Apellido y Nombre: ..... Legajo: .....

Arquitectura (40)			Modelo de Dominio (30)				Persistencia (30)			Final
A (25)	B (5)	C (10)	A (10)	B (10)	C (5)	C (5)	A (15)	B (10)	C (5)	

¡Importante!

- Se aprueba con 60 puntos y al menos un 50% de cada punto
- Entregue cada punto (Arquitectura, Dominio y Persistencia) en hojas separadas, escriba en una sola carilla, ponga apellido y numere cada hoja
- LEA TODO EL PARCIAL ANTES DE ARRANCAR

Elggak

Contexto

Elggak es una organización que provee servicios para la enseñanza, difusión y competencias de análisis de datos y aprendizaje automático (o su término más cool, machine learning). El núcleo de todas sus funciones son las competencias. Las mismas consisten en desafíos, donde se presenta un conjunto de datos, sobre el cual se propone predecir una determinada salida y el que ofrezca un modelo que calcule la “mejor” predicción, es quien gana la competencia.

Conjunto de datos o datasets

Los mismos consisten en una tabla de varias columnas / variables, cuyos valores pueden ser texto, numérico o imágenes y pueden ser continuos o categóricos (acotado a una serie de opciones). Hay 2 tipos de variables, las de entrada y las de salida. Las de entrada son los input del modelo y las de salida son el resultado esperado/conocido para ese caso. Existen 2 sets de datos distintos: el set de entrenamiento, donde las variables de entrada y de salida son visibles por el competidor y los de sets de datos de test, en los cuales los valores de salida no son visibles para los competidores

Evaluación de los modelos

El modelo se evalúa viendo las diferencias entre los valores predichos por el modelo del usuario sobre el set de datos de testing y los valores de salida. Usando estas diferencias se calcula un puntaje, y hay varias formas de calcularlo, por ejemplo la suma de residuos al cuadrado (RSS) para valores numéricos, el promedio de la distancia de Jaccard para etiquetados (PDJ), etc. Con estos puntajes, que dependiendo del método un valor bajo puede ser mejor que un valor alto (como en RSS) y viceversa en otros (PDJ se define entre 0 y 1, siendo 1 el mejor valor posible).

Competencias

Una competencia tiene un nombre, descripción, set de datos de entrenamiento, uno de test,una fecha de inicio, una fecha de cierre de inscripciones y una fecha de finalización. También puede o no ofrecer premios, por ejemplo monetarios, pero la administración de eso queda por fuera del sistema de software. A la competencia, pueden anotarse todos los usuarios del sistema que quieran, que pasan a ser competidores de la misma. Un usuario puede estar anotado en todas las competiciones que quiera al mismo tiempo. Hay 2 tipos de entregables en la competición (a) las que proveen un entorno de trabajo en la nube, donde el competidor tiene acceso a una IDE online, con todo el entorno (dependencias, librerías, versiones) resuelto y se proveen los recursos computacionales para que el mismo pruebe su solución (b) el competidor sube los resultados

obtenidos para cada fila del set de datos de testeo y un informe donde documente el modelo y la implementación utilizada.

Competidor

Una vez anotado, el competidor debe subir los resultados para evaluar sobre el set de test o indicar que la notebook sobre la cual trabajó ya tiene una nueva versión del modelo. Esto puede hacerlo cuantas veces quiera hasta que se termine la competencia. Es necesario poder consultar un histórico de resultados. A medida que se evalúan los resultados, se va armando una tabla de posiciones ordenada por el puntaje, pública para todos. Cuando se cierra la competición, y fueron evaluados todos los modelos, el orden en la lista determina a el / los ganadores.

Ejemplos

Competencia 1: Predecir altura de una persona en función de la altura de los padres

- Ambiente: Provisto por la plataforma. Modo de evaluación RSS.
- Dataset de entrenamiento: las primeras 3 columnas son variables de entrada y la última es la de salida o valor a predecir

N	Altura del padre	Altura de la madre	Sexo del Hijo	Altura del Hijo
1	164	155	M	165
2	171	165	F	160
3	175	182	M	182
4	181	160	M	176
	….			

- Dataset de test / evaluación: La altura del hijo **no** es conocida por los competidores

N	Altura del padre	Altura de la madre	Sexo del Hijo	Altura del Hijo
1	167	165	M	166
2	190	170	F	172
…	…	…	…	…

Competencia 2: Determinar si una imagen es meme de EAMEO. Ambiente NO Provisto por la plataforma

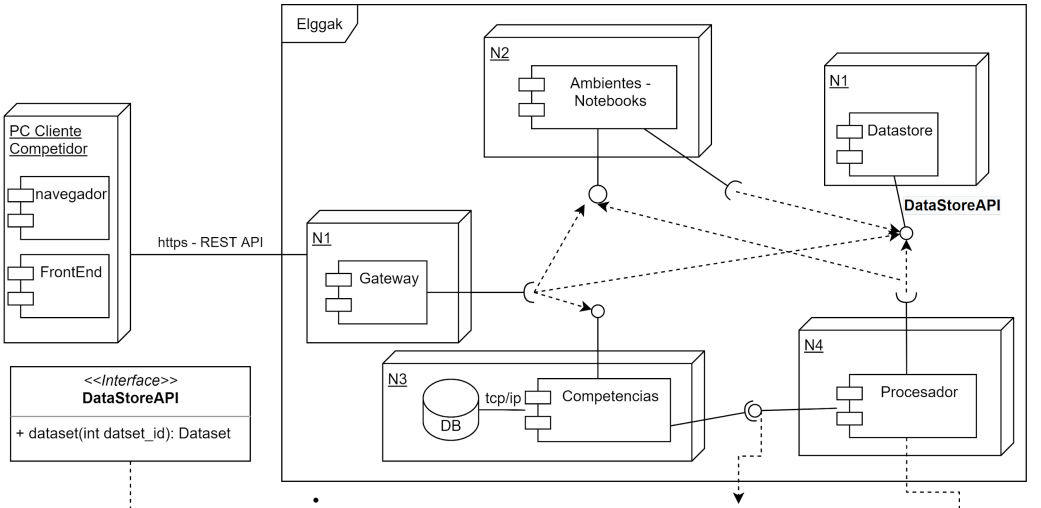
N	Imagen	fecha	cantidad de comentarios	Es de EAMEO (salida)
1	imagen1	10-04-2020	565	SI
2	imagen2	11-05-2022	903	NO
…	…	…	…	…

Competencia 3: Etiquetar aves en un video. Ambiente NO Provisto por la plataforma. Evaluación: Promedio de distancia de Jaccard.

N	Imagen	fecha	Hora	lugar	aves (salida)
1	video1	10-04-2020	3pm	Aeropuerto	chimango, aguilucho
2	video2	10-04-2020	4am	Plaza San Martin	loro, pájaro loco
…	…	…	…		…

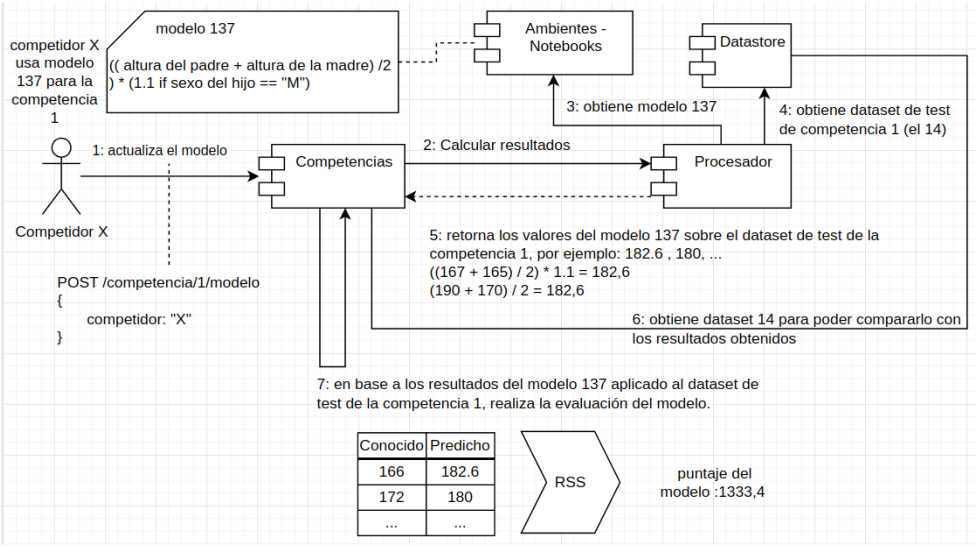
Arquitectura

En el back, la arquitectura tiene 5 componentes. El **Gateway** recibe los pedidos del front y los dirige al servicio adecuado. El de **Ambientes** provee el entorno de trabajo / IDE a los competidores, donde se almacena la implementación de sus modelos. **Datastore** tiene los datasets con los que se trabaja, los mismos se almacenan solo ahí. **Procesador** toma la implementación de un modelo y lo aplica a un dataset, el mismo NO tiene estado ni base de datos, es decir, recibe un pedido y retorna las predicciones. Finalmente **Competencias**, nuestro componente, posee los datos de las competencias, los competidores y realiza la evaluación de los modelos o resultados que envían los usuarios.

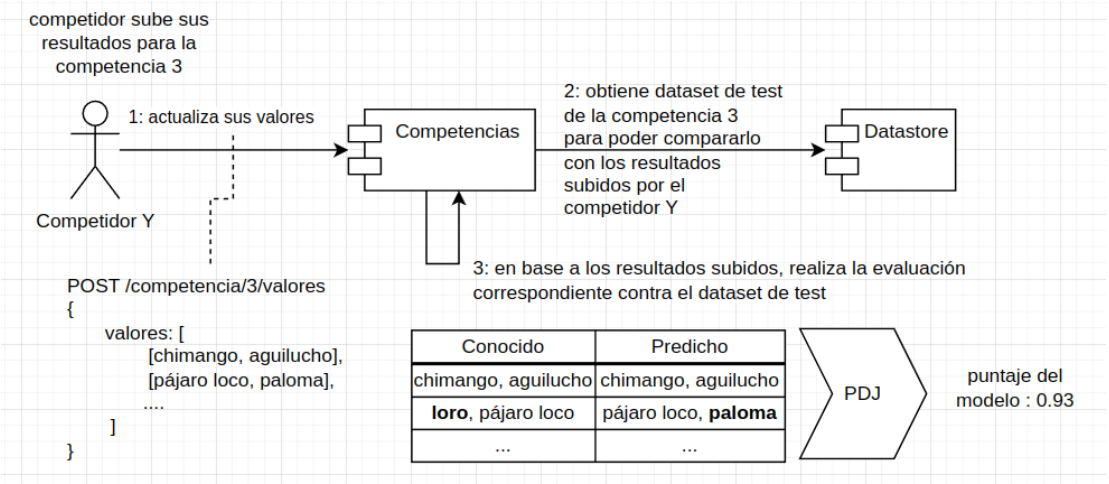


Usted cuenta con estas clases para la evaluación de modelos, tenga en cuenta que a futuro van a sumarse más.

Caso 1: Competidor X actualiza su modelos en Competencia 1



Caso 2: Competidor Y actualiza sus resultados en la Competencia 3



Arquitectura (40)

1. Se reporta que cuando varios competidores suben una nueva implementación de sus modelos al mismo tiempo, se generan grandes demoras en la respuesta y varios procesos dentro del componente Procesador se terminan abruptamente por falta de recursos.
    a. Modifique la arquitectura para evitar estas demoras (haga el diagrama), explique las modificaciones (10 ptos)
    b. De un ejemplo de cómo funciona la subida del nuevo modelo, use el diagrama anterior para explicar (10 ptos)
    c. A nivel interfaz de usuario, ¿qué cuidado debería tener si el proceso de carga es largo ? (varios minutos) (5 ptos)
2. ¿A qué propiedad del sistema ayuda que el componente procesador no tenga estado? (5 ptos)
3. Un colega suyo propone el uso de una caché en el componente Procesador para el endpoint expuesto. ¿Es una decisión adecuada? Justifique. De ejemplos (10 puntos)

Dominio (30)

Diseñar el componente "**Competencias**" que resuelva los requerimientos aplicando el paradigma OO

1. Armar la especificación del modelo utilizando diagramas UML (diagrama de clases obligatorio) (10)
2. Justificar las decisiones de diseño que se tomen, por ejemplo, haciendo referencia a los principios que guían al diseño o las consecuencias de aplicar un determinado patrón. (10 puntos)
3. Explique con un diagrama de secuencia, pseudocódigo o prosa el caso de uso: el competidor Y sube nuevos resultados para la competencia 3. (5 puntos)
4. Explique con un diagrama de secuencia, pseudocódigo o prosa el caso de uso: el competidor X dice que su notebook tiene un nuevo modelo para la competencia 1. (5 puntos)

Persistencia (30)

Diseñar el modelo de datos del componente "**Competencias**" para poder persistir en una base de datos relacional a través de un ORM.

1. Armar la especificación usando un DER físico. Indicando las entidades, sus campos, claves primarias, las foráneas, cardinalidad, modalidad y las restricciones según corresponda. (15)
2. Justificar: (10)
    a. Qué elementos del modelo es necesario persistir y cuáles no
    b. Cómo resolvió los impedance mismatches.
    c. Las estructuras de datos que deban ser desnormalizadas, si corresponde.
3. Explicar cómo se almacenan los datos para el caso de uso: "Generar Tabla de posiciones para la competencia X" (5 puntos)