Primera previa

Ajustes polinomiales

Autores:

Natalia Rios Agudelo

John Alejandro Obando Gil

*IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e:

natalia.rios@utp.edu.co

[john.obando@utp.edu.co](mailto:john.obando@utp.edu.co)

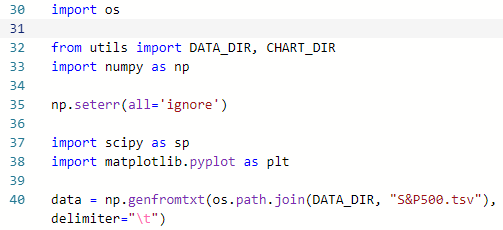
En este documento trabajaremos lo aprendido en la lección 6 de la materia computación blanda, aplicando ajustes polinomiales de grado 1, 2, 3 entre otros al índice bursátil Standard & Poor's 500, uno de los más importantes de Estados Unidos que representa la actualidad de las 500 empresas más relevantes de la bolsa de New York. Durante los días 18 de febrero hasta el 20 de marzo de 2020 este sufrió uno de las peores caídas en su historia, comparada con la del 2008 o incluso la gran depresión de 1927, las causas de esta gran caída se relacionan con el paro de la economía debido a la pandemia del covid-19, estudiaremos a lo largo de este documento las posibles situaciones en que podría concluir este problema para aquellos días, ya que lo que queremos es conocer el posible comportamiento que pudiera haber tomado al finalizar las siguientes dos semanas, si seguiría bajando, o tomaría alza.

Se tomaron 743 datos cada hora durante 31 días incluyendo fines de semana, pero como este índice no opera los fines de semana, estas horas se tomaron como valores erróneos o vacíos (nan) por lo que en total tenemos 23 días hábiles para un total de 551 datos con los que trabajaremos.

Para desarrollar el estudio de los ajustes polinomiales sobre el S&P500 se trabajó con las siguientes herramientas, python 3.7 y la plataforma repl.it, en las cuales se utilizaron las librerías de os, numpy, scipy y matplotlib con las cuales hemos venido trabajando a lo largo del curso, para la toma de datos se utilizó la herramienta Excel, llenando dos simples columnas, la primera con el índice de los datos (1-743) y la segunda con el valor de los datos (precio del índice bursátil).

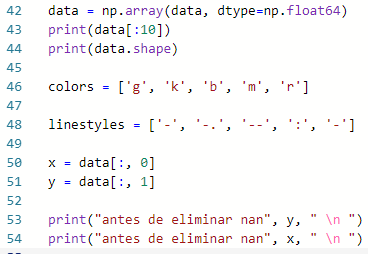
La forma en que trabajamos con los datos se dividió de la siguiente manera, se importaron los archivos de trabajo, se crearon las variables, se generaron los modelos y por último se graficaron las funciones.

SE IMPORTAN LOS ARCHIVOS DE TRABAJO



Se importan los directorios de trabajo, directorios de datos y directorios de gráficos que se mostrarán más adelante, se importa el archivo excel que contiene los datos con los que se trabajo, se importa la librería os que es la del sistema operativo, se también se importan las librería scipy y matplotlib las cuales respectivamente se utilizan para, Scipy que contiene módulos para optimización, álgebra lineal, integración, interpolación, funciones especiales, entre otros y matplotlib para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática Numpy.

SE CREAN LAS VARIABLES



Se establecen los tipos de datos, los colores y tipos de líneas a utilizar en las gráficas, imprimimos en consola el valor de los primeros 10 datos de nuestro archivo y el tamaño completo de nuestro archivo data, se crean los respectivos vectores para x & y, correspondientes a la primera y segunda columna del data que como se dijo anteriormente es el archivo de Excel que contiene el índice de datos(1-743) y el valor de los datos (precio del índice bursátil), se imprimen en consola los datos antes y después de eliminar los datos erróneos o vacíos, que más adelante explicaremos de una mejor forma.

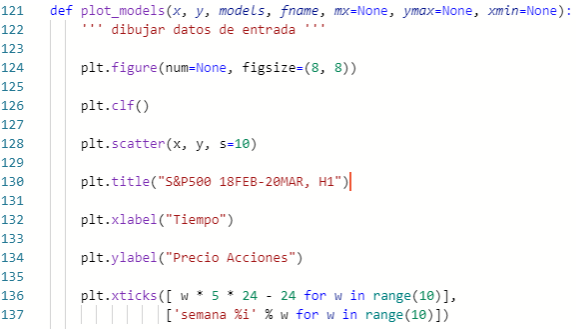
**Eliminación de datos erróneos:**

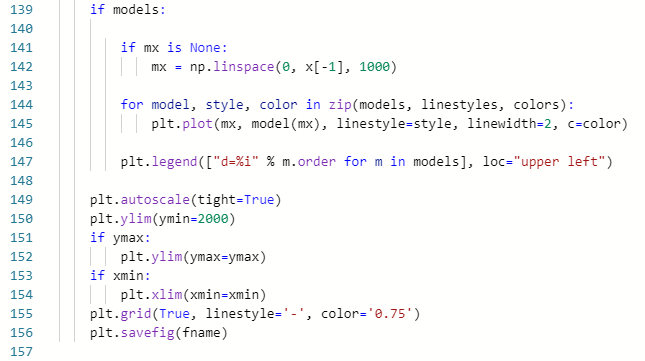


Se imprime el número de entradas incorrectas con valores erróneos o **nan** que pertenecen a los fines de semana en nuestro archivo data, con ayuda del método sum se suman todos los valores **nan** encontrados y se listan, la función **isnan** elimina estos valores vacíos o **nan** en el vector ya que estos valores afectan la visualización de los datos y el estudio óptimo de los mismos, se crea un nuevo vector **x** en el cual los valores **nan** han desaparecido y se aplica lo mismo sobre el vector y, lo cual hace que tanto el vector **x** como el vector **y** queden sincronizados, sin valores **nan,** para verificar que funciona correctamente se imprime en pantalla el antes y después de eliminar los valores erróneos de los vectores **x** & **y**, verificando la eliminación de los **nan**.

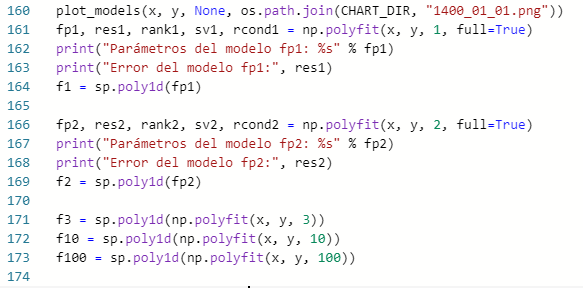
Al realizar el anterior proceso el índice del vector **x** queda con espacios en el vector es decir al llegar al primer fin de semana (dato 96) el siguiente valor en el vector es el dato 144 (48 datos después pertenecientes al día sábado y domingo de las 48 horas de los mismos) y así sucesivamente por cada fin de semana. Por este motivo es necesario comprimir de nuevo este vector **x** para reordenar su índice, y para ello usamos los métodos **list, range y len.** De esa forma nuestros vectores ahora quedan listos para graficar, con un total de 551 datos cada uno.

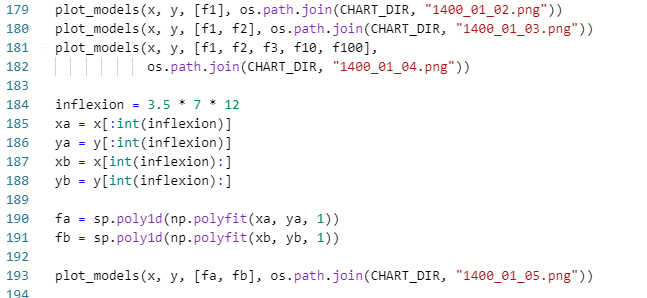
SE GENERAN LOS MODELOS





Para generar los modelos crearemos una única función la cual utilizaremos para todas las gráficas, dentro de la función declaramos elementos principales para el diseño de la gráfica empezando por su tamaño (8,8) el tamaño de los puntos que representan los datos (10), le ponemos un título principal, un título al eje **x** y uno al eje **y**, luego establecemos el rango de las semanas, que serán el tiempo en nuestro eje **x,** calculamos los valores de **w**, los cuales se agrupan en paquetes de **w**\*7\*24, esto permite determinar los valores de w desde 1 hasta 5, indicando con ello que se tiene un poco más de 4 semanas, se utilizó 5 para calcular las semanas ya que no se tomaron en cuenta los fines de semana, también se le resto a esta operación -24 porque la semana en la que iniciamos la tomamos desde el día 18 de febrero que fue martes y un día anterior a este fue festivo, por lo tanto esta semana fue de 4 días. Se establece el rango el rango en el eje y entre 2000 y el mayor valor de nuestro vector para una mejor visualización de los datos.



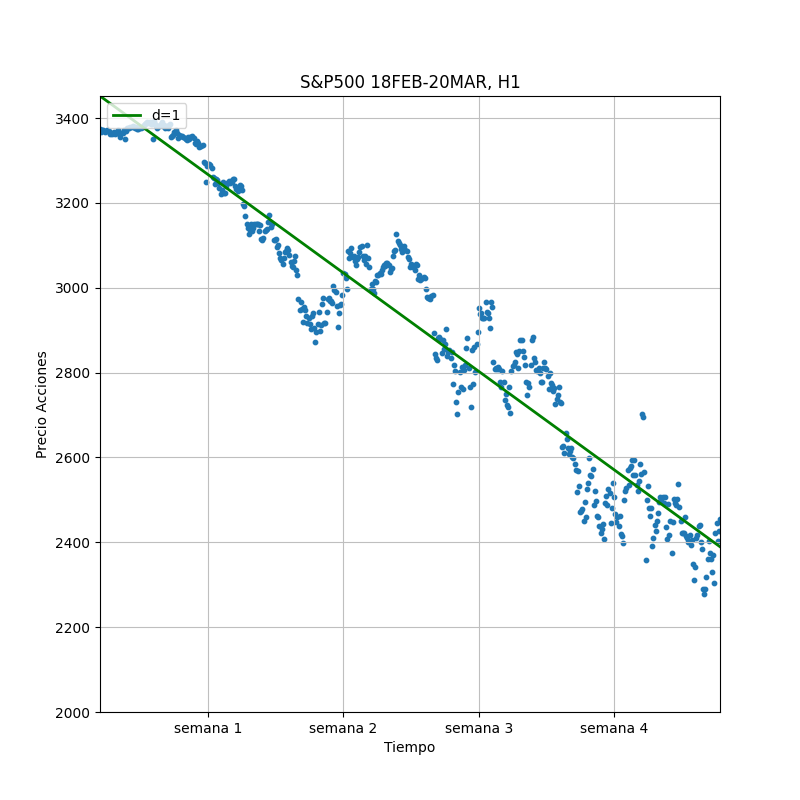
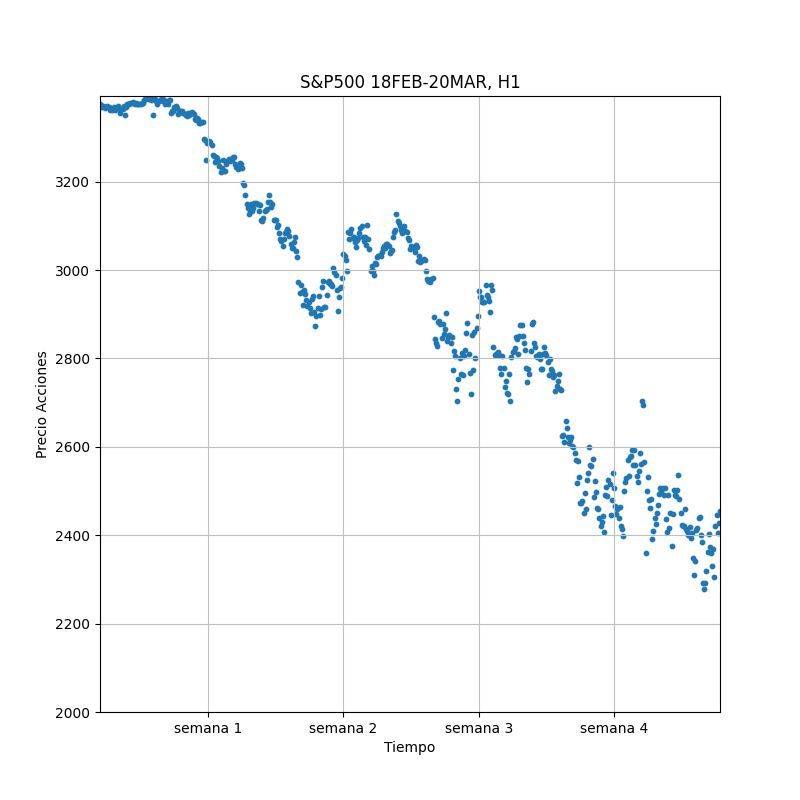


El módulo os.path es un submódulo del módulo OS en Python que se utiliza para la manipulación de nombres de rutas comunes, os.path.join() es un método que concatena varios componentes de ruta con exactamente un separador de directorio ('/') después de cada parte no vacía excepto el último componente de ruta. Ejecutamos la función creada anteriormente para definir un modelo y le pasamos como parámetros **x** & **y**, no agregamos ningún modelo, la ruta donde tenemos el archivo, el nombre del mismo, de esta forma tenemos la primera representación gráfica de nuestros datos.

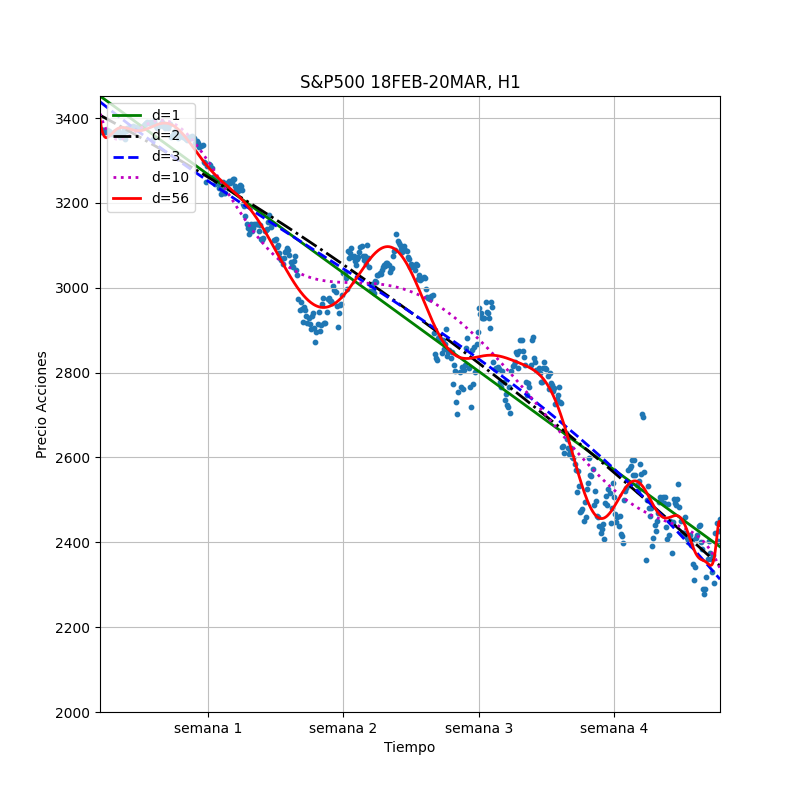
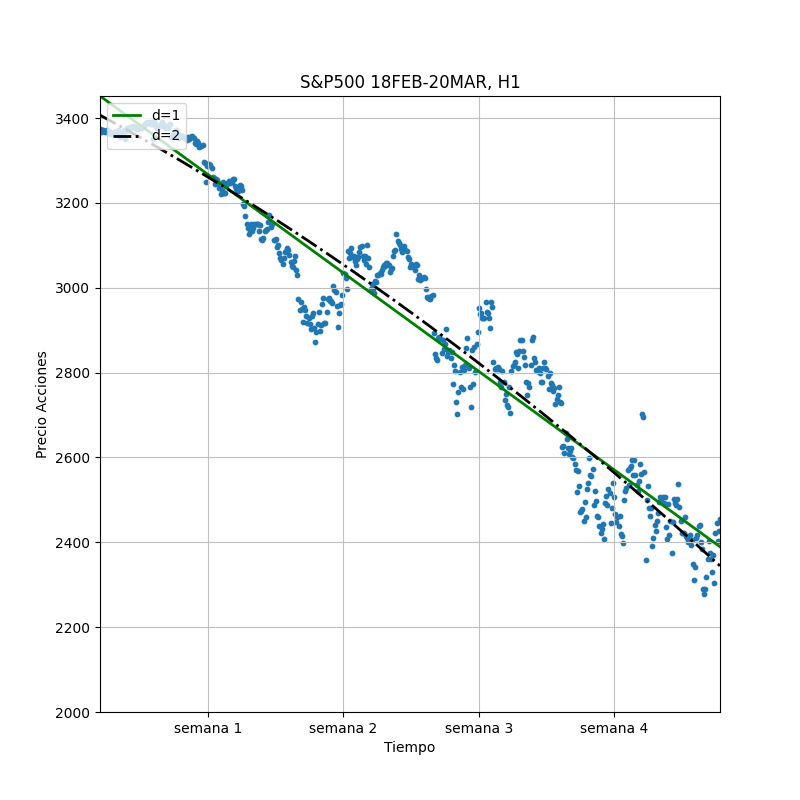
Llamamos de nuevo la función creada anteriormente para definir un modelo, esta vez lo hacemos para graficar los modelos y vamos agregar cada uno de nuestros diferentes modelos creados anteriormente, ahora ubicamos cada modelo, donde **f1**, **f2**, **f3**, **f10** y **f100**: es una función que tomara una matriz de valores observados (**x**)y devuelve una matriz de valores predichos (**y**), como tercer parámetro en nuestra función plot\_models, de esta forma graficamos 3 modelos más. Si observamos en la gráfica los puntos que no están dentro del rango de nuestros modelos representan errores, y entre más alejad, más aumenta la magnitud de nuestro error con cada modelo grabado podemos observar mejores resultados ajusta y dibuja un modelo utilizando el conocimiento del punto de inflexión.

SE GRAFICAN LAS FUNCIONES

Función Original (figura 1) Polinomio primer grado (figura 2)



Polinomio de segundo grado (figura 3) Polinomio de tercer y cuarto grado (figura 4)



CONCLUSIONES

* En la gráfica podemos observar los resultados que nos arrojaron nuestros datos del S&P500 tomados durante casi 5 semanas cada hora, se puede ver que la semana 1 es más corta debido a el lunes festivo por lo que se visualizan solo datos de 4 días, las semanas 2, 3 y 4 son de 5 días y la última semana nuevamente de 4 días, para un total de 23 días hábiles de operación contamos con datos que fluctúan en el rango de 3400 a 2200 puntos de cotización en el precio.

Con estos datos podemos observar en la gráfica una fuerte caída de más de 1200 puntos en tan solo un mes, lo cual se ve comprobado en la segunda figura con una pendiente muy inclinada hacia abajo, así mismo, en la tercera figura podemos refutar nuevamente con la función cuadrática una curvatura en dirección hacia la baja del índice, y por último en la figura 4 con los polinomios de 3 y 4 grado podemos ver las fuertes fluctuaciones que sufrió el mercado durante finales de mes en las últimas dos semanas, esto lo podíamos ver reflejado también en la primera figura al observar tantos puntos dispersos por toda la gráfica, este es un punto importante ya que impide que se pueda conocer con certeza la dirección que tomara en las próximas dos semanas, además de que los datos son muy pocos y en el mercado bursátil necesitaría una mayor cantidad de datos, ya que es muy importante conocer la historia del valor a estudiar.

BIBLIOGRAFIA

[1] <https://www.metatrader4.com/es>

[2] <https://numpy.org/>

[3] <https://www.scipy.org/>

[4] <https://matplotlib.org/>

[5] S&P 500. (2020, 29 de junio). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 20:55, septiembre 28, 2020 desde <https://es.wikipedia.org/w/index.php?title=S%26P_500&oldid=127347905>.