



Memoria Proyecto

Sistemas de Información de
gestión y Business Intelligence

Adrián Diez Valbuena

ÍNDICE

ÍNDICE	1
INTRODUCCIÓN	2
OBJETO DEL TRABAJO	2
- Ámbito de las apuestas:	2
- Ámbito de competición:	3
OBJETIVOS	4
HERRAMIENTAS Y METODOLOGÍA	4
TRABAJO	5
¿Qué hace el software?	5
¿Cómo lo hace?	9
ANÁLISIS CRÍTICO	14
ANÁLISIS DAFO.....	14
Debilidades	15
Fortalezas	15
Amenazas	15
Oportunidades	15
LÍNEA DE FUTURO	16
BIBLIOGRAFÍA	17

INTRODUCCIÓN

Esta es la memoria de mi trabajo de la asignatura Sistemas de Información de gestión y Business Intelligence (SIBI), y en ella trataré de explicar todo el contexto del propio trabajo, sus objetivos y como funciona.

Mi proyecto se trata de una aplicación capaz de pronosticar resultados de carreras ciclistas. Esta idea surge ya que es un deporte que me gusta y creo que es lo algo que podría realizar gracias a la plataforma de Neo4j de manera que sea capaz de obtener resultados eficaces y realistas.

Para ello he utilizado la plataforma de Neo4j Desktop para crear y manejar una base de datos que contenga toda la información que voy a necesitar, y una vez realizada, he implementado las funcionalidades de la aplicación mediante el lenguaje de programación Python.

OBJETO DEL TRABAJO

El principal y más importante objeto del trabajo es el ciclismo, y, en concreto, la predicción de resultados basándonos simplemente en resultados anteriores.

Esta idea puede tener una gran utilidad en dos principales ámbitos:

- **Ámbito de las apuestas:**

Es el ámbito principal del trabajo ya que, tal y como ha sido implementada la aplicación es al cual más puede aportar.

El ciclismo está empezando a evolucionar y extenderse por el mundo poco a poco. Es por lo cual que ya se está empezando a apostar en este deporte, algo que puede ser muy llamativo para las personas que vean o practiquen el deporte. Es por ello que esta aplicación puede ofrecer una gran ayuda a la hora de realizar estas apuestas.

Por ejemplo, muestro una imagen de las cuotas que fueron establecidas para las apuestas del Giro de Italia de este año (2019):

PARTICIPANTES			
	Ganador	Entre 3 primeros	Entre 10 primeros
P. Roglic	2,75	1,45	1,18
Tom Dumoulin	3,25	1,50	1,18
Simon Yates	4,00	1,65	1,22
V. Nibali	8,00	2,50	1,18
M.A. Lopez Moreno	10,00	3,00	1,18
M. Landa	18,00	4,00	1,25
R. Carapaz	50,00	8,00	1,30

Mi sistema es capaz de dar la mejor opción para cada corredor en cada carrera, lo que podría ayudar en gran medida la decisión a tomar en caso de querer apostar, permitiendo intuir si una cuota es excesiva o si, por el contrario, es una gran oportunidad.

La aplicación podría tener un gran éxito en este contexto debido, principalmente, al desconocimiento del rendimiento de algunos corredores en carreras determinadas. Esto no sería el caso del Tour de Francia o la Vuelta a España ya que son carreras bastante seguidas, sino que iría más enfocado a carreras como monumentos (carreras de un solo día) en las que en este país hay un mayor desconocimiento y menor influencia de los medios de comunicación.

- **Ámbito de competición:**

En menor medida, esta aplicación podría servir a los equipos a la hora de preparar una estrategia para las carreras. A pesar de no estar preparada totalmente para ello, es posible saber que corredor es el favorito en una determinada carrera y gracias a ello controlarlo de manera que no se vaya para poder dar un vuelco a la carrera.

Es posible que en un futuro se pueda modificar la aplicación de manera que se puedan dar recomendaciones acerca de estas situaciones.

OBJETIVOS

El principal objetivo de este proyecto, y del taller que es la asignatura en general, es aprender a trabajar con un nuevo software que puede que dentro de no mucho sea el futuro, en concreto Neo4j. Además de trabajar con grafos de conocimiento de manera que se puedan manejar los datos que contiene de una manera sencilla y eficaz.

En cuanto al reto que me he propuesto a realizar gracias a esta plataforma, lo principal es conseguir realizar un sistema consistente comprendido por una base de datos y un código en Python que sea capaz de pronosticar resultados de carreras ciclistas, de manera que los resultados obtenidos sean razonables y útiles para manejar y poder ayudar a gente que no tenga un amplio conocimiento en el tema, principalmente, en el tema de las apuestas que poco a poco van cogiendo fuerza en este deporte.

HERRAMIENTAS Y METODOLOGÍA

Estas son las herramientas que he ido utilizando a medida que he ido realizando la aplicación:

- **Neo4j Desktop:** plataforma de escritorio del software de Neo4j, esta plataforma permite la creación de bases de datos e interacción con las mismas. Las bases de datos con las que funciona son orientadas a grafos, lo cual nos permite acceder a la información de una manera más rápida. Es parte fundamental del proyecto ya que sin ella no podría haber realizado la base de datos fundamental para el funcionamiento del sistema. El lenguaje de bases de datos con el que trabaja Neo4j es Cypher.
- **Excel:** he utilizado este programa simplemente a la hora de hacer la recopilación de datos necesaria para la realización de la base de datos, por lo que no tiene mayor importancia.
- **Python:** Lenguaje de programación que he utilizado para la realización del código que implementa las funciones que comprenden la aplicación.
- **Sublime Text:** Editor utilizado para la realización del código que nos permite ejecutarlo también, teniendo instalado el lenguaje de programación necesario.

En cuanto a la metodología utilizada, ésta viene dada, en gran parte, gracias a la página web de Neo4j, que nos permite aprender a utilizar su software mediante unos cursos que nos ofrecen en la misma.

El curso fundamental es el de introducción, ya que nos introduce en el tema de las bases de datos orientadas a grafos o grafos de conocimiento, nos inicia en el manejo de la aplicación de Neo4j y, por último, nos enseña a aprender el lenguaje de bases de datos que ellos utilizan, Cypher. En mi opinión, creo que un curso muy bien detallado, sencillo de entender y muy útil.

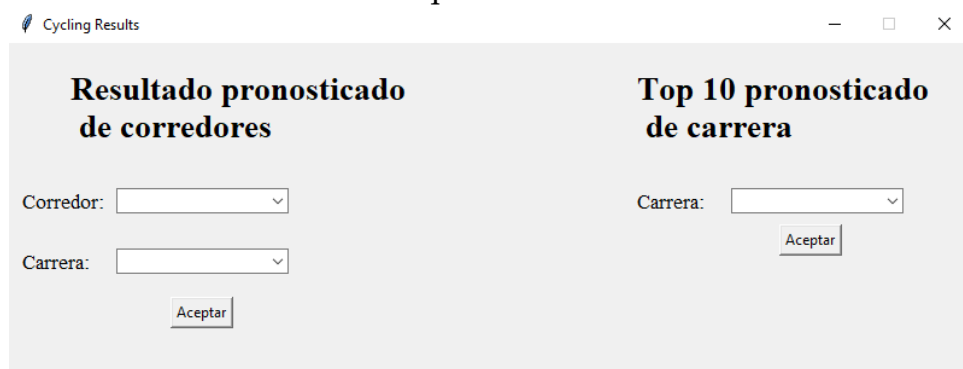
TRABAJO

Esta es la parte más extensa de esta memoria y trata sobre la aplicación en sí, en este apartado trataré de explicar lo que hace la aplicación y como lo hace.

¿Qué hace el software?

La aplicación consta de una ventana principal, que es la que se muestra al ejecutar la misma. En esta pantalla tenemos dos diferentes opciones:


- Pronosticar el resultado de un corredor para una carrera determinada seleccionando en un combo box el corredor y en otro la carrera. Solo estarán disponibles los corredores y carreras que están introducidas en la base de datos, pudiendo introducirse en cualquier momento más de ambos y seguiría funcionando correctamente.
- Obtener el top 10 pronosticado de una carrera concreta, seleccionándola en el lado izquierdo de la ventana.




Una vez pulsado el botón aceptar, la aplicación mostrará una nueva ventana con las predicciones obtenidas por el código.

En caso de haber optado por la primera opción, es decir, obtener el resultado de un corredor para una cierta carrera, la ventana mostrada contendrá datos sobre el mismo corredor junto con sus opciones en la carrera seleccionada anteriormente.

Por ejemplo, la predicción para el corredor Stybar en la carrera Paris-Roubaix:



Zdenek Stybar
Czech Republic 
33 años
Deceunink-Quick Step

Resultados anteriores:
2015: 2
2017: 2
2018: 9
2019: 8

Este corredor tendrá altas opciones de quedar en primer lugar.

Este sería el resultado para una carrera como Paris-Roubaix, que es una carrera de un día. Sin embargo, para carreras por etapas, puede haber alguna variación en caso de que el corredor sea un esprintero que dispute la clasificación por puntos o un escalador que lo haga con la clasificación de montaña:



Peter Sagan
Slovakia 
29 años
Bora-Hansgrohe

Resultados anteriores:
Peter Sagan ha ganado 12 etapas en esta carrera.

Este corredor no ha corrido o disputado la clasificación general de esta carrera. No hay datos suficientes para pronosticar.

Es uno de los candidatos a ganar la clasificación de la regularidad.

Resultados Corredor



Rafal Majka

Poland

30 años

Bora-Hansgrohe

Resultados anteriores:

2015: 3
2017: 39
2018: 13
2019: 6

Rafal Majka ha ganado 1 etapas en esta carrera.

Este corredor tendrá muchas posibilidades de quedar entre los 10 primeros.

Podrá ganar el maillot de la montaña.

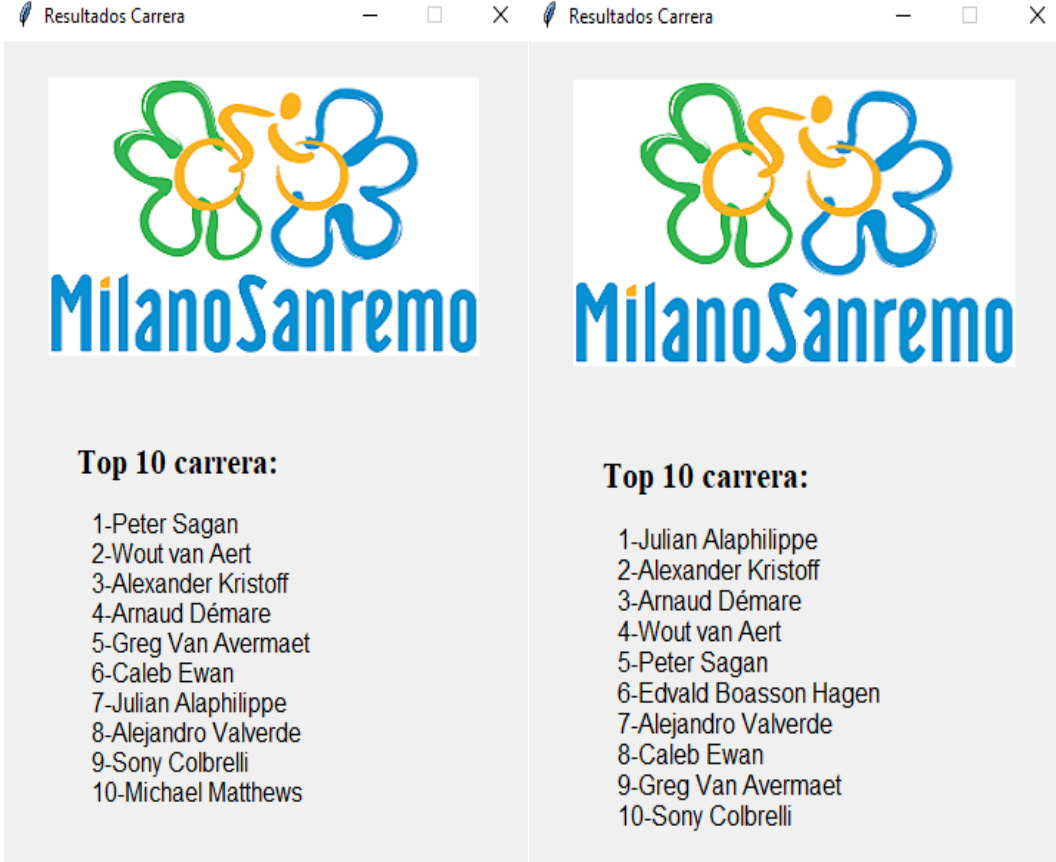
Todo esto serían casos correspondientes a la primera funcionalidad de la aplicación, en caso de querer pronosticar el top 10 de una carrera en concreto, la aplicación mostrara esto en una ventana nueva, además en el caso de las vueltas por etapas también pronosticará los ganadores de las clasificaciones de la regularidad y la montaña.

Como he mencionado en el objeto del proyecto, este tipo de predicciones pueden ser muy útiles a la hora de decidirse por un corredor a la hora de realizar una apuesta.

Por ejemplo, si vemos que una casa de apuestas propone que un top 10 de Rafal Majka en la Vuelta se paga a muy buena cuota, podemos ver que esta sería una buena decisión gracias a la aplicación.

Como el ciclismo es un deporte muy difícil de predecir ya que entran en juego diferentes factores (como caídas, pinchazos, enfermedades...) que, en cierto modo, son ajenos al corredor he introducido un factor aleatorio que puede hacer variar este top 10. Este factor lo explicaré posteriormente, ahora mostraré imágenes de estas variaciones.

Por ejemplo, si pido en dos ocasiones que me muestre un top 10 de la carrera Milano-San Remo:



Resultados Carrera

Milano Sanremo

Top 10 carrera:

- 1-Peter Sagan
- 2-Wout van Aert
- 3-Alexander Kristoff
- 4-Arnaud Démare
- 5-Greg Van Avermaet
- 6-Caleb Ewan
- 7-Julian Alaphilippe
- 8-Alejandro Valverde
- 9-Sony Colbrelli
- 10-Michael Matthews

Resultados Carrera

Milano Sanremo

Top 10 carrera:

- 1-Julian Alaphilippe
- 2-Alexander Kristoff
- 3-Arnaud Démare
- 4-Wout van Aert
- 5-Peter Sagan
- 6-Edvald Boasson Hagen
- 7-Alejandro Valverde
- 8-Caleb Ewan
- 9-Greg Van Avermaet
- 10-Sony Colbrelli

Como se puede observar, en el primer caso Peter Sagan aparece en primer lugar, sin embargo, en el segundo, la aplicación le pronostica un 5º puesto. Esto se debe a que el factor aleatorio anteriormente dicho ha variado el resultado final.

En caso de ejecutar dos veces para obtener el top 10 de la Vuelta a España:

Resultados Carrera



Top 10 general:

- 1-Primoz Roglic
- 2-Chris Froome
- 3-Thibaut Pinot
- 4-Tom Dumoulin
- 5-Tadej Pogacar
- 6-Alejandro Valverde
- 7-Nairo Quintana
- 8-Steven Kruijswijk
- 9-Miguel Ángel López
- 10-Rigoberto Urán

Ganador regularidad:

Elia Viviani

Ganador montaña:

Julian Alaphilippe

Resultados Carrera



Top 10 general:

- 1-Chris Froome
- 2-Primoz Roglic
- 3-Nairo Quintana
- 4-Alejandro Valverde
- 5-Miguel Ángel López
- 6-Tadej Pogacar
- 7-Thibaut Pinot
- 8-Tom Dumoulin
- 9-Steven Kruijswijk
- 10-Rigoberto Urán

Ganador regularidad:

Elia Viviani

Ganador montaña:

Thomas de Gendt

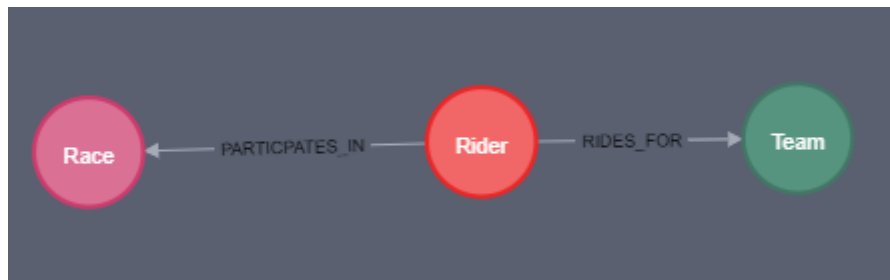
Podemos observar de nuevo variaciones en todos los resultados y también en las clasificaciones de la regularidad y la montaña.

¿Cómo lo hace?

En primer lugar, los datos son almacenados en una base de datos orientada a grafos creada con el Neo4j Desktop.

En la base de datos almaceno datos sobre corredores, equipos, y carreras. Además, almaceno los resultados obtenidos en los últimos 5 años en la relación entre corredor y carrera.

La base de datos posee el siguiente esquema:



La información más relevante es la que el propio corredor posee, así como los resultados de el mismo en cada carrera, que van contenidos en la relación entre ambos.

Estos serían los atributos que contienen los nodos de corredores, donde mountain significa si es un corredor que suele disputar la clasificación de la montaña y points es si el corredor suele pelear por ganar la clasificación de la regularidad:

```
{
  "name": "Alejandro Valverde",
  "country": "Spain",
  "mountain": "no",
  "age": 39,
  "points": "no"
}
```

Aquí podemos ver un ejemplo de la relación entre un nodo de corredor y los nodos de las carreras en las que participa:



Este sería el esquema de las relaciones entre corredor y carreras, simplemente es un vector que almacena los resultados de más antiguo a más reciente:

```
{
  "results": [
    "4",
    "1",
    "27",
    "6",
    "11"
  ]
}
```

Con todo esto tengo toda la información necesaria para realizar la aplicación y finalizo la base de datos. A continuación, he creado el código en Python.

Lo primero que hace el programa al ejecutarse es sacar todos los corredores y carreras contenidos en la base de datos y colocarlos en los como box.

A la hora de realizar las predicciones (aquí la parte más importante de la aplicación), he decidido establecer yo mismo unos coeficientes a los resultados anteriores almacenados en el grafo. He decidido tomar esta decisión ya que he buscado algunos algoritmos de predicción y ninguno me ha convencido para realizar este tipo de predicciones.

Procedo a explicar como he establecido los coeficientes.

En primer lugar, calculo cuantos resultados ha obtenido el corredor en la carrera en cuestión en los últimos cinco años, es decir, si un corredor solo ha participado 3 veces de las últimas 5, eso contará como 3 resultados. Ahora el sistema tiene 3 posibilidades:

- Si tiene 5 o 4 resultados, realizo una media ponderada de estos, de manera que los resultados más recientes ponderan más que los más antiguos. El coeficiente momentáneo sería esta media.
- Si tiene 2 o 3 resultados, simplemente realizo una media aritmética de estos. El coeficiente momentáneo sería esta media.
- En el caso de solo tener un solo resultado, teniendo en cuenta que en el ciclismo pueden ocurrir muchas casualidades, el coeficiente es igual a ese resultado multiplicado por 1.4.

```

coeficiente = 0
if(contador == 5):
    coeficiente = 0.3 * float(res[0]) + 0.25 * float(res[1]) + 0.2 * float(res[2]) + 0.15 * float(res[3]) + 0.1 * float(res[4])
elif(contador == 4):
    coeficiente = 0.3 * float(res[0]) + 0.27 * float(res[1]) + 0.23 * float(res[2]) + 0.2 * float(res[3])
elif(contador == 3):
    coeficiente = (float(res[0]) + float(res[1]) + float(res[2]))/3
elif(contador == 2):
    coeficiente = (float(res[0]) + float(res[1]))/2
elif(contador == 1):
    coeficiente = float(res[0])
    if(float(res[0]) < 4):
        coeficiente = float(res[0])*1.4

```

Una vez tengamos el coeficiente correspondiente a los resultados, procedemos a aplicar el factor aleatorio. Este factor simplemente se trata de calcular un aleatorio entre 1 y 6, y, en caso de ser 1, 2, o 3, se le sumará al coeficiente anterior 2, 3.5 y 4.5 respectivamente.

Por último, si la carrera es un monumento, es decir, una carrera de un solo día, hago que, si el coeficiente obtenido hasta el momento es menor de 35 y si un aleatorio del 1 al 10 es igual a 3, el coeficiente obtenido hasta ahora se divide por dos. Esto lo hago ya que en este tipo de carreras es mucho más probable que ocurran resultados sorprendentes e inesperados, por lo tanto, creo que debería de ser así.

```

#Un factor aleatorio
if(coeficiente > 0):
    factor = random.randrange(6)
    if(factor == 1):
        coeficiente = coeficiente + 2
    elif(factor == 2):
        coeficiente = coeficiente + 3.5
    elif(factor == 3):
        coeficiente = coeficiente + 4.5

    if(coeficiente < 35 and random.randrange(10) == 3 and tipo == "Monument"):
        coeficiente = coeficiente/2

```

Todo esto va acerca de cómo obtener el coeficiente de un corredor para una carrera en concreto. Ahora explicare que ocurre si se quiere obtener un resultado concreto, o un top 10.

En caso de querer obtener la predicción de un top 10, la aplicación obtendrá todos los coeficientes de todos los corredores para la carrera seleccionada y simplemente los mostrará ordenados de menor a mayor.

Ahora bien, en el caso de querer un pronóstico de un corredor en concreto, la aplicación sacará el coeficiente correspondiente y después contemplará varios rangos de valores para el mismo:

- Si es monumento:
 - Si el coeficiente es igual a 0, dirá que el corredor no ha corrido la carrera.
 - Si es menor de 5, pronosticará que el corredor tiene altas opciones de quedar en primer lugar.
 - Si esta entre 5 y 15, se obtendrá que el corredor tendrá muchas posibilidades de entrar en el podio.
 - Si esta entre 15 y 35, la aplicación mostrará que el corredor podrá estar fácilmente en el top 10.
 - Si es mayor de 35 se pronosticará que no se esperan grandes cosas de ese corredor en la carrera.

- Si es vuelta por etapas:
 - Si el coeficiente es igual a 0, dirá que el corredor no ha corrido la carrera o disputado la general.
 - Si es menor de 3.5, pronosticará que el corredor tiene altas opciones de quedar en primer lugar.
 - Si esta entre 3.5 y 10, se obtendrá que el corredor tendrá muchas posibilidades de entrar en el podio.
 - Si esta entre 10 y 30, la aplicación mostrará que el corredor podrá estar fácilmente en el top 10.
 - Si es mayor de 30 se pronosticará que no se esperan grandes cosas de ese corredor en la carrera.

Y así sería como mi aplicación hace sus predicciones. Ahora explicare el porqué de esos valores.

Los valores que he ido aplicando no han sido fijos desde un inicio, ya que como he mencionado anteriormente, el ciclismo es un deporte difícil de predecir. Sin embargo, a medida que he ido obteniendo resultados, he ido modificando estos coeficientes de manera que vayan quedando resultados más realistas a mi parecer hasta que he encontrado los que más se asemejan a lo que busco.

En cuanto a que en los monumentos el rango de los coeficientes sea mayor, esto es debido a que son carreras donde corredores con menor identidad pueden obtener mejores resultados por el mero hecho de tratarse de un solo día. Sin embargo, en las vueltas por etapas esto no suele ocurrir, ya que día a día los corredores más fuertes van acaparando las posiciones altas de la clasificación general.

ANÁLISIS CRÍTICO

Una vez finalizada la práctica ya puedo analizar los resultados obtenidos y si he cumplido mis objetivos iniciales.

Teniendo en cuenta que para realizar este proyecto solamente he contado con un tiempo limitado de unos 2 meses y medio, algo que limita mucho la capacidad para implementar nuevas funcionalidades y entender nuevas tecnologías, estoy muy contento con los resultados obtenidos a pesar de que sí que podría ser una aplicación más completa.

En primer lugar, analizando los resultados que la aplicación produce, puedo decir que estoy bastante satisfecho ya que son resultados muy coherentes y que podrían ser reales en su mayoría.

En cuanto a las funcionalidades que implementa la aplicación, sí que me hubiera gustado implementar algún sistema que ayude a los equipos a preparar las estrategias de una carrera, si bien es cierto que podría servir en cierto modo, creo que no es algo que pueda ser completamente útil. Sin embargo, tal y como está programada la aplicación, este aspecto es algo que se podría implementar con más tiempo.

ANÁLISIS DAFO



Debilidades

- Tiempo limitado: La asignatura comienza a mediados de septiembre y se debe presentar el trabajo a mediados de diciembre, por ello no es fácil realizar un sistema completo del todo y que abarque una gran cantidad de funciones.
- Aplicación sencilla: Quizá a causa del tiempo limitado, la aplicación es sencilla y no muy llamativa, algo que podría ser un problema a la hora de atraer posibles usuarios.

Fortalezas

- Sistema de predicción eficaz: El sistema es capaz de predecir resultados que son muy realistas y posibles, por lo que hará que la aplicación sea bien valorada a la hora de obtener resultados.
- Aplicación fácil de usar: Es muy sencillo utilizar el sistema en sí, directamente viendo la pantalla principal solo hay que hacer tres clics con el botón para realizar una predicción.
- Posibilidad de aumentar base de datos: Tal y como esta programada la aplicación, es posible modificar la base de datos de manera que el sistema no sufra ningún cambio a la hora de efectuar las predicciones. Algo que es bastante importante a la hora de expandirse.

Amenazas

- Foros en la web: Existen foros en internet que permiten a los usuarios contactar con gente que posee bastante conocimiento en el mundo ciclista y pueden aconsejarlos a la hora de realizar alguna apuesta, por lo que es una competencia importante y directa.
- Falta de capacidad para abrirse al mercado: Al fin y al cabo esto es una práctica realizada en una asignatura por lo que, de momento, no hay medios para colocarse en el mercado.

Oportunidades

- Mercado sin explotar: El mercado de las apuestas de ciclismo está empezando a coger fuerza en los últimos años, por lo que hay un sitio en el mercado que ahora mismo está vacío.
- Desconocimiento de la gente: En muchos casos las personas quieren realizar apuestas pero no saben si las cuotas son rentables o si serían un gran riesgo, por lo que esta aplicación les podría ser muy útil.
- Baja competencia: No existen sistemas similares al que he realizado.

LÍNEA DE FUTURO

En cuanto al futuro, creo que tal y como está programada la aplicación podría ampliarse de muchas maneras y en muchos aspectos sin que su funcionamiento sufriese algún cambio a peor.

En primer lugar, es posible y bastante sencillo editar la base de datos para añadir más carreras y corredores. Además, se debería introducir en la carpeta “fotos”, las fotos correspondientes a su corredor y carrera. Simplemente haciendo esto la aplicación mostraría todos estos nuevos datos y podría trabajar sobre ellos sin tener que añadir nada al código de Python.

Es posible, y bastante interesante, configurar la aplicación para que pueda mostrar recomendaciones a equipos que estén compitiendo en una carrera y quieran obtener consejos acerca de a que corredores deben controlar en determinada carrera, así como poder elaborar una estrategia si el equipo dispone de varios corredores que a priori, podrían obtener un gran resultado en la clasificación final.

Otra función que sería posible implementar, es la de recomendar fichajes a determinado equipo, es decir, si por ejemplo un equipo como Movistar quiere mejorar en una carrera como Ronde Van Vlaanderen, el sistema buscaría que fichaje sería más recomendable y posible para el equipo teniendo en cuenta la nacionalidad, la edad o los compañeros de cada corredor.

BIBLIOGRAFÍA

- Sitio web del cual he obtenido todos los datos acerca de ciclistas y carreras, e incluso las fotos de la aplicación:
<https://www.procyclingstats.com/>
- Sitio web de Neo4j donde es posible descargar el software de base de datos y realizar los cursos de aprendizaje:
<https://neo4j.com/>
- Sitio web de Anaconda, necesario para programar en Python:
<https://www.anaconda.com/>
- Sublime Text, editor muy recomendado:
<https://www.sublimetext.com/>
- Sitio web de la asignatura SIBI:
<http://sicodinet.unileon.es/login/index.php>