultados de cada test son detallados a continuación.

1. Attrak diff ⁴

Para iniciar, se compara lo relacionado a factores pragmáticos (pragmatic quality - PQ), relacionados a utilidad y usabilidad vs. hedónicos (hedonic Quality - HQ), relacionados a estimulación e identificación; con ello se establece una matriz donde lo ideal es que se tenga una gran ponderación de ambos, de lo contrario la aplicación no sería un aporte real al usuario.

En el caso de lo observado para DashAI (imagen 2), se logra este objetivo, sin embargo, puede caer ligeramente en ser un producto orientado a una tarea específica, “generación del modelo (parámetros)” la cual es detallada más adelante en la sección de [hallazgos](#).

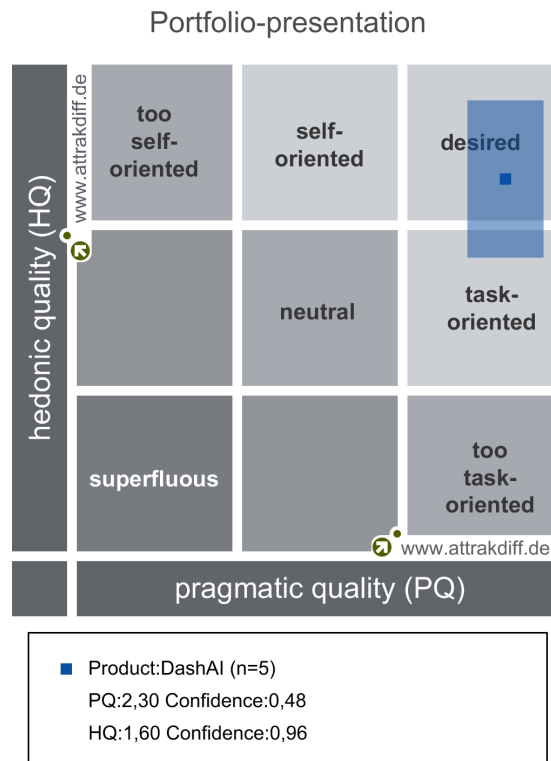
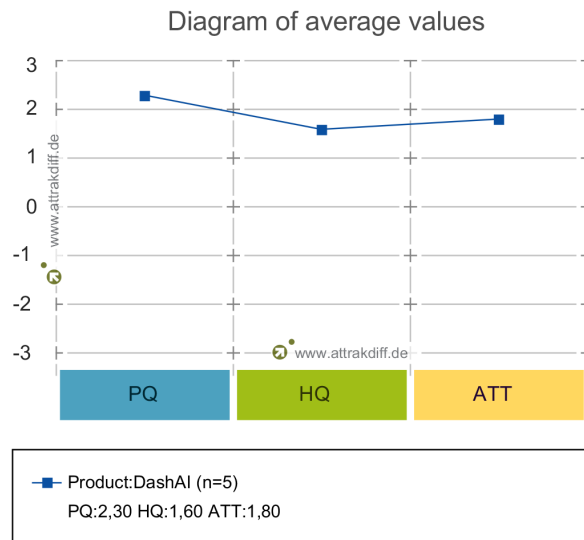


Imagen 2 - AttrakDiff Pragmatic vs Hedonic Quality

⁴ Gráficos generados en www.attrakdiff.de luego de realizar los 5 test.

Posteriormente revisamos los puntajes promedios de cada categoría (imagen 3), incluyendo en este caso la atractividad del producto (ATT) en estudio, donde si bien todos gozan de calificación positiva, la categoría con más baja calificación es HQ con 1,6 de un máximo de 3, seguido de ATT con un 1,8; dejando con lo más alto a PQ, es decir confirmando la utilidad de la aplicación.



Revisando las categorías más en detalle (imagen 4), notamos que la calificación de HQ es influenciada por una visión *"barata"* y *"de bajo gusto"* que le provoca al usuario el producto en cuestión; esto es complementado por la visión para la atractividad del producto es influida por a la idea de *"fea"*, cabe mencionar que estos son los puntos más débiles para mejorar, aún así, sigue siendo positiva la respuesta general del usuario frente a la aplicación, teniendo un factor positivo para todas las combinaciones de palabras, destacando por ejemplo las cualidades de simple, predecible (funcionamiento), buena y estructurada.

Imagen 3
 Attrak Diff: promedio por categorías

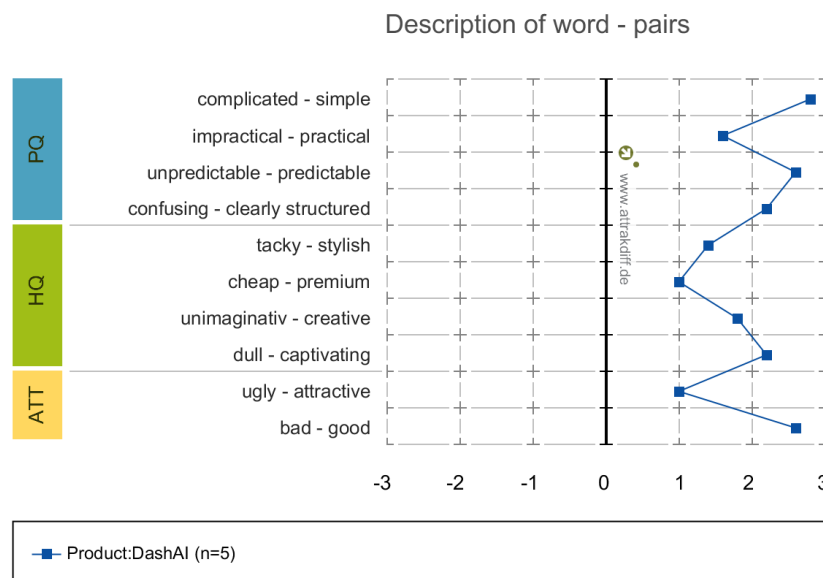


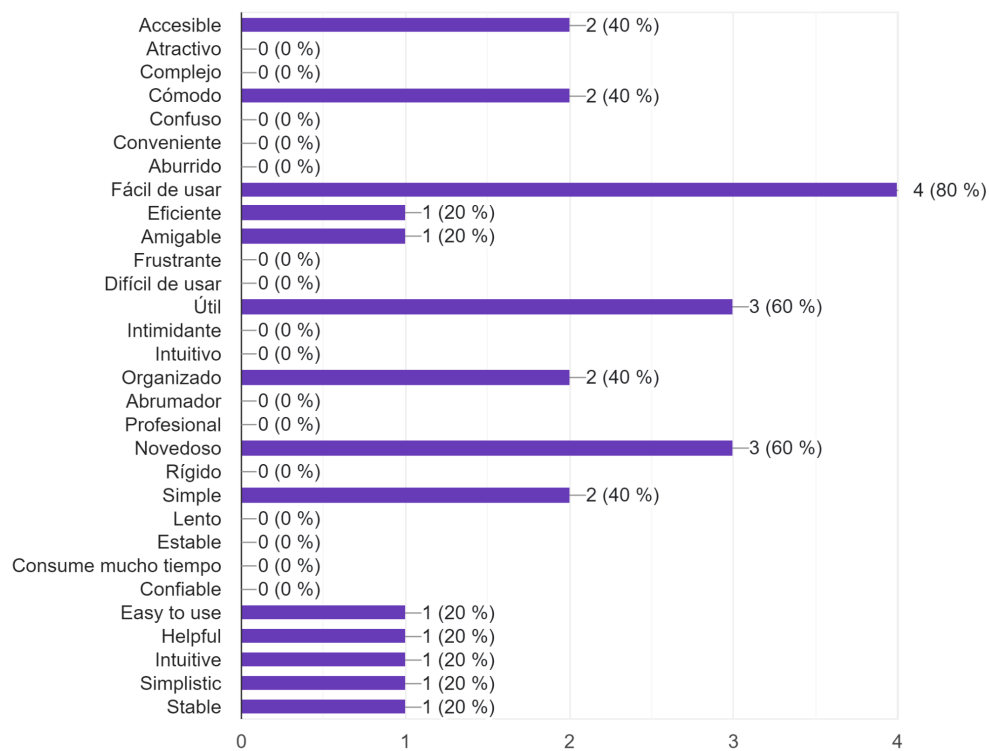
Imagen 4 - Attrak Diff: detalle cualidades por categoría

2. Test de deseabilidad

En la imagen inferior se puede observar los resultados de las distintas votaciones de los usuarios. Contando la versión en inglés del piloto, se puede observar, la característica más destacable del sistema fue “Fácil de usar” con 5 votos en total. Luego le sigue “útil” con 4 votos, “novedoso” y “simple”, ambos con 3 votos.

Lista de palabras.

5 respuestas



b. Hallazgos

Recuperando lo encontrado en las evaluaciones anteriores, procedemos a ligarlo a las notas tomadas durante las sesiones, las cuales agrupamos generando el siguiente diagrama de afinidad:

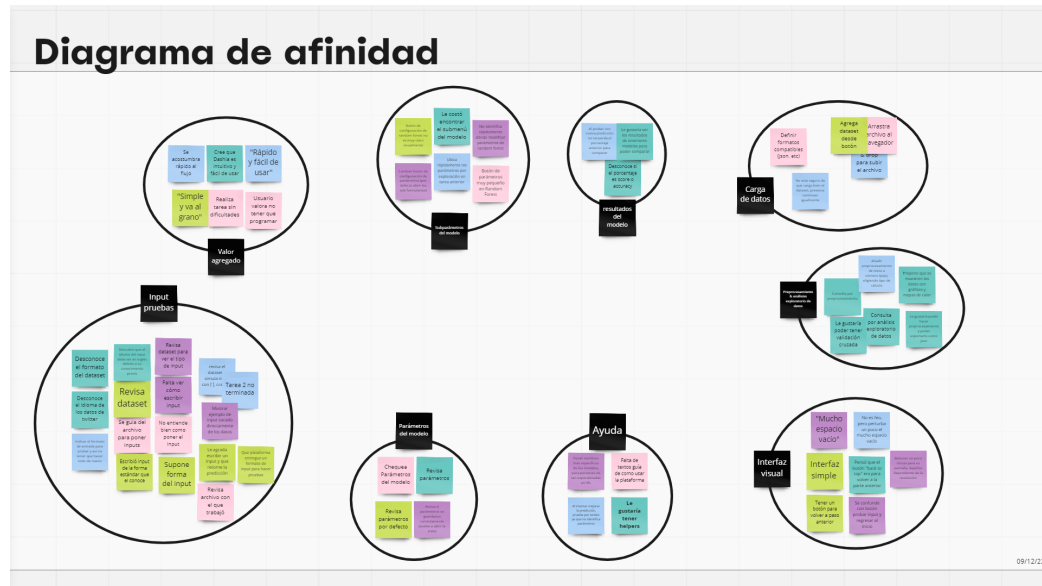


Imagen 6 - Diagrama de afinidad con notas de color vinculadas a cada usuario (5)

Con ello se observó que el usuario identifica claramente el fin y le genera una buena impresión, valorando por ejemplo el no tener que programar, junto a ello se encontraron diversos clusters, algunos importantes a nombrar que pueden ser de utilidad en una siguiente iteración futura son por ejemplo:

- **Interfaz visual:** donde se manifestaron detalles sobre botones pequeños, la gran cantidad de espacio vacío en los costados, la deficiencia de botones para volver un paso hacia atrás e inclusive error al hacer click para probar el input, provocando que se volviera al inicio.
- **Parámetros y sub-parámetros:** donde a falta de un indicador de guardado volvían a abrir el panel para revisar los inputs y también en un modelo que contiene otro modelo, los parámetros no fueron claros de encontrar (botón "O") provocando una pequeña confusión.
- **Pre Procesamiento y análisis exploratorio de datos:** esta funcionalidad fue consultada por los usuarios debido a su ausencia y el alto tiempo requerido, siendo un aspecto para algunos tedioso.

A partir de esto último y revisando las sesiones, se encontró lo planteado en el grafo PQ vs HQ del [Attrak Diff](#), que es que la aplicación se encuentra **enfocada** a una tarea, en este caso, la **generación del modelo (parámetros)**; dejando **afuera** otras complementarias del ciclo general de un modelo ML, por ejemplo la **exploración de los datos o la limpieza** de estos antes de entrenar el modelo, como así también la **baja calidad del input de pruebas** que fue el cluster con más cantidad de observaciones, mejoras y comentarios; ya que inclusive un **usuario no pudo terminar la tarea 2:** "Colocar un input para observar la predicción. Puede revisar el dataset si lo encuentra necesario", debido a que supuso un input con formato incorrecto⁵ lo que provocó que DashAI dejara de responder teniendo que refrescar la página, perdiendo lo realizado anteriormente. Con todo lo nombrado en el párrafo, es inclusive posible una **fuga de usuarios objetivo tras la pérdida de confianza o la deficiencia en el procesamiento/exploración** previa

⁵ El input en cuestión tenía el formato "[4.0, 2.2, 3.8, 0.1]" ; donde el que funciona correctamente es sin "[]"

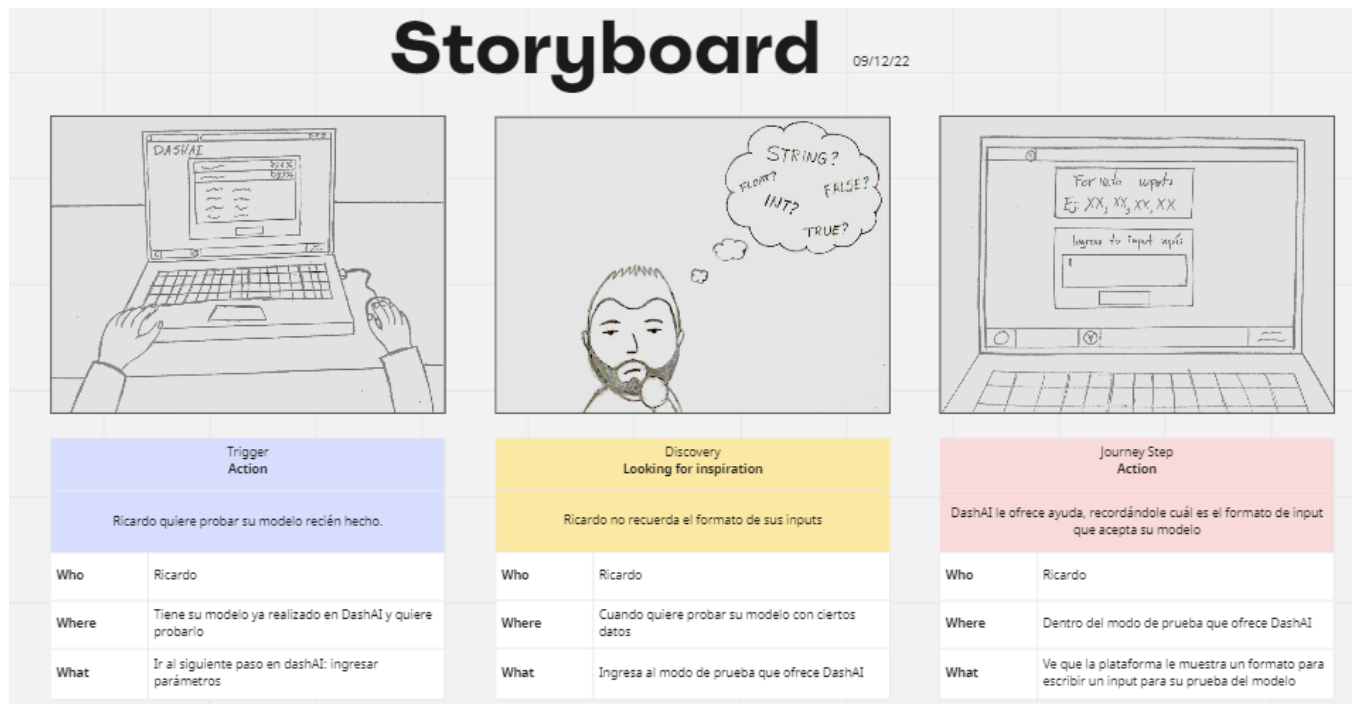
requerida para uso completo de ML, fiando más del proceso manual que conlleva mayor tiempo.

Es por la razón anterior que se decidió **trabajar con el asunto de los inputs de prueba**, ya que se encontró importante de arreglar, junto a que sería lo más adecuado de implementar en corto plazo trayendo consigo una mejora considerable en la experiencia y confianza del usuario. A continuación se procede a contextualizar en detalle este problema.

c. Análisis específico y mejora propuesta

i. Storyboard

Como se nombró anteriormente, se decidió trabajar con el asunto de input de prueba. Por tanto, el flujo del storyboard se basa en la situación dónde el usuario ya tiene su modelo hecho pero quiere realizar pruebas de predicción, lo que dió como resultado lo siguiente:



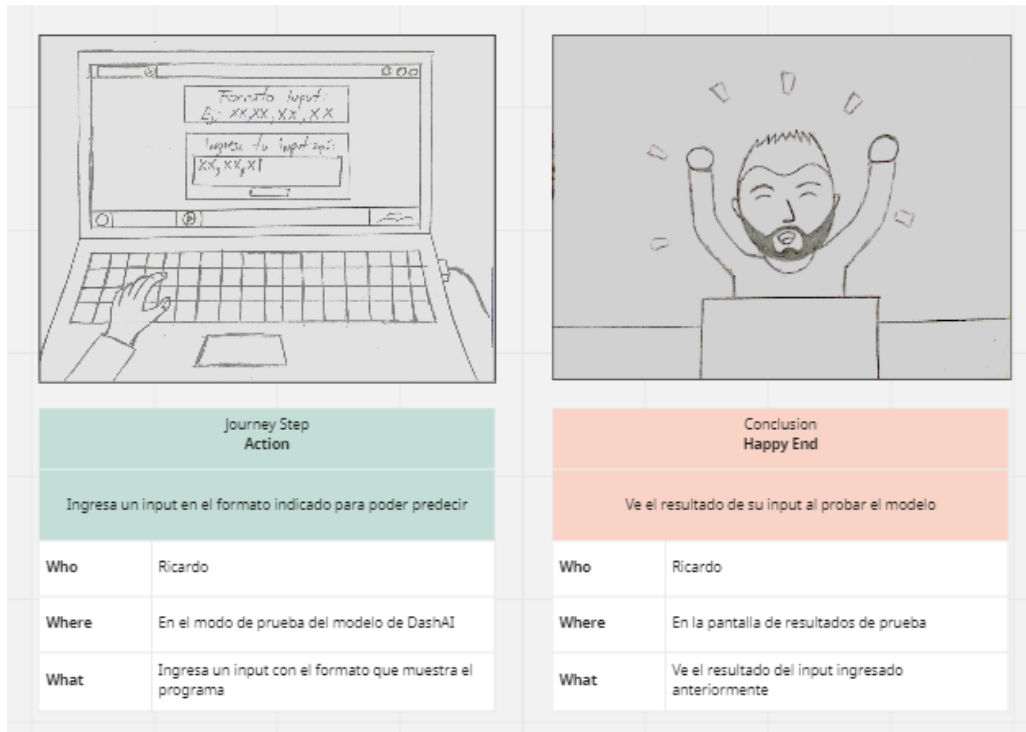


Imagen 7 - Storyboard de “Ingreso input de pruebas” realizado en Miro

ii. **Scenario mapping**

Basándonos en el problema planteado, surgieron las 4 etapas del flujo de trabajo que se pueden ver en la imagen inferior. De estas se plantearon en equipo ideas, comentarios y preguntas para comprender mejor el contexto. Luego se formaron clusters con los ítems que estuvieran más relacionados entre sí, dónde se destacan “*Problemas/mejoras con formulario*” en el paso de revisar el formato del input, “*permitir varios inputs*” y “*helpers para escribir input*” en el paso 3 y “*revisión de resultados*” y “*guardar resultados*” en el último paso. Consideramos estos clusters los más relevantes para entregar una solución al problema, por lo tanto nos basamos en estos para realizar los siguientes métodos.

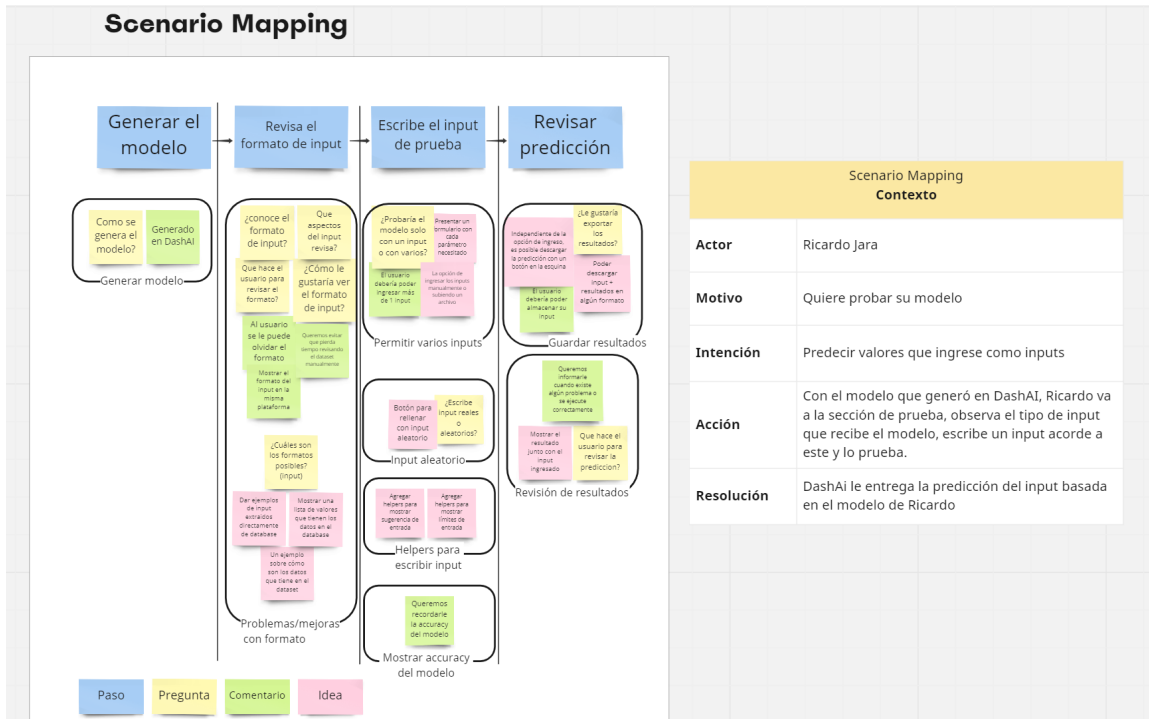


Imagen 8 - Mapa de escenario y su contexto, realizado en Miro

iii. Schematic diagram

Debido a lo anterior, el schematic diagram resultó de la siguiente manera, dónde se puede observar que el comienzo del flujo es desde la pantalla del resultado del modelo realizado y el final es cuando se descargan los resultados de la predicción al ingresar un input. Se hace la diferencia entre ingresar un input manual y uno a través de un archivo adjunto. Se omitió el flujo de regresar a pantallas anteriores y, en este caso, el usuario siempre querrá descargar un archivo con sus resultados.

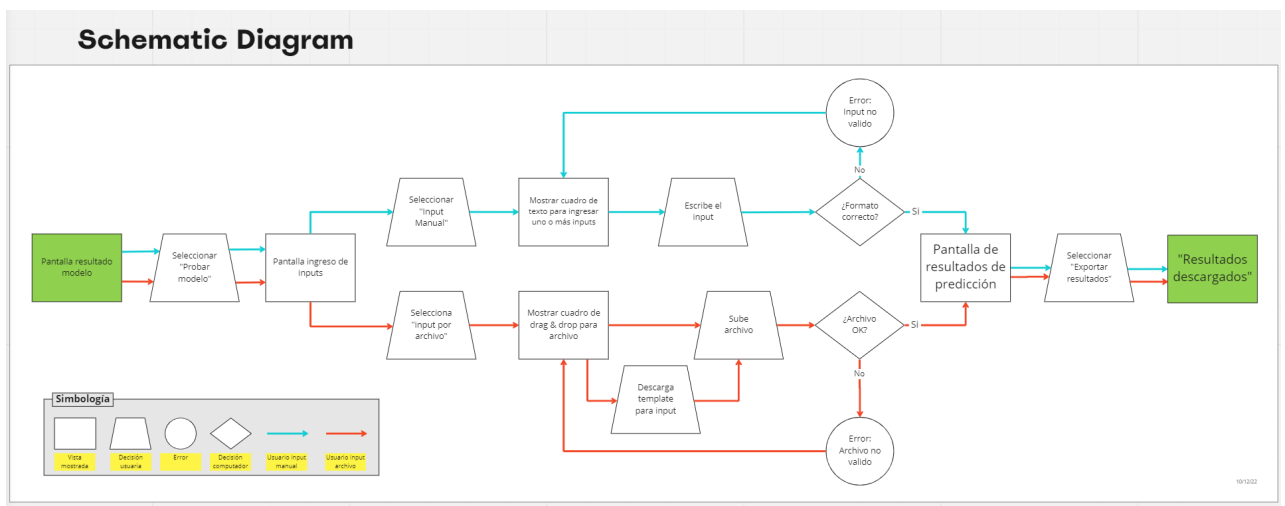


Imagen 9 -Diagrama esquemático del flujo propuesto, realizado en Miro

iv. Prototipo

Como resultado de todo el trabajo anterior realizado, se plantea el siguiente prototipo como solución al problema identificado. Se puede ver que se integra una sección que indica en cuál paso del proceso se encuentra el usuario, se agrega una sección para que el usuario seleccione el tipo de input que quiere ingresar y las respectivas pantallas de cada caso al escogerlos. Además, se muestra una pantalla con los resultados de la predicción, donde se le da al usuario la opción de descargar los resultados, tal como se identificó en el scenario mapping.



Imagen 10 - Capturas de interfaz desarrollada en Figma

Enlace a prototipo funcional en Figma [[Anexo vii](#)]

7. Anexo

i. “Investigación sobre el proceso de trabajo con Machine Learning” - Diciembre 2022 | F.Aspe, R.Cayazaya, V.Morales, J.Torres.

https://drive.google.com/file/d/1sVuWEajlnklpYi4dN2KUutYAxqen1U97/view?usp=share_link

ii. OpenCENIA/DashAI (github.com) - Versión analizada, commit 2f3b2c0

<https://github.com/OpenCENIA/DashAI/tree/2f3b2c0c9edf77a2f89c15f78a36a26564bd6ca1>

iii. Enlace a sesión piloto de Test de usuario (con permiso del usuario)

https://drive.google.com/file/d/1TbC683QnTQ9MGv58cK3gvJAI0ckSrhs7/view?usp=share_link

iv Enlace a notas de test de usuarios

https://docs.google.com/spreadsheets/d/1QlIHgJnp-szPpJ178R-v1KKac_648-rAkF1aL4AY7_c/edit?usp=share_link

v. Versión corta de AttrakDiff utilizada (short version)

<https://esurvey.uid.com/survey/#1ca122a4-2100-4ba4-b34b-76fe1369f6f6>

vi. Miro Board con desarrollo de diagramas y otros.

https://miro.com/app/board/uXjVP8XkDgc=/?share_link_id=465850168044

vii. Prototipo realizado en figma, Flujo interactivo (requiere cuenta en Figma)

<https://www.figma.com/proto/CaHpvXYfh49vI4ndRzxzc5/Proyecto-3?scaling=contain&starting-point-node-id=3%3A2>

vii. Carpeta con grabaciones de sesiones de test de usuario

https://drive.google.com/drive/folders/13po0mt3s0UFDLsQ9X8Ph2fASqOtnaHoV?usp=share_link

Tabla de tiempos

Tiempo [hrs]	Francisco Aspe	Rodrigo Cayazaya	Víctor Morales	Javier Torres	Total
Problem Statement y User persona (informe anterior)	0	0	0	0	0
Elaboración, mejora y ejecución de los Test de usuario					
Diagrama de afinidad	1,5	1,5	1,5	1,5	6
Storyboard	1	1	1+?	1	4
Scenario Mapping	0,75	0,75	0,75	0,75	3
schematic diagram	0,75	0,75	0,75	0,75	3
Prototipo Figma					
Total	10.5	11	10.5	10.5	42.5