Anteproyecto de Trabajo de Fin de Grado

Autor: Diego Fernández Rueda

Tutora: Ana Castillo Martínez

Grado en Ingeniería Informática, Universidad de Alcalá

Resumen

Este trabajo pretende realizar una aplicación capaz de recuperar la información publicada en portales web sobre los resultados de un juego de azar (bono loto, primitiva, etc.) con técnicas de Web Scraping. Toda la información recuperada se almacenará en una base de datos con el fin de poder aplicar técnicas de análisis de datos para mostrar estadísticas y para poder realizar predicciones que puedan guiar las potenciales apuestas del usuario en función de los datos ganadores históricos. Se ofrecerán distintos criterios de ajuste de la predicción en función de los datos históricos y se mostrará su precisión de predicción mediante su aplicación retroactiva a los sorteos pasados.

Palabras clave

Web scraping; automatización; bases de datos; análisis de datos; predicción

Introducción

Este proyecto tiene como objetivo la recopilación, tratamiento y análisis de datos aplicado a dos juegos de azar de características similares en cuanto a reglas, desarrollo y funcionamiento, pero distinto en cuanto a volumen de información

ofrecida. Estos juegos de azar son el sorteo de la Primitiva y el sorteo de Bonoloto. Ambos comparten el mismo método de obtención de combinación ganadora y ambos, aparentemente, son completamente aleatorios.
Cada apuesta en ambos sorteos se hace seleccionando 6 números de una tabla de 49 números (del 1 al 49). Después el terminal que valida la apuesta asigna automáticamente un número, del cero al nueve, de forma independiente a los anteriores llamado reintegro. Durante el desarrollo del trabajo, nos centraremos en los seis primeros números elegidos por el usuario para hacer el análisis de resultados y nos desprendemos del llamado reintegro en los pasos de tratamiento. La razón es que el reintegro es un número no elegible por el jugador y, por tanto, la utilidad de su posible predicción es nula cuando no puede seleccionarse en la apuesta.
El sorteo de la lotería Primitiva tiene orígenes en el reinado de Carlos III, siendo en 1763 el primer sorteo con un sistema similar al actual. Esta serie de sorteos fueron suprimidos en el año 1862 y no fue hasta 1985 cuando se retomaron de nuevo. Es importante tener en cuenta que estos sorteos quedaron registrados, pero resultan inválidos para los datos que recolectemos, puesto que el método de obtención de los resultados para las combinaciones ganadoras dista mucho del actual. Es a partir de 1985 cuando la forma de conseguir las combinaciones se hace de una manera estandarizada y los registros quedan digitalizados, aptos para la extracción haciendo uso de web scraping. Este sorteo se celebra todas las semanas del año los jueves y desde 1990 también los sábados.
Por otro lado, el sorteo de Bonoloto comenzó en 1988 pero se celebra cada día de la semana. Este dato es importante a la hora de comparar los tipos de predicción entre ambos casos.
Teniendo en cuenta que ambos sorteos tienen la estructura para análisis de 6 bolas elegidas por el usuario, el dato que más cambia entre los dos es la cantidad de datos con los que contamos. Por parte de la primitiva contamos por el momento con 3583 registros de sorteos ganadores mientras que con la Bonoloto manejamos 7669 registros. El hecho de que sea una cantidad que dobla a la otra, hará que en las fases del proyecto futuras, los datos obtenidos por parte de las predicciones sean más certeras en el caso de la bonoloto. Es interesante ver cómo los análisis se verán afectados por el hecho de la disposición de una mayor y menor cantidad de datos con los que trabajar.
Con el análisis de estos datos podremos determinar si, a pesar de tratarse de tareas sin patrón de comportamiento teóricamente, cumplen con alguno y puede ser determinante en el resultado ganador de un futuro sorteo.

El web scraping consiste en la extracción de datos de un sitio web. Esta información la recopilamos y exportamos a un formato que sea más útil para el tratamiento de estos. En nuestro caso, una base de datos PostgreSQL.

Aunque el web scraping se puede hacer manualmente, lógicamente una herramienta automatizada es totalmente esencial cuando se extraen datos de sitios

web. Para este proyecto, se va a tener que tratar con más de cien mil datos originales y más de veinte mil generados a partir de estos en la fase de tratamiento y análisis. La cantidad de datos, así como la distribución de estos en la web, hace imposible su extracción manual, que podría implicar a un tiempo equivalente a 100 veces el necesario de forma automática como se ha optado a hacer. Como sitio web de referencia para recopilar datos se ha seleccionado https://lawebdelaprimitiva.com; esta web tiene dos ventajas principales que la ha hecho resaltar sobre otras web con datos históricos de los sorteos que analizaremos. La primera es la sencillez (relativa) del diseño web. La extracción de los datos es más fácil y ofrece menos problemas a la hora de desarrollarse si el diseño de la página no tiene muchos elementos css que entorpezcan el proceso. La segunda es la disposición de los datos. En la web, los datos se muestran en listas separadas por los años de cada sorteo, en caso de producirse un nuevo sorteo, este registro queda grabado en la lista, haciendo que no haya que extraer de otros lugares en la web el valor buscado.

Selenium (*Selenium*, n.d.) es un framework específico para aplicaciones web que facilita la automatización de pruebas funcionales. Hemos elegido Selenium por la facilidad que ofrece para controlar remotamente las instancias de los navegadores y emular las interacciones de los usuarios. Permite simular las tareas que son habituales en un usuario como completar campos de texto, seleccionar opciones de los desplegables o hacer click en determinados enlaces. Una de las funcionalidades más interesantes que se usarán en el desarrollo del trabajo es el Scroll Down de la página (o Scroll Up, según el caso) para poder constatar que el texto se obtiene de una porción muy concreta de la página web. Será necesario también el uso de Click y selección de opciones en los menús desplegables en la web junto con el scroll, lo que supone una carga extra de tiempo y recursos. El sistema deberá facilitar la actualización de los datos de sorteos a medida que se vayan celebrando para tener siempre actualizada la colección de datos históricos.

Los datos extraídos en el sitio web elegido no se encuentran estructurados de una manera sencilla de obtener. Por eso, se opta por la extracción mediante el uso de hilos. Un hilo es una tarea que se ejecuta al mismo tiempo que otra tarea. En nuestro caso, los hilos que se usan comparten los mismos recursos al acceder conjuntamente a la misma base de datos. El hecho de implementar hilos en la fase de extracción de datos y la creación de nuevas tablas de análisis para su tratamiento supone una ventaja sustancial en tiempo, tanto al ejecutar el programa como al crearlo, para comprobar los errores de compilación y ejecución. Los hilos se implementarán en Python (*Python*, n.d.), de manera que puedan reducir a casi la mitad de tiempo el necesario para la creación de los registros en la base de datos.

El sistema explotará posteriormente los datos almacenados en la base de datos para analizarlos, extraer estadísticas, permitir usar distintos criterios de elaboración

de predicciones y comprobar su precisión si se hubiera aplicado a los datos históricos de sorteos. Estos datos permitirían guiar al usuario para decidir qué tipo de apuesta desea realizar. El sistema de gestión de base de datos (*Sistema De Gestión De Bases De Datos*, n.d.) elegido será Postgres (*PostgreSQL*, n.d.)etc etc

Objetivos y campo de aplicación

El trabajo tiene por objetivo principal el desarrollo de un sistema para obtener patrones y reglas para predecir una combinación ganadora de los dos sorteos considerados, lotería primitiva y bonoloto, en función de los datos de sorteos previos. Como objetivos secundarios relacionados con el objetivo principal se incluyen los siguientes:

1. Recopilación de los datos de los sorteos realizados hasta la fecha y su actualización sencilla con los nuevos sorteos que se celebren, de tal forma que la base de datos esté siempre actualizada. Los registros individuales de cada sorteo almacenan su fecha, cada número de la combinación ganadora en el orden de su aparición en el sorteo.
2. Recogida optimizada y ágil de datos con web scraping y actualización diaria y semanal de los datos en las tablas mediante utilización de hilos.
3. Generación de estadísticas básicas con números más habituales en combinaciones ganadoras, etc.
4. Generación de tablas de análisis de últimas coincidencias en sorteos anteriores y distancia de sorteos entre el presente y la coincidencia pasada como base para criterios de predicción.
5. Generación automática de propuestas de predicción tras generar una función de precisión con potenciales aciertos en toda la serie histórica tras seleccionar un criterio de elaboración de predicciones.

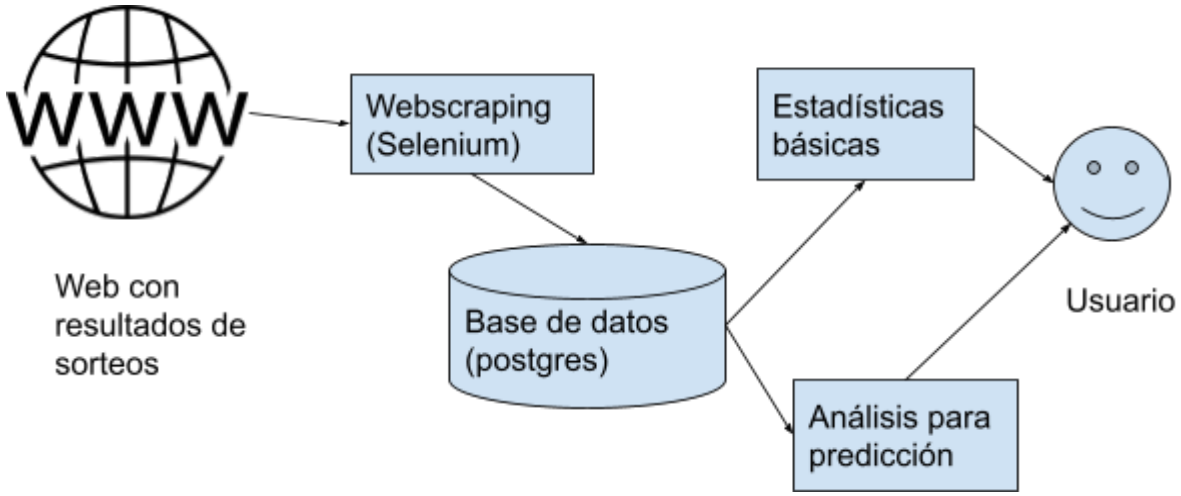
Las limitaciones en el alcance y características acordadas para la viabilidad de este trabajo son las siguientes:

1. El sistema de web scraping se personaliza para su uso con el sitio web seleccionado y para los dos juegos de sorteo elegidos. Aunque su adaptación a otros sitios web y sorteos es posible y se sugerirá como líneas de trabajo futuras, no se ha incluido en el trabajo para que fuera viable dentro de los límites de esfuerzo y tiempo establecidos. En el caso de los sorteos, al cambiar la configuración y las reglas, obligaría a repensar los criterios de análisis de datos y rediseñar los algoritmos de propuestas de predicción.

2. No se pretende que el sistema cuente con un frontal web para su manejo ni implantarse con otros sistemas de bases de datos, aunque, obviamente, ambas opciones son posibles y también se sugerirá como líneas de trabajo futuras. El sistema estará disponible como instalación final mediante máquina virtual para facilitar su posible uso futuro.
3. No se pretenden implantar métodos avanzados de aprendizaje y ajuste de resultados ni con entornos que los facilitan, sino solo experimentar realizando una implementación manual directa, con conceptos básicos de funciones objetivo y análisis con bases de datos.

Descripción del trabajo

Hay que hacer un diagrama de bloques básico del proceso que hace el sistema (en realidad lo de la introducción es tan detallado ya que buena parte debería ir aquí y en la introducción dejar más la justificación)



Metodología y plan de trabajo

Planteamiento y diseño del sistema

Se comenzará con la realización de un pequeño esquema y plan de ruta a seguir durante el desarrollo del trabajo. Se harán esquemas de la estructura de la base de datos y esquemas de los flujos de actividad para los hilos para la posible optimización de las funciones definidas en Python que actúan.
Se tendrá que determinar hasta qué punto el análisis de los datos recogidos debe ser exhaustivo y elegir los criterios a tomar para obtener los resultados deseados.
El enfoque a tomar en las fases debe ser teniendo en cuenta las fases futuras de optimización del sistema y detección de fallos (especialmente a la hora de tratar la

base de datos) para que se haga de una manera eficiente y sin redundancia de tareas.

Desarrollo

Fase de programación del sistema siguiendo las pautas de la fase anterior. Creación de la base de datos en PostgreSQL, introducción de los datos y comprobación de fallos en el proceso. Generación de nuevas tablas y junto con análisis de los datos secundarios obtenidos. Comprobación de capacidad de resistencia en caso de fallo en la base de datos.
La estructura final del trabajo constará de un bloque de código desarrollado en Python y usando la herramienta Selenium. Este código se encarga de introducir los registros en la base de datos de la aplicación y la actualización tanto de los datos a coger de la web como de los generados a partir de estos para la fase de análisis. La fase de análisis será realizada también mediante una serie de funciones en código Python. Habrá, como se ha especificado previamente, una base de datos relacional implementada en PostgreSQL que albergará todos los registros capturados y generados, usados durante recolección y análisis. El acceso a esta base de datos se hará de manera remota y por parte del código Python sin tener que recurrir a la interfaz en ningún momento del desarrollo. Tanto inserción de datos, actualización y extracción de estos es iniciado y completado por las clases presentes en el archivo Python.
Un posible task scheduler que se dedique a hacer la actualización de manera automática. Actualización mediante hilos, junto con los pasos anteriores de actualización de datos de ganadores, de los datos de última coincidencia. Se implementará a la par que la actualización de los datos, una generación de información de último sorteo en el que aparece el dato ganador y la diferencia de tiempo entre el actual y este. Uso de hilos también para la creación de las tablas y la generación de los datos introducidos en ellas.
La información vendrá ofrecida por una aplicación de escritorio, no contará con frontend online.

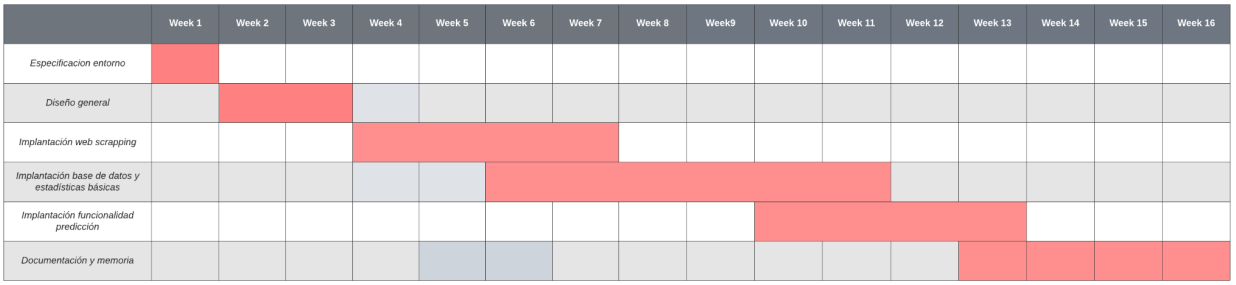
Evaluación de resultados

Análisis de los datos obtenidos y la coherencia de estos. Se plantea como objetivo una predicción de combinación a elegir en el futuro, comprobación de la calidad de las predicciones y la precisión de éstas con la introducción progresiva de nuevos datos introducidos en el sistema.

Posible optimización

Optimización del sistema de análisis y de obtención de datos. En este último mejorar la cantidad de tiempo necesario para la extracción. Esta fase se puede desarrollar con pruebas de tiempos entre distintas métodos de uso de hilos

Plan de trabajo



- Especificación del entorno: 1 semana
- Diseño general de la aplicación: 2 semanas
- Implantación del web scraping: 4 semanas
- Implantación de base de datos y de las estadísticas básicas: 6 semanas
- Implantación de la funcionalidad de predicción: 4 semanas
- Documentación y memoria: a lo largo de casi todo el proyecto y extendida más allá del final de la anterior, 3 semanas aproximadamente

Medios

Teniendo en cuenta que cada fase del proyecto requerirá de una herramienta diferente tanto para la implementación como para el desarrollo en el tiempo de los análisis de los resultados de los ganadores de cada sorteo:

-Python: el lenguaje de programación Python ofrece una ejecución rápida si se compara con otros lenguajes. La API utilizada en Python facilita la conexión al navegador a través de Selenium. Selenium-WebDriver está completamente implementado y soportado en Python. Es decir, la API de Selenium 2.0 tiene menos llamadas que el API de Selenium 1.0

-Selenium: el webdriver selenium es elegido por su popularidad y el gran soporte que se puede encontrar en la web. Cuenta con soporte Python y funciona con múltiples sistemas operativos en caso de ser necesario migrar el sistema a otro dispositivo. Permite, además, el uso de un sistema concurrente multihilo donde el webdriver es abierto en cada uno de estos.

-PostgreSQL: servidor de bases de datos utilizado durante las asignaturas de bases de datos de la carrera de ingeniería informática. Desarrollado en código abierto y gratuito son ventajas que se han tenido en cuenta, además de ser multiplataforma. Cuenta con una interfaz gráfica que nos puede ayudar a lo largo del desarrollo del

sistema para comprobar que no hay errores en las ejecuciones o datos introducidos en orden o formato incorrecto.

-Scheduled task: será útil para la automatización completa del proyecto. No será necesaria la presencia de un usuario para accionar el sistema de actualización y análisis en función de los nuevos registros en la base de datos. Permite establecer un periodo de tiempo que dice cada cuánto una porción de código(tarea) se debe ejecutar.

Se usará github para facilitar la gestión de los elementos del desarrollo de software y de base de datos. Así mismo, la configuración final del sistema definitivo se dejará preparado en una máquina virtual con VMWare que facilite su implantación y uso futuro.

En cuanto al uso de equipos y hardware, las pruebas realizadas confirman que el ordenador portátil disponible por el alumno será suficiente para el manejo de datos y la capacidad de procesamiento requerida. El ordenador cuenta con las siguientes características:

- Procesador: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz
- RAM instalada: 16,0 GB (15,8 GB usable)
- Tipo de sistema Sistema operativo de 64 bits, procesador basado en x64

Bibliografía

PostgreSQL. (n.d.). Wikipedia. Retrieved November 13, 2022, from https://es.wikipedia.org/wiki/PostgreSQL

Python. (n.d.). Wikipedia. Retrieved November 13, 2022, from https://es.wikipedia.org/wiki/Python

Selenium. (n.d.). Wikipedia. Retrieved November 13, 2022, from https://es.wikipedia.org/wiki/Selenium

Sistema de gestión de bases de datos. (n.d.). Wikipedia. Retrieved November 13, 2022, from https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos

Libros consultados:

CHAPAGAIN, Anish. *Hands-On Web Scraping with Python: Perform advanced scraping operations using various Python libraries and tools such as Selenium, Regex, and others*. Packt Publishing Ltd, 2019. https://www.oreilly.com/library-access/?next=/library/view/Hands-on-web/9781789533392/

Learn PostgreSQL : build and manage high-performance database solutions using PostgreSQL 12 and 13. Luca Ferrari author Enrico Pirozzi author, 1st edition. Birmingham ; Mumbai : Packt Publishing 2020 https://www.oreilly.com/library-access/?next=/library/view/Learn-PostgreSQL-/9781838985288/