ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ ΜΕΜ704
Πρόβλεψη Χρηματιστηριακών Τάσεων με Μεθόδους Μηχανικής Μάθησης

ΔΟΥΛΗΣ ΠΑΝΑΓΙΩΤΗΣ, 1822 ΤΣΙΑΜΑΛΟΣ ΕΥΑΓΓΕΛΟΣ, 2604

26 Μαΐου 2021

ΕΙΣΑΓΩΓΗ

Στα πλαίσια αυτής της εργασίας είναι η πρόβλεψη τάσεων που αφορούν χρηματιστηριακές τιμές. Στόχος ήταν να εξετάσουμε την δυνατότητα πρόβλεψης μιας χρηματιστηριακής τιμής σε διάφορα χρονικά πλαίσια και με διάφορες μεθόδους. Η βασική ιδέα είναι να προσπαθήσουμε να δούμε αν είναι δυνατή η πρόβλεψη μιας τιμής γνωρίζοντας τις παρελθοντικές τιμές ή κάποιες άλλες βοηθητικές παροντικές τιμές.

ΜΕΘΟΔΟΛΟΓΙΑ

Τα δεδομένα που χρησιμοποιήσαμε προήλθαν από την Google Finance και αποτελούνται από το ιστορικό των ποσοτήτων που περιγράφουν την τιμή μιας μετοχής ή ενός κρυπτονομίσματος (dates , high-low prices, open-close prices, volume):
open-price: Η τιμή της αξίας της μετοχής κατά το άνοιγμα της αγοράς (πρώτη καταγραφή)
high: Η υψηλότερη τιμή που καταγράφηκε μέσα στην ημέρα
low: Η χαμηλότερη τιμή που καταγράφηκε μέσα στην ημέρα
close-price: Η τιμή της αξίας της μετοχής κατά το κλείσιμο της αγοράς (τελευταία καταγραφή)
volume: Το σύνολο όλων των συναλλαγών μέσα στην ημέρα
Πήραμε δεδομένα που περιγράφουν το ιστορικό σε καθημεριμή βάση (daily) ή εβδομαδιαία βάση (weekly) για 5 έτη. Έχοντας αυτά τα δεδομένα, χησιμοποιούμε smoothers για την ομαλοποίηση των δεδομένων προκειμένου να απαλλαχθούμε, έως έναν βαθμό, από την μεγάλη μεταβλητότητα που υπάρχει στις χρηματιστηριακές χρονοσειρές. Λαμβάνοντας τα ομαλοποιημένα δεδομένα, δοκιμάζουμε την δυνατότητα πρόβλεψης της τιμής της επόμενης μέρας ή εβδομάδας χρησιμοποιώντας αυτοπαλινδρομικό μοντέλο, γραμμική παλινδρόμιση και ένα νευρωνικό δίκτυο.

Smoothers
Για την εξομάλυνση της χρονοσειράς θα χρησιμοποιήσουμε τους Simple Moving Average, Weighted Moving Average, Exponential Moving Average.
Ο Απλός Κινητός Μέσος όρος (Simple Moving Average) τάξης l θα εξομαλύνει την χρονοσειρά με ένα φίλτρο που αφορά την κάθε χρονική στιγμή και τις προηγούμενες l-στιγμές της. Δηλαδή, αν $Y_t, t = 1, ..., n$ η χρονοσειρά, τότε ορίζουμε τον

$$SMA(t) = \hat{Y}_t = \frac{1}{l}\sum_{1}^{l} Y_{t-l+1} \text{ για } t = l, \ldots, n$$

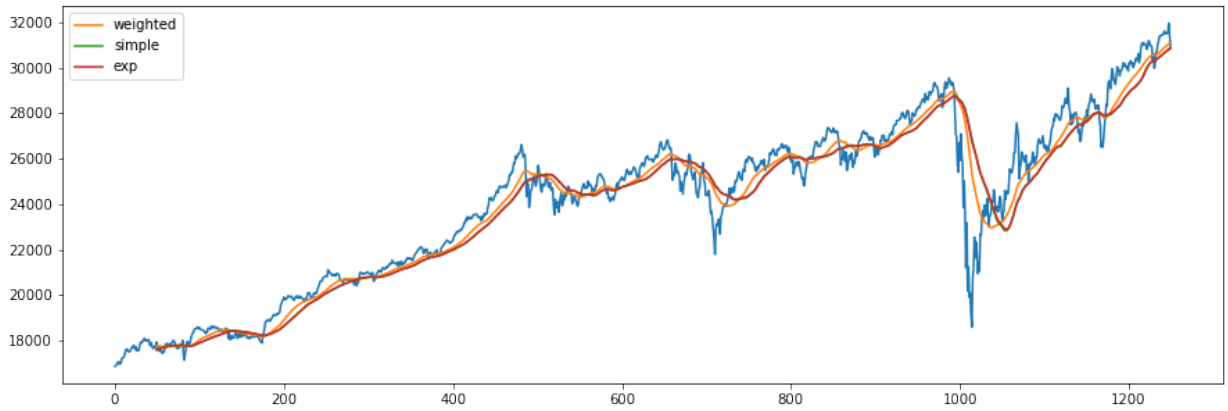
Ομοίως και ο Κινητός Μέσος Όρος με βάρη (Weighted Moving Average) τάξης l θα εξομαλύνει την χρονοσειρά με ένα φίλτρο που αφορά την κάθε χρονική περίοδο και τις προηγούμενες l-τιμές της, δήνοντας όμως μεγαλύτερο βάρος στις πιο πρόσφατες τιμές:

$$WMA(t) = \hat{Y}_t = \frac{\sum_{j=1}^{l} \frac{j}{l} \cdot Y_{t-j+1}}{\sum_{j=1}^{l} \frac{j}{l}} \text{ για } t = l, \ldots, n$$

Η Εκθετική Εξομάλυνση(Exponential Moving Average) τάξεως l χρησιμοποιεί τον Απλό Κινητό Μέσο Όρο και εκφράζεται ως εξής:
Αν $\hat{Y}_t t = l, \ldots, n$ είναι ο κινητός μέσος όρος l-τάξεως, τότε

$$EMO(t) = a \cdot \hat{Y}_t + (1-a) \cdot Y_{t-1} \text{ για } t = l, \ldots, n \text{ για } a = \frac{2}{l+1}$$

Οι εξομαλύνσεις αυτές χρησιμοποιούνται στα πλαίσια εμπειρικών κανόνων του χρηματιστηρίου.
Όσο πιο μεγάλο είναι το χρονικό παράθυρο του φίλτρου τόσο πιο ομαλή γίνεται η χρονοσειρά.



Σχήμα 1: Παράδειγμα Εξομάλυνσης Χρονοσειράς, Τάξη 50

Αυτοπαλινδρομικό Μοντέλο

Το αυτοπαλινδρομικό μοντέλο επιλέχθηκε διότι πρόκειται για μία από τις πιο απλές ιδέες για την πρόβλεψη τιμών σε μια χρονοσειρα στο βραχυχρόνιο επίπεδο. Το αυτοπαλινδρομικό μοντέλο k-τάξεως:

$$Y_t = \theta_0 + \sum_{j=1}^{k} \theta_j Y_{t-j}, k \geq 1$$

χρησιμοποιεί τις close-prices των k προηγούμενων στοιχείων της χρονοσειράς προκειμένου να προβλέψει την ζητούμενη. Βασίζεται στην γραμμική συσχέτιση των δεδομένων και ακολουθεί έντονα την τάση των τελευταίων ημερών. Για να το κάνουμε αυτό, σπάμε την χρονοσειρά στην μορφή:

$$\begin{aligned}
&(Y_1, \ldots, Y_k) \longrightarrow Y_{k+1} \\
&(Y_2, \ldots, Y_{1+k}) \longrightarrow Y_{2+k} \\
&\vdots \\
&(Y_{n-k}, \ldots, Y_{n-1}) \longrightarrow Y_n
\end{aligned}$$

Η μέθοδος που χρησιμοποιείται είναι αυτή των κανονικών εξισώσεων:
Ορίζουμε $\theta = (\theta_0, \ldots, \theta_k)^t \in \mathbb{R}_k$, και

$$X = \begin{pmatrix} 1 & Y_1 & Y_2 & \ldots & Y_k \\ 1 & Y_2 & Y_3 & \ldots & Y_{k+1} \\ \vdots & & & & \\ 1 & Y_{n-k} & Y_{n-k+1} & \ldots & Y_{n-1} \end{pmatrix} \text{ και } y = \begin{pmatrix} Y_{k+1} \\ Y_{k+2} \\ \vdots \\ Y_n \end{pmatrix}$$

τότε το θ προσδιορίζεται ως

$$\theta = (X^t X)^{-1} X^t y$$

Οι τιμές του k που δοκιμάζουμε είναι μικρές, καθώς χρησιμοποιώντας την συνάρτηση μερικής αυτοσυσχέτισης $PACF(k)$ δείχνει πως η πιο ισχυρή συσχέτιση βρίσκεται στις πιο πρόσφατες ημέρες. Η συνάρτηση μερικής αυτοσυσχέτισης εκφράζει την άμμεση επίδραση Y_{t-k} συνιστώσας στην Y_t.

Γραμμική Παλινδρόμιση

Αυτήν την φορά θα προσπαθήσουμε να προβέψουμε την τιμή του close-price παίρνοντας ως δεδομένες τις τιμές όλων των στοιχείων που περιγράφουν την χρηματιστηριακή τιμή. Αν Y_t είναι η close-price της μετοχής την χρονική στιγμή t, τότε θεωρούμε το μοντέλο γραμμικής παλινδρόμισης:

$$Y_{t+1} = a_0 + a_1 \cdot Y_t + a_2 \cdot high_t + a_3 \cdot low_t + a_4 \cdot open_t + a_5 \cdot volume_t + e_t$$

Συγκεκριμένα αναμένουμε πως το μεγαλύτερο βάρος, πέρα από το close-price, θα υπάρχει και στο volume, διότι δείχνει το πόση δραστηριότητα υπήρε στην αγορά.
Όπως προηγουμένως φέρνουμε τα δεδομένα στην μορφή:

$$a = (a_0, \ldots, a_5)^t \in \mathbb{R}^6$$

$$X = \begin{pmatrix} 1 & Y_1 & high_1 & low_1 & open_1 & volume_1 \\ 1 & Y_2 & high_2 & low_2 & open_2 & volume_2 \\ \vdots & & & & & \\ 1 & Y_n & high_n & low_n & open_n & volume_n \end{pmatrix} \text{ και } y = \begin{pmatrix} Y_2 \\ Y_3 \\ \vdots \\ Y_{n+1} \end{pmatrix}$$

Νευρωνικό Δίκτυο

Ουσιαστικά θα κάνουμε τις ίδιες διαδικασίες, αλλά με μοντέλα νευρωνικών δικτύων. Σε πρώτο στάδιο θα χρησιμοποιήσουμε πάλι τις τιμές του close-price των προηγούμενων k ημερών για να προβλέψουμε την ζητούμενη:
Αν h_θ είναι η πρόβλεψη του μοντέλου με το νευρωνικό δίκτυο, τότε θα είναι:

$$h_\theta(Y_{t-k}, Y_{t-k+1}, \ldots, Y_{t-1}) = Y_t + e_t$$

Τα νευρωνικά δίκτυα αναγνωρίζουν μη-γραμμική συσχέτιση και θα θέλαμε να δουμε την διαφορά σε σχέση με την γραμμική παλινδρόμιση.

Αρχικά αν $x = \begin{pmatrix} x(1) \\ x(2) \\ \vdots \\ x(k) \end{pmatrix} \in \mathbb{R}^k$, θεωρούμε το νευρωνικό δίκτυο:

$$\begin{aligned} a_0 &= x \\ a_1 &= \sigma(W_1 \cdot a_0 + b_1) \\ &\vdots \\ a_{r-1} &= \sigma(W_{r-1} \cdot a_{r-2} + b_{r-1}) \\ a_r &= h_\theta = W_r \cdot a_{r-1} + b_r \end{aligned}$$

όπου r ο αριθμός των επιπέδων, $a_j \in \mathbb{R}^{m_j}, j = 0, \ldots, r$ με m_j να είναι το πλήθος των κόμβων στο j-επίπεδο, για $j = 0, \ldots, r$,και $W_j \in Mat_{m_j \times m_{j-1}}(\mathbb{R})$, $b_j \in \mathbb{R}^{m_j}$, $j = 1, \ldots, r$ και σ η συνάρτηση ενεργοποίησης, η οποία μπορεί να είναι μια από τις:

$sigmoid(x) = \frac{1}{1+e^{-x}}$

$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

$relu(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$

Πρόκειται δηλαδή για νευρωνικό δίκτυο που κάνει παλινδρόμιση.

Οπότε και πάλι οργανώνουμε τα δεδομένα στην μορφή $X = \begin{pmatrix} Y_1 & Y_2 & \ldots & Y_k \\ Y_2 & Y_3 & \ldots & Y_{k+1} \\ \vdots & & & \\ Y_{n-k} & Y_{n-k+1} & \ldots & Y_{n-1} \end{pmatrix}$ (χωρίς τις

μονάδες στην πρώτη στήλη αυτήν την φορά) και $y = \begin{pmatrix} Y_{k+1} \\ Y_{k+2} \\ \vdots \\ Y_n \end{pmatrix}$ εάν θέλουμε να κάνουμε πρόβλεψη για την τιμή του close-price χρησιμοποιώντας τις προηγούμενες κ-τιμές της. Επίσης οργανώνουμε τα δεδομένα μας στην μορφή: $X = \begin{pmatrix} Y_1 & high_1 & low_1 & open_1 & volume_1 \\ Y_2 & high_2 & low_2 & open_2 & volume_2 \\ \vdots & & & & \\ Y_n & high_n & low_n & open_n & volume_n \end{pmatrix}$ και $y = \begin{pmatrix} Y_2 \\ Y_3 \\ \vdots \\ Y_{n+1} \end{pmatrix}$ αν θέλουμε να προβλέψουμε την τιμή του close-price χρησιμοποιώντας τις υπόλοιπες χρηματιστηριακές τιμές της ίδιας ημέρας.
Το νευρωνικό δίκτυο εφαρμόζει τον αλγόριθμο της μεγίστης κλίσης με χρήση δέσμης (Batches).

Κανονικοποίηση
Για την κανονικοποίηση των δεδομένων χρησιμοποιήθηκαν διάφοροι scalers. Αν $x = (x_1, \ldots, x_n)^t$ είναι οι τιμές ενός χαρακτηριστικού, τότε ορίζουμε

$$MinMaxScaler(x_j) = \frac{x_j - min_{1 \le i \le n} x_i}{max_{1 \le i \le n} x_i - min_{1 \le i \le n} x_i} \in [0,1] \text{ για } j = 1, ..., n$$

$$MaxAbsScaler(x_j) = \frac{x_j}{max_{1 \le i \le n} |x_i|} \in [0,1] \text{ για } j = 1, ..., n$$

$NormalScaler(x_j) = \frac{x_j - \mu}{\sigma}$ για $j = 1, ..., n$, μ ο δειγματικός μέσος όρος, σ η δειγματική διασπορά

ΥΛΟΠΟΙΗΣΗ-ΛΟΓΙΣΜΙΚΟ

Έγινε η προσπάθεια υλοποίησης των αλγορίθμων από το μηδέν. Το λογισμικό που χρησιμοποιήσαμε αποττελείται από 3 κλάσεις: myRegression, myNeuralNetwork, Utilities
ΠΡΟΣΟΧΗ: ΟΛΑ ΤΑ ΑΡΧΕΙΑ ΠΡΕΠΕΙ ΝΑ ΒΡΙΣΚΟΝΤΑΙ ΣΤΟ ΙΔΙΟ DIRECTORY ΓΙΑ ΝΑ ΤΡΕΞΟΥΝ. ΠΕΡΙΣΣΟΤΕΡΕΣ ΛΕΠΤΟΜΕΡΙΕΣ ΓΙΑ ΤΟΝ ΤΡΟΠΟ ΠΟΥ ΛΕΙΤΟΥΡΓΟΥΝ ΤΑ ΠΡΟΓΡΑΜΜΑΤΑ ΥΠΑΡΧΟΥΝ ΣΤΟ Project_info.txt

Class Utilities (Αρχείο ProjectUtilities)
Η κλάση Utilities περιέχει μεθόδους που βοηθούν στην υλοποίηση των άλλων δύο κλάσεων και μεθόδους που θα χρησιμοποιηθούν στην οργάνωση και προετοιμασία των δεδομένων. Περιέχει τις μεθόδους για την συνάρτηση ενεργοποίησης:

- Utilities.logistic
- Utilities.relu
- Utilities.tanh

και τις παραγώγους τους :

- Utilities.logistic_derivative
- Utilities.relu_derivative
- Utilities.tanh_derivative

τις μεθόδους για τους smoothers:

- Utilities.simpleMovingAverage(y,k): y διάνυσμα με την χρονοσειρά, k η τάξη του κινητού μέσου όρου
- Utilities.weightedMovigAverage(y,k): y διάνυσμα με την χρονοσειρά, k η τάξη του κινητού μέσου όρου

- Utilities.expMovingAverage(y,k): y διάνυσμα με την χρονοσειρά, k η τάξη της εκθετικής εξομάλυνσης

και την μέθοδο:

- Utilities.formatData(Y, k, addUnit = False): Λαμβάνει μια χρονοσειρά και την προετοιμάζει για την παλινδρόμιση η το νευρωνικό δίκτυο. Y διάνυσμα με την χρονοσειρά, k πόσα προηγούμενα στοιχεία θα κρατήσει ως δεδομένα

Class MyRegression (Αρχείο Project_Regression)
Η κλάση myRegression αφορά την γραμμική παλινδρόμιση που χρησιμοποιείται στην εργασία. Περιέχει το attriute:

- self.theta το οποίο είναι το διάνυσμα με τις παραμέτρους του μοντέλου στην μορφή $\theta = (\theta_0, \ldots, \theta_k)^t$

και τις μεθόδους

- fit(self,X,y) που εφαρμόζει την μέθοδο των κανονικών εξισώσεων και επιστρέφει το $\theta \in \mathbb{R}^{k+1}$. Ο πίνακας X περιέχει τα δείγματα σε γραμμές και πρέπει να μην έχει μονάδες στην πρώτη του στήλη. Το y είναι ένα διάνυσμα με τις αναμενόμενες τιμές.

- predict(self,X) που εφαρμόζει το μοντέλο για το θ που υπολογίστηκε κατά την εκπαίδευση και επιστρέφει το $y \in \mathbb{R}^n$ με τις προβλεπόμενες τιμές για κάθε δείγμα. Ο πίνακας X περιέχει τα δείγματα σε γραμμές και πρέπει να μην έχει μονάδες στην πρώτη του στήλη.

- loss(self,X,y) η οποία υπολογίζει το τετραγωνικό σφάλμα για το σύνολο δεδομένων $X \in \mathbb{R}^{n\times k}$ με αναμενόμενες τιμές το $y \in \mathbb{R}^n$ δεδομένου του $\theta \in \mathbb{R}^k$ που προκύπτει από την εκπαίδευση.

Class MyNeuralNetwork (Αρχείο ProjectNeuralNetwork)
Η κλάση αυτή υλοποιεί ένα νευρωνικό δίκτυο. Έχει κατασκευαστεί ώστε να υλοποιεί ένα νευρωνικό δίκτυο με 1 ή 2 ενδιάμεσα επίπεδα, μεταβλητό αριθμό από inputs και ενδιάμεσων κόμβων και μια επιστρεφόμενη τιμή ($output \in \mathbb{R}$).
Διαθέτει την δυνατότητα επιλογής ανάμεσα σε παλινδρόμιση (regressor) ή δυαδική ταξινόμιση (classifier με $output \in \{0, 1\}$). Την λειτουργία του νευρωνικού δικτύου ως classifier δεν θα την χρησιμοποιήσουμε στην εργασία, αλλά την προσθέσαμε δοκιμαστικά. Ως συνάρτηση ενεργοποιήσης μπορεί να χρησιμοποιηθεί μια από τις logistic, tanh, relu και στο τελικό επίπεδο δεν εφαρμόζεται καμία συνάρτηση ενεργοποίησης αν πρόκειται για παλινδρόμιση, ή εφαρμόζεται η logistic αν πρόκειται για ταξινόμηση. Ο αλγόριθμος εκμάθησης είναι ο αλγόριθμος μεγίστης κλίσης στην στοχαστική του μορφή με δέσμες. Δίνεται η δυνατότητα επιλογής του ρυθμού μάθησης, του tolerance, του μεγέθους της δέσμης, του μέγιστου αριθμού εποχών. Ο αλγόριθμος εκμάθησης σταματά αν το tolerance επιτευχθεί ή αν ξεπεραστεί ο μέγιστος αριθμός εποχών. Η συνάρτηση κόστους που χρησιμοποιείται για την παλινδρόμιση είναι το τεραγωνικό σφάλμα

$$J_\theta(y, \hat{y}) = \frac{1}{2n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2$$

και η συνάρτηση κόστους που χρησιμοποιήθηκε για την ταξινόμιση είναι η

$$J_\theta(y, \hat{y}) = -\frac{1}{n}\sum_{j=1}^{n} y_j \cdot \log \hat{y_j}$$

Η κλάση αυτή χρησιμοποιεί την κλάση Utilitites. Περιέχει τις μεθόδους:

- fit(self,X,y) που αναλαμβάνει την εκπαίδευση του μοντέλου. X είναι ο πίνακας με τα δεδομένα σε σειρές, δηλαδή $X \in \mathbb{R}^{n\times k}$ και $y \in \mathbb{R}^n$ το διάνυσμα με τις αναμενόμενες τιμές. Η διαδικασία αυτή ορίζει τα βάρη και τα bias ξεκινώντας αρχικά με τυχαία επιλογή τιμών για αυτά.

- predict(self,X) που αναλαμβάνει την πρόβλεψη του μοντέλου. X είναι ο πίνακας με τα δεδομένα σε σειρές, δηλαδή $X \in \mathbb{R}^{n\times k}$. Επιστρέφει το διάνυσμα $y \in \mathbb{R}^n$ προβλέψεις με βάση τα βάρη και τα bias που ορίστηκαν από την fit.

- weights_(self) που επιστρέφει τα βάρη του μοντέλου σε μια λίστα
- bias_(self) που επιστρέφει τα bias του μοντέλου σε μια λίστα
- getErrorRecord(self) που επιστρέφει το ιστορικό του σφάλματος κατά την εκπαίδευση.

Τα δεδομένα πρέπει να έχουν κανονικοποηθεί κατάλληλα για την χρήση στο μοντέλο.

Επιπλέον Λογισμικό
Χρησιμοποιήθηκε η βιβλιοθήκη sklearn και συγκεκριμένα οι κλάσεις sklearn.preprocessing, sklearn.linear_model, sklearn.neural_network

ΑΠΟΤΕΛΕΣΜΑΤΑ

Χρησιμοποιώντας τις προηγούμενης τιμής του Close
Αρχικά λαμβάνουμε τα δεδομένα στην μορφή:

	Date	Open	High	Low	Close	Volume
0	20/05/2016 16:00:00	31.20	32.11	31.08	31.96	134706794
1	27/05/2016 16:00:00	32.05	32.92	31.88	32.79	118988631
2	03/06/2016 16:00:00	32.84	33.15	32.57	32.86	112990084
3	10/06/2016 16:00:00	32.87	33.63	32.80	33.43	111092459
4	17/06/2016 16:00:00	33.18	33.34	32.26	32.42	137609612

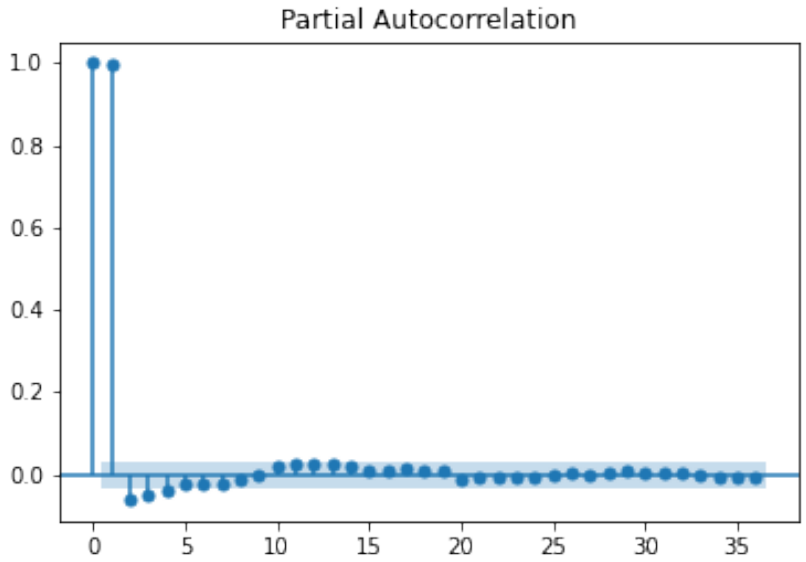
Σχήμα 2: Ενδεικτικό DataFrame με τα δεδομένα που λαμβάνουμε από την Google Finance

Σε πρώτο στάδιο ομαλοποιούμε την χρονοσειρά του close εφαρμόζοντας κάποιον από τους smoothers. Τότε παίρνουμε μια καινούρια χρονοσειρά με λιγότερα στοιχεία από την αρχική, λόγω της επίδρασης του κινητού μέσου όρου, αναλόγως με την τάξη του. Αυτό δεν μας επηρεάζει καθώς οι τιμές που αφαιρέθηκαν αντιστοιχούν στι αρχικές τιμές, και όχι στις πιο πρόσφατες (Όπως φαίνεται για παράδειγμα στο Σχήμα 1, οι ομαλοποιημένες χρονοσειρές ξεκινούν από την 50η μέρα και έπειτα όταν πρόκειται για εξομάλυνση τάξεως 50).

	Smoothed Close
0	31.9246
1	31.8942
2	31.8520
3	31.7924
4	31.7606

Σχήμα 3:

Προκειμένου να περάσουμε στην αυτοπαλινδρόμιση, χρησιμοποιήσαμε τον δείκτη $PACF(k)$ που δείχνει την άμμεση γραμμική συσχέτιση της Y_{t-k} με την Y_t για $k = 1, 2, \ldots$. Το συχνότερο είναι να παρατηρείται πως η άμμεση γραμμική επίδραση βρίσκεται στις 2 προηγούμενες τιμές.



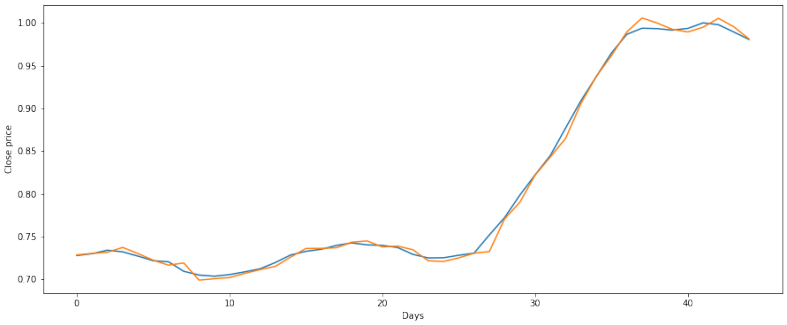
Σχήμα 4: Συνήθης εικόνα του δείκτη $PACF(k)$

Επομένως διαλέγουμε να κάνουμε αυτοπαλινδρόμιση τάξεως $k = 2$. Για να το κάνουμε αυτό φέρνουμε τα δεδομένα στην παρακάτω μορφή με την μέθοδο Utilities.formatData :

	t - 2	t - 1	t
0	31.9246	31.8942	31.8520
1	31.8942	31.8520	31.7924
2	31.8520	31.7924	31.7606
3	31.7924	31.7606	31.7378
4	31.7606	31.7378	31.6884

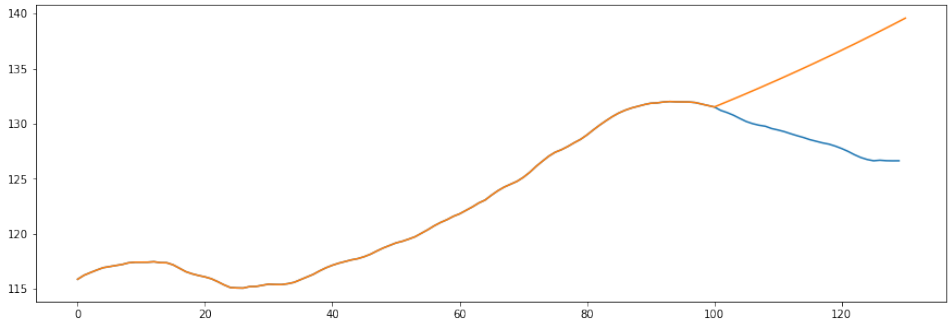
Σχήμα 5:

Για την αυτοπαλινδρόμιση χωρίζουμε τα δεδομένα κατά ένα ποσοστό για training και τα υπόλοιπα για testing. Η εκπαίδευση θα γίνει στις αρχικές τιμές και η επαλήθευση θα γίνει στις υπόλοιπες, δηλαδή θα σπάσουμε τα δεδομένα σε 2 μέρη με μη τυχαίο τρόπο.
Για την γραμμική παλινδρόμιση χησιμοποιούμε την myRegression και την sklearn. Δοκιμάζοντας διάφορες χρηματιστηριακές μετοχές από διάφορες τάξεις μεγέθους, φαίνεται πως η αυτοπαλινδρόμιση τάξης 2 έρχεται σχετικά κοντά στην πρόβλεψη της επόμενης ημέρας, ανγνωρίζοντας την τάση. Για παράδειγμα παίρνουμε μια εικόνα της μορφής:



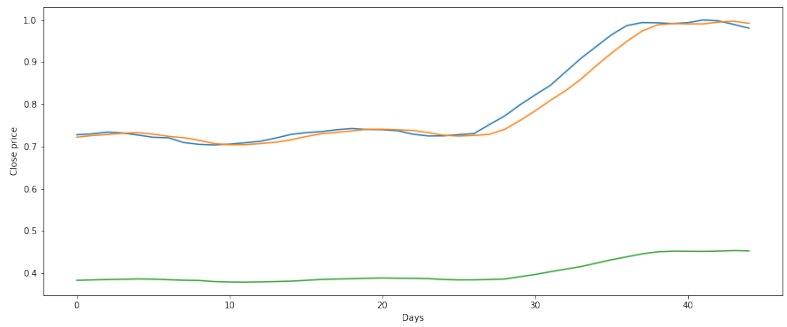
Σχήμα 6: Αυτοπαλινδρόμιση τάξης 2

Στην εικόνα με μπλέ χρώμα είναι η πραγματική τιμή και με πορτοκαλί είναι η πρόβλεψη των αυτοπαλινδρομικών μοντέλων. Τα μοντέλα myRegression και την sklearn συμπίπτουν. Βέβαια δεν φαίνεται να μπορούμε να χρησιμοποιήσουμε την πρόβλεψη της αυτοπαλινδρόμισης για προέκταση της χρονοσειράς για περισσότερες μέρες (δηλαδή λαμβάνοντας την πρόβλεψη της Y_{t+1} ημέρας ως δεδομένη για να προβλέψουμε της Y_{t+2}) διότι το μοντέλο θα διατηρήσει την τελευταία τάση που αναγνώρισε και θα συνεχίσει γραμμικά από εκεί σαν μια ευθεία γραμμή.



Σχήμα 7: Παράδειγμα Προσπάθειας Επέκκτασης της Χρονοσειράς

Δοκιμάζουμε τώρα την ίδια διαδικασία με το νευρωνικό δίκτυο. Αυτήν την φορά δοκιμάζουμε διάφορα σενάρια για το πόσες θα είναι οι προηγούμενες τιμές που θα χρησιμοποιήσουμε για την πρόβλεψη, αλλά πάντα στα πλαίσια των 2 με 7 ημερών.
Η ιδέα αυτή δεν φαίνεται να λειτουργεί με νευρωνικά δίκτυα. Το νευρωνικό δίκτυο της sklearn φαίνεται να μπορεί σε λίγες περιπτώσεις να προβλέπει ως ένα βαθμό τις μελλοντικές τιμές, όμως το νευρωνικό δίκτυο MyNeralNetwork2 δεν κάνει καλή προσέγγιση.



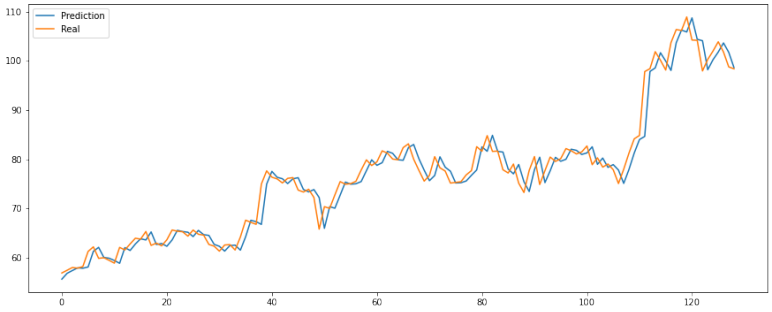
Σχήμα 8: Νευρωνικά Δίκτυα

Στην παραπάνω εικόνα, μπλε είναι η πραγματική τιμή, πορτοκαλί είναι η τιμή της sklearn, και πράσινη είναι η τιμή της MyNeuralNetwork. Και αυτή είναι μια συμπεριφορά που παρατηρείται μεμονωμένα σε αυτό το παράδειγμα. Σε πολλές περιπτώσεις η sklearn αποτυγχάνει σε μεγάλο βαθμό επίσης, και δεν

έχουμε βρει κάποιο σύνολο παραμέτρων (ρυθμός μάθησης, αριθμός κόμβων κλπ) για το οποίο παρατηρείται καλή συμπεριφορά ομοιόμορφα στα παραδείγματα.

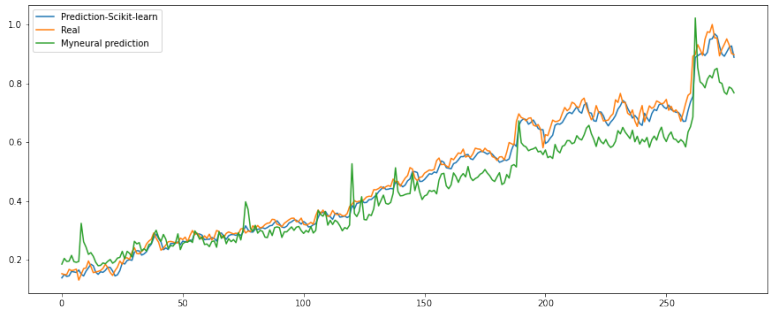
Χρησιμοποιώντας τις άλλες χρηματιστηριακές τιμές

Αυτή τη φορά χρησιμοποιούμε το DataFrame που φαίνεται στο Σχήμα 2. Κάνουμε πρώτα γραμμική παλλινδρόμηση με χρήση της myRegression και της sklearn. Και σε αυτήν την περίπτωση οι προβλεψεις της myRegression και της sklearn συμφωνούν ακριβώς και μπορούν να προβλέπουν σχετικά ικανοποιητικά την πορεία της χρονοσειράς.



Σχήμα 9: Συμπεριφορά της Γραμμικής Παλινδρόμισης

Αυτή την φορά και τα νευρωνικά δίκτυα φαίνεται να έχουν πιο ομαλή συμπεριφορά, χρησιμοποιώντας 200 κόμβους και ρυθμό μάθησης 0.0001.



Σχήμα 10: Συμπεριφορά των Νευρωνικών Δικτύων

ΣΥΜΠΕΡΑΣΜΑΤΑ

Εν γένει, όσον αφορά το πολύ βραχυχρόνιο επίπεδο του να προβλέπει κανείς μια τιμή για την επόμενη χρονική στιγμή φαίνεται πως το απλό μοντέλο της γραμμικής παλινδρόμισης θα δώσει μια σχετικά αξιόπιστη αναμενόμενη τιμή, η οποία ουσιαστικά δίνει την αναμενόμενη πορεία της τάσης της χρηματιστηριακής τιμής. Τα νευρωνικά δίκτυα φαίνεται να μην λειτουργούν ίσως διότι η πληροφορία που λαμβάνουν δεν είναι αρκετή για να εντοπίσουν κάποιου είδους pattern στα δεδομένα. Πολύ συχνά εμφανίζεται και το πρόβλημα του overfitting. Δηλαδή τα νευρωνικά δίκτυα δεν φαίνεται να είναι κατάλληλο εργαλείο για τέτοιου είδους ανάλυση.
Στα πλαίσια της εργασίας, πέρα από την διερεύνηση που κάναμε, ένα μεγάλο μέρος ήταν η υλοποίηση των εργαλείων που θα χρησιμοποιούσαμε. Δηλαδή η καταγραφή όλων των εργαλείων που θα χρειαστούμε, η μελέτη των θεωρητικών μοντέλων και η υλοποίησή τους με έναν ολοκληρωμένο τρόπο. Χρειάστηκε οργάνωση και καλή συνεργασία προκειμένου να μπορούμε σαν ομάδα να χρησιμοποιούμε τον κώδικα

που αναπτύξαμε, και ήταν απαραίτητος ο προσεκτικός σχεδιασμός πριν την υλοποίηση. Το πιο μεγάλο κομμάτι ήταν η υλοποίηση του νευρωνικού δικτύου, για το οποίο προσπαθήσαμε να δώσουμε όσο πιο πολλές δυνατότητες γίνεται στον χρήστη, προκειμένου να έχει την ελευθερία να αλλάζει τις παραμέτρους κατάλληλα, αναλόγως με τις απαιτήσεις του.
Έχοντας κάνει αυτήν την διερεύνηση, μια άλλη διερεύνηση που θα ήταν ενδιαφέρουσα είναι κάποιου είδους ταξινόμηση κατά την οποία ένα μοντέλο θα μπορεί να προτείνει ποιά θα είναι η πορεία της τιμής (ανοδική, καθοδική, σταθερή) λαμβάνοντας υπόψη στοιχεία όπως την πορεία της τιμής στις προηγούμενες μέρες, την πορεία τιμών άλλων χρηματιστηριακών μετοχών με τις οποίες μπορεί να σχετίζεται, ή ακόμη και του τίτλους των ειδήσεων της ημέρας.

ΠΑΡΑΔΟΤΕΑ

Παραδίσουμε με αυτήν την αναφορά 3 προγράμματα python:

- ProjectRegression.py που αφορά την κλάση MyRegression
- ProjectUrilities.py που αφορά την κλάση Utilities
- ProjectNeuralNetwork.py που αφορά την κλάση MyNeuralNetwork

ένα αρχείο Project_Info.txt που περιέχει οδηγίες για τον τρόπο που χρησιμοποιούνται τα προγράμματα
ένα Jupyter Kernel με όνομα Project_Example.ipynb το οποίο έχει ένα παράδειγμα για τον τρόπο που λειτουργούν τα προγράμματα
ένα αρχείο Crocs.csv που περιέχει ενδεικτικά δεδομενα για το Project_Example.ipynb

ΒΙΒΛΙΟΓΡΑΦΙΑ

- https://www.google.com/finance
- Hastie, T.; Tibishirani, R.; Friedman, J. The Elements of Statistical Learning; 2017; pp. 1–764 (https://web.stanford.edu/~hastie/ElemStatLearn//printings/ESLII_print10.pdf)
- Κωσταντίνος Σμαραγδάκης: Διαλέξεις Περιγραφικής Στατιστικής (https://kesmarag.gitlab.io/descriptive-statistics/calendar.html)
- Β. Κομπουλάζος, Γ. Παπακώστας, Εισαγωγή στην Υπολογιστική Νοημοσύνη, Κάλλιπος 2015 (https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/3443/1/1253_%ce%9a%ce%91%ce%9c%ce%a0%ce%9f%ce%a5%ce%a1%ce%9b%ce%91%ce%96%ce%9f%ce%a3_%cf%84%ce%bf%ce%92%ce%b9%ce%b2%ce%bb%ce%af%ce%bf-KOY.pdf)
- https://www.investopedia.com/articles/active-trading/052014/how-use-moving-average-buy-s asp
- Time Series Forecasting using ARIMA, Neural Networks and Neo Fuzzy Neurons (https://www.researchgate.net/publication/255624538_Time_Series_Forecasting_using_ARIMA_Neural_Networks_and_Neo_Fuzzy_Neurons)
- Hands-On Machine Learning with Scikit-Learn and TensorFlow, Aurélien Géron http://index-of.es/Varios-2/Hands%20on%20Machine%20Learning%20with%20Scikit%20Learn%20and%20Tensorflow.pdf