Presentación

-PowerPoint

-¿Enseñar código del modelo?

-Web

-Portada: título autor, tutores, grado

-índice breadcrumbs

-introducción: contar el problema 2 MIN

-objetivos1 MIN

-brevemente: ASPECTOS TEORICOS 1MIN MEDIO

-métodos herramientas: TENCOLOGIA SIN ENTRAR EN PROFUNDIDAD. DESGING SCIENCE, SCRUM. CONTEXTO METODOLOGICO. OPEN NN, NODEJS. 2 DIAPOS. SIN TEXTO 2MIN

-Aspectos relevantes: metodología-diseño-comunicación-arquitectura completa ¾ min

-Demo de la web ¾ min

-Resultados tiempo junto a lo de arriba

-conclusiones

**- Introducción/contexto:** situación de la contaminación, contaminantes y límites en Madrid, índice de calidad del aire.

**- Objetivos:** para solucionar lo mencionado anteriormente, se va a hacer tal cosa, con los objetivos finales tal y tal.

**- Conceptos teóricos básicos de redes neuronales:** lo necesario para que después se entienda la parte de desarrollo del modelo.

**-¿Metodología aquí?** DS, Scrum.

**-Desarrollo del proyecto:**

**-Datos históricos:** explicar lo que se ha recogido, con qué función y cómo

**-Diseño del modelo:**

-Conjunto de datos: enseñar correlaciones (ya que es algo muy visual y puede ser interesante).

-Arquitectura red neuronal: capas, por qué se eligen esas.

-Algoritmos: ventajas y desventajas del elegido, por qué este.

-Función de coste: Minkowski para minimizar picos.

-Resultados: tablas de validación, valores de errores. Mucho énfasis en los buenos resultados obtenidos.

**-Diseño interfaz web:**

-Directrices: colores, profesionalidad, etc.

-Aspectos técnicos: NodeJS, recogida datos tiempo real APIs,, integración del modelo. ENSEÑAR AQUÍ LA INTERFAZ WEB.

**-Arquitectura final:** diagrama de componentes.

**-Conclusiones:** lo aprendido a lo largo del trabajo, partes más complicadas, etc.

**-Mejoras:** oportunidades para continuar el trabajo, aspectos a mejorar, etc.

Guion

1. El proyecto se centra en la contaminación atmosférica. Es uno de los grandes problemas a los que se enfrenta todo el mundo, pero en concreto las grandes ciudades como por ejemplo Madrid. Debido a que afecta tanto a las personas como al medio ambiente, reducir la contaminación debe ser una prioridad para las administraciones públicas.

En esta imagen vemos la gran cantidad de estaciones de medición de la contaminación atmosférica que hay en Madrid. Cada una ofrece datos públicos sobre los valores actuales que nos van a ser útiles en este proyecto.

1. La idea sobre la que se basa el trabajo es predecir los niveles de contaminación atmosférica que va a haber en los próximos días. Esta idea no es nueva y ya existen modelos que tratan de realizar esta tarea, y se dividen en dos tipos: mecanicistas y no mecanicistas. Los mecanicistas necesitan gran cantidad de datos y que estos sean precisos para dar buenos resultados, mientras que los no mecanicistas son más cómodos y prácticos porque no involucran la investigación de procesos complejos de los contaminantes. Un tipo de modelos no mecanicistas son los generados con técnicas de inteligencia artificial, como por ejemplo las redes neuronales artificiales que es lo que se usa en este trabajo.

En la imagen vemos un esquema básico de una red neuronal, con una capa de entrada y una de salida y varias intermedias que se llaman ocultas, siendo una capa una agrupación de neuronas.

1. Por lo tanto, estos son los objetivos del trabajo. El principal es conseguir una predicción los más fiable posible de los niveles de contaminación atmosférica en Madrid. Para ello será necesario obtener un buen modelo a partir de una red neuronal que nos de los resultados.

Como objetivos secundarios tenemos, por un lado, construir una manera de presentar esta predicción a la gente que le pueda interesar, que pueden ser personas que no tengan conocimientos de inteligencia artificial ni de modelos de predicción por lo que tiene que centrarse en los resultados.

Y por último tenemos el objetivo de conseguir el despliegue y el entrenamiento continuo del modelo. El concepto de despliegue continuo quiere decir que se busca obtener la predicción, por ejemplo, siempre para el día o la semana siguientes, no solamente predecir unos valores para un momento concreto y quedarnos ahí. Y por otro lado el entrenamiento continuo se basa en añadir los datos que se van recabando día a día al entrenamiento de la red neuronal, que es un proceso que se va a explicar posteriormente. Añadir los datos de un día no va a variar mucho los resultados, pero cuando esto se hace a lo largo de un periodo de tiempo más largo pues mejora los resultados.

1. En este apartado voy a introducir algunos conceptos teóricos de los que se va a hablar más tarde en los aspectos relevantes. En primer lugar estos son los contaminantes cuyos valores se van a predecir. En cuanto al tipo de contaminantes si es primario quiere decir que se emite directamente por el tráfico, las fábricas o lo que sea y si es secundario quiere decir que se forma por reaccines químicas entre los contaminantes primarios. El límite es el marcado por la Comunidad de Madrid.

PM2.5 y PM10 son los dos tipos de materia particulada. Las párticulas son el contaminante mas peligroso y cuanto más pequeñas sean peor porque se pueden introducir incluso en el torriente sanguíneo. Por ejemplo 2.5 sangre y 10 solo pulmones. El otro que se suele marcar como más peligroso es el ozono troposférico (ozono a nivel de suelo, no tiene nada que ver con el atmofesrico de la capa de ozono). Y por utlimo están el dióxido de nitrógeno y el de azufre.

1. La predicción se va a hacer sobre el índice de calidad del aire en vez de concentraciones absolutas. Es un índice que aunque no es estándar en el mundo entero lo utilizan una gran cantidad de países. En concreto la comunidad de Madrid se basa en el estadounidense. Transforme las concentraciones absolutas en microgramos por metro cuadrado (que es la unidad en la que hemos visto los limites en la diapositiva anterior) a un valor entre 0 y 500, cuanto más peor ya que significa más contaminación. Esto es más cómodo de entender para el publico general ya que no tiene que conocer cuales son los valores normales o anormales para un contaminante.
2. Bueno para introducir el concepto de red neuronal brevemente pues a partir de unos datos de entrada, que en nuestro caso va a ser una serie temporal de datos de los contaminantes y meteorológicos, se construyen una red neuronal con características como el algoritmo de entrenamiento o la función de coste (que se explicaran a fondo después) y a partir de ahí se obtiene el modelo.
3. Pasamos al apartado de métodos y herramientas que se han usado en el trabajo y empezamos por las metodologías. Como es un proyecto diferente a lo que podría ser un diseño de software tradicional, se ha usado una metodología enfocada al mundo científico que se llama Design Science Research Methodology. Aporta las pautas y los pasos que debe tener un proyecto de investigación científico en el que se desarrolla software. Como vemos en la imagen sigue un proceso por etapas sobre las que se va iterando y realizando una serie de tareas.

Esta metodología se acompaña con Scrum para aspectos del desarrollo de software como la planificación temporal o la estimación del esfuerzo.

1. Pasamos a las herramientas que se han utilizado. Para crear la red neuronal hay bastnates librerías que permiten crear redes neuronales, quizá la mas famosa es TensorFlow que es una librería de Python. En este caso el lenguaje de programación ha sido C++ junto a la librería open source OpenNN. Contiene gran cantidad de características como algoritmos de entrenamiento, de índices de error o tipos de capas que se explicaran proximante y además el entrenamiento de la red es bastante rápido por lo que se pueden realizar muchas pruebas diferentes.
2. Para la comunicación de los resultados se va a crear una interfaz web donde a través de gráficas que son elementos muy intuitivos se muestre la información obtenida. Para la capa de servidor se utiliza Node JS junto a Express para la parte de la vista. Estas herramientas tienen gran cantidad de librerías que facilitan diferentes tareas como la recogida de datos o la creación de las propias graficas. Todo el código de la interfaz web esta escrito sobre el trio de html css y javascript.
3. Entramos en el apartado de aspectos relevantes, donde se van a exponer las partes más importantes del desarrollo del trabajo. Empezamos con la metodología. Estas son las fases que nos indica Design Science. En este trabajo, la identificación del problema es de lo que hemos hablado anteriormente de conocer con antelación una predicción de la contaminación, los objetivos también los he mencionado, el diseño y desarrollo va a ser la creación del modelo a partir de las pruebas y la obtención de un modelo, la demonstración es usar este modelo con los datos de Madrid y ver que obtenemos unos resultados con sentido (da igual que sean mas o menos correctos todavía), la evaluación se basa en ver si son buenos estos resultados y si no lo son volver atrás para intentar mejorarlos y por último la comunicación va a ser el desarrollo de la interfaz web.
4. Ahora voy a explicar cómo ha sido el diseño del modelo y las fases del mismo. En primer lugar ha sido necesario obtener un conjunto de datos históricos que son los que vemos en la imagen. Hay datos temporales como puede ser el día de la semana, valor medio del contaminante ese día e información meteorológica. Este conjunto de datos no se puede usar así como así para obtener la red neuronal sino que hay que transformarlo introduciendo un número de lags y otro de steps ahead. El número de lags significa la información previa que aparece en cada muestra. Una muestra es cada línea, por ejemplo si es 2 en esta línea aparecerá una columna con T\_AVG\_lag1 con el valor del día anterior. Realizando pruebas he comprobado que cambiar este número no mejoraba excesivamente los resultados por lo que va a ser 2 que es el tipico. Y por otro lado el de steps ahead son en este caso los días posteriores sobre los que se quiere una predicción, para este trabajo van a ser 7 para tener la predicción semanal.
5. Esto se hace para que el motor de inteligencia artificial pueda obtener correlaciones entre las variables y ajustar la red neuronal como corresponda. Por ejemplo, en esta imagen vemos las correlaciones para el contaminante PM2.5. Un valor positivo significa probabilidad directa, o sea si aumenta un valor aumenta también este. Vemos que la mayor dependencia se encuentra con los valores de los contamiantes para el día anterior, sobre todo del PM10 y de si mismo. Por otro lado un valor negativo quiere decir proporcionalidad inversa. En este caso cuanto mayor es la velocidad del viento menores son los valores de contaminación. El sentido de esto es la dispersión que provoca una mayor velocidad del viento, lo que provoca que la concentración en un punto concreto sea menor.
6. Vamos a pasar a la parte de la arquitectura de la red neuronal, donde se realizaron pruebas comparando los niveles de error según las diferentes características. Por una parte hay que elegir las capas que va a tener la red neuronal. Existen gran cantidad de tipos de capas, aunque para problemas de predicción los tipos más famosos son la capa de perceptrón y la capa LSTM que significa Long Short Term Memory, para otros problemas como los de clasificación esta la probabilística por jemeplo. La capa de perceptrón es muy simple comaprada con la LSTM y la tenemos en el esquema de la derecha. A la entrada se le aplica una función de combinación utilizando pesos y sesgos, que son los que van modificando la red neuronal a lo largo del proceso de entrenamiento, y luego una función de activa que devuelve una salida entre 0 y 1 a partir de la entrada. Esta función puede ser lineal pero la tradicional es la tangente hiperbólica.

Por otro lado tenemos el algoritmo de entrenamiento. El diagrama de actividad de estos algoritmos es para todos igual: se evalua la red neuronal y si no se ha llegado a un criterio de parada, como puede ser un error de entrenamiento mínimo o un tiempo máximo o similares, se mejoran los parámetros y se vuelve a entrenar. Existen gran cantidad de algoritmos de entrenamiento. En la imagen podemos ver cinco de estos. Se observa que los que mejor tiempo de entrenamiento ofrecen son los que peor uso de la memoria tienen por lo que hay que encontrar el balance para nuestro proyecto. Como el conjunto de datos que usamos, aun teniendo 2500 muestras, no se considera muy grande respecto a otros proyectos con redes neuronales, nos van a interesar uno rápido ya que no va a haber mucho gasto de memoria. Tras realizar pruebas y comparar el error finalmente se ha usado el método de Quasi-Newton, que matemáticamente lo que hace es encontrar o ceros o máximos y mínimos de una función.

Y finalmente tenemos la función de coste, que nos da la estimación sobre la calidad de los resultados obtenidos. Tiene dos términos: el de error y el de regularización. Utiliza un subconjunto de validación para indicar la diferencia entre las predicciones de la red neuronal y los resultados correctos.

En cuanto al término de error que es el más importante también existen diferentes tipos, siendo el más típico la suma de los errores al cuadrado. Tras las pruebas correspondientes para este trabajo finalmente se ha usado el error de Minkowski, cuya formula es la de la imagen de la derecha. La peculiaridad es que introduce un parámetro al que eleva el numerador, que mínimo será 1. Cuanto mas cerca de 1 menos importarán los picos que haya y cuanto mayor más peso tendrán en el resultado de error final. Como nosotros buscamos dar importancia a lo que se conoce como eventos singulares, se ha usado como parámetro 3 para darles importancia.

1. Esta es la arquitectura final de la red neuronal, que como no se ve mucho la voy a abrir en grande. A partir de las entradas que son una muestra del conjunto de datos se pasa a la capa de escalada (bolas amarillas). Es una capa que se usa en prácticamente todas las redes neuronales para que todas las entradas tengan un rango adecuado. Las normaliza entre -1 y 1 aunque hay mas tipos.

Luego esta la capa mas importante que es la primera de perceptrón que son las 10 bolas azules. La función de activación es tangente hiperbólica y aquí es donde tiene lugar el verdadero ajusta de la red. Se eligen los pesos y sesgos que van a definir el modelo. Aquí el numero de neuronas es importante ya que con pocas no se entrena bien y demasiadas es inútil y se entrena demasiado, por lo tanto hay que realizar pruebas con distinto numero de nueronas y ver cual se ajusta al conjunto de datos.

La segunda capa azul es la segunda capa de perceptrón que sirve para pasar de 10 neuronas a 35, que son las variables de salida. Simplemente tiene una función de activación líneal y no tiene cálculos como tal

Luego las bolas naranjas son la capa de desescalado que sirven para pasar los valores que antes se han escalado entre -1 y 1 a los valores normales ya con sentido, por ultimo las bolas moradas son una capa de filtrado para que la salida no se salga de unos límites. Para este trabajo como si que nos importan los valores anormales que son los eventos singulares pues esta capa no realiza ninguna función. Y ya entonces se obtienen las 35 salidas, una para cada contamiantes por día de la semana.

1. Ahora que ya he explicado como se ha obtenido el modelo voy a pasar a ver los aspectos de la interfaz web que luego se enseñará en la demostración. En primer lugar defini las características que debía tener, que son esas directrices. Queremos centrarnos en la información, sin añadir aspectos no tan útiles que puedan distraer al usuario. Por tanto se van a utilizar pocos colores, simplemente el azul y blanco para los fondos con el texto en negro, va a haber un pequeño apartado con la información actual al inicio, luego la predicción como tal y finalmente un poco de información por si se quiere saber algo mas. Ahora lo enseño en la parte de demostración.
2. Como aspectos técnicos la interfaz tenemos por un lado la capa de servidor y por otro la vista, para las que se utiliza nodejs y express respectivamente. Casi toda la funcionalidad se encuentra en el servidor siguiendo la filosofía de node js. Por un lado, hay que recoger los datos en tiempo real para meterlos al modelo y calcular las salidas, esto se hace a través de dos APIs de internet. La de la contaminación se llama AQICN y nos da los datos actuales de Madrid de los coinco contamiantes y la del tiempo se llama Visual Crossing y nos da tanto para días pasados ya que son necesarios para los lags de cada muestra como los actuales.

La integración del modelo que es un aspecto muy importante y que me llevo algo de tiempo hasta que funciono bien se hace a través de ejecutables. Se compila el código de C++, y el ejecutable que se obtiene lo utiliza la web para pasarle las entradas como argumento. Luego se leen las salidas obtenidas y se pasan a unos gráficas creads con la librería Charts js. Todas son graficas de barras tradicionales pero la barra se pone del color del AQI que corresponda, por ejemplo si es entre 0 y 50 se pone verde. Esto hace que sea mas intuitivo para el usuario. Por último tenemos el despliegue continuo y el entrenamiento continuo. El despliegue es el proceso que he explicado anteriormente de recoger los datos actuales y meterlos al ejectuable para obtener la predciion de la semana que viene y el entrenamiento continuo lo que hace es añadir al conjunto de datos, por lo que la web tiene que tener acceso al mismo, una muestra por día con los datos de ese día. De esta manera todos los días se va a ir añadiendo la información diaria, además hay otro ejecutable que cuando se añade esta informacion reentrena la red con el conjunto de datos modificado.

1. Esta es la arquitectura final del proyecto con un diagrama de componentes. Vemos que tenemos el servidor web, al que los clientes pueden conectarse mediante cualquier navegador web y HTTP. Este servidor tiene el modelo creado a partir del motor de inteligencia artificial y es el encargado de realizar los cálculos y todo el proceso de obtención de las salidas. Y por otro lado tenemos la conexión con las API a través de las que se obtiene un fichero JSON del que se extraen los datos necesarios.

- Product Backlog e historias de usuario

- Proceso iterativo e incremental, división en sprints

- Enfoque científico

- Comunicación de resultados a todo tipo de público

- Documentación mínima y necesaria