Regressore Matematico Semplice e Complesso in C

Che cos'è un Regressore Lineare?

Un **Regressore Lineare è un algoritmo di apprendimento automatico** sul quale si basa l'intera disciplina del Machine Learning e tutti quei algoritmi paragonabili all'intelligenza artificiale.

Il funzionamento principale di un regressore matematico è quello di, "**apprendere**", **tramite l'acquisizione di variabili indipendenti(x) e le loro relative variabili dipendenti (y)**, la funzione logica che relaziona questi dati.

In matematica siamo abituati a dare un valore x ad una funzione e ricevere in output un valore Y in base alla x che abbiamo fornito. Se il nostro compito è quello di trovare una y per ogni x, **il compito di un regressore è quello fornirci la funzione che relaziona x e y** che gli abbiamo fornito in precedenza.

Il vantaggio più grande che fornisce il regressore si verifica **dopo l'apprendimento**. Infatti una volta che l'algoritmo sarà riuscito a comprendere la logica e la funzione che relaziona i nostri dati, sarà anche in grado di "**prevedere**", ogni volta che lo interrogheremo, il valore di y per una x che non abbiamo nemmeno **mai inserito**.

Tutto ciò è in grado di farlo calcolando la così detta **Funzione Regressiva Lineare.**

Cosa si intende per "Regressione Lineare"?

Quando si ha a che fare con la matematica analitica dei dati, ci si imbatte nella maggior parte dei casi, in dati e campioni messi in relazioni tra loro tramite determinate logiche e funzioni matematiche. Il problema più grande, però, è quando i dati che si vanno ad analizzare si basano **su fenomeni reali e non funzionalmente perfetti.** Ad esempio, la relazione che incorre tra l'altezza di un padre e l'altezza del relativo figlio.

Apparentemente questi due dati possono sembrare in realtà casistiche che non seguono una precisa logica e che non hanno una relazione tra di loro, ma per questo motivo vengono calcolate le **Regressioni Lineari**.

Infatti la prima volta che fu stato utilizzato il termine "Regressione Lineare" accadde a fine ottocento dal biologo inglese Galton. Quest'ultimo decise, come nell'esempio, di raccogliere l'altezza di genitori e figli notando l'esistenza di una relazione tra le due altezze**. Comprese che più alti erano i genitori, più alti erano pure i figli.**

Tuttavia, trattandosi di dati "**reali**", non sempre la relazione teorica **veniva rispettata** e capitava talvolta che nonostante i genitori fossero molto alti, i figli non erano per nulla paragonabili.

Nonostante ciò, la relazione "illegale" seguiva, se pur con scarsi risultati, la **Funzione Regressiva** di tutte le altezze spostandosi (**regredendo**) verso la media.

Che cos'è la funzione regressiva lineare?

La funzione Regressiva Lineare non è altro che, in termini matematici, la **funzione lineare** generata con un **coefficiente angolare statistico** e una **intercetta approssimativa. La funzione rappresenta una retta che** è in grado di generare il minor numero di imprecisioni e incertezze quando si vanno a interrogare i dati relazionati, rispetto a quelli già preposti e raccolti.

Che differenza c'è tra un classico regressore e il mio?

La differenza è abbastanza sottile in realtà. **Tutto cambia nella metodologia di allenamento**: il classico regressore utilizza la tecnica dei "**Minimi quadrati**" per trovare la retta regressiva, invece il mio è basato sul **comportamento e l'andamento delle varie relazioni all'interno del sistema che si sta analizzando**. Il metodo tradizionale produce una linea regressiva perfetta che **basata sull'errore minimo generato**, invece la mia retta regressiva, come si potrà vedere durante la spiegazione dell'algoritmo, tiene conto anche la **direzione delle relazioni**, includendo così nell'allenamento, **l'imprecisione dei dati**.

Algoritmo Regressore Matematico Semplice

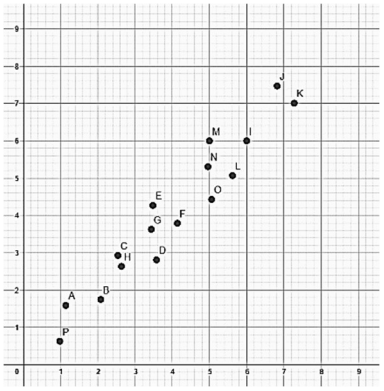
1. Generare il dataset

Il primo step per creare un regressore è senza dubbio quello di ottenere tutti i dati necessari all'apprendimento del **Modello.**

Definiamo **dataset** l'insieme di relazioni che il modello utilizzerà per **addestrarsi**.

Bisogna necessariamente definire le strutture dati dove salvare le varie informazioni come le coordinate y e x di un punto nel grafico. Quindi, è opportuno tenere sempre ordinato e ben gestito lo spazio dedicato alle y e alle x.

2. Manipolazione dei dati



Siccome il computer lavora soltanto in maniera analitica, non può certo utilizzare e maneggiare dati non ordinati in struttura, di conseguenza è **necessario** che almeno per le x, tutte le relazioni siano in ordine crescente.

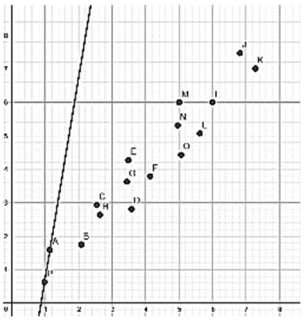
Inoltre, è opportuno tenere in considerazione che meno dati si hanno a disposizione, meno precisa sarà la predizione del regressore.

Se la fase di manipolazione dati viene effettuata con cura e attenzione, anche se non letteralmente, **il computer potrà lavorare e calcolare come se vedesse dei puntini su un grafico logico e strutturato in memoria.**

3. Allenamento modello semplice

Data la funzione $y = mx + q$, si definisce **m** il coefficiente angolare della funzione e **q** l'intercetta.

Definiamo in oltre **Allenamento,** il processo logico-matematico che permette di **modellare** la nostra funzione lineare generica.



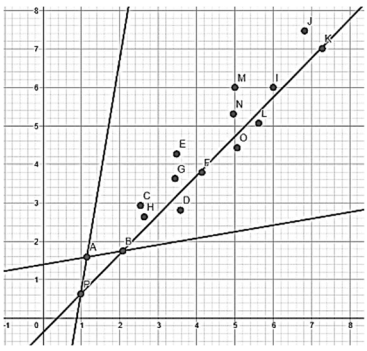
Per generare la funzione regressiva è necessario prima calcolare le **sub-funzioni** che mettono in relazione tutti i punti in modo **perfetto**. In parole povere**, bisogna creare ogni funzione possibile con tutti i punti a disposizione**.

Si può calcolare la funzione di una retta che passa tra due punti in due step:

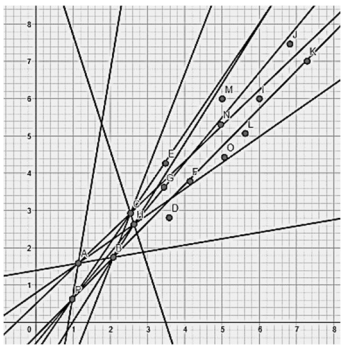
-**Step 1**: **Calcolo della pendenza**, tale grazie alla formula $\boldsymbol{m = \frac{\Delta y}{\Delta x} = \frac{yb - ya}{xb - xa}}$

-**Step 2**: Risoluzione dell'equazione di primo grado $\boldsymbol{q = y - mx}$ con la quale si può trovare l'ultima incognita della nostra sub-funzione.

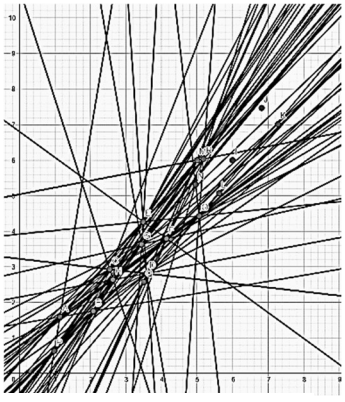
Una volta calcolata la nostra sub-funzione, si manda immediatamente in memoria e si passa alla prossima ripetendo questo procedimento, ricordando però che, **per ogni punto, bisognerà calcolare una sub-funzione che passa per ogni punto precedente** quindi se il punto 2 si calcola col punto 1, il punto 3 si calcolerà con il punto 1 e il punto 2 così via fino creare tutte le funzioni possibili.



1

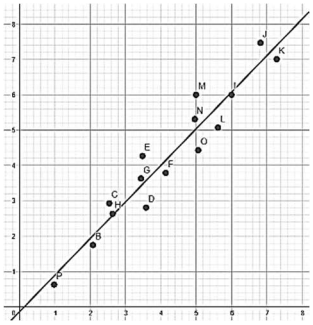


2



3

4. Ottenere la funzione lineare regressiva



Dopo aver calcolato quel numero esponenziale di sub-funzioni, il regressore deve, detto in ambito di Machine Learning, **Allenare il modello,** dove per modello si intende infatti il prodotto finale nonché la funzione regressiva che si stava cercando.
Per fare ciò, il programma accede a tutte le strutture dati Funzione e calcola banalmente la media aritmetica di tutte le pendenze e le intersezioni che ha trovato e calcolato:
Una volta che avrà trovato la funzione Regressiva, si dice che il modello sarà **Allenato**.
Senza dubbio il suo allenamento è basato sulla quantità e la "qualità dei dati", quindi, più i dati su cui si è allenato sono imprecisi, più sarà imprecisa la sua previsione. Anche la quantità è importante, dato che se gli forniamo poche relazioni, il modello avrà un alto tasso di **errore.**
Alla fine potremo ridurre l'**allenamento** nella **formula** matematica qua affianco dove $\hat{m}$ e $\hat{q}$ sono i rispettivi valori teorici della pendenza e intercetta della retta regressiva.

$$\boldsymbol{nsubfunz} = \sum_{i=2}^{npunti} (\boldsymbol{i-1})$$

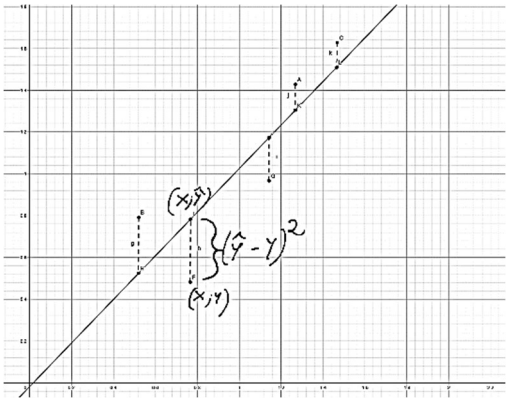
$$\hat{\boldsymbol{m}} = \frac{\sum_{n=2}^{nsubfunz}\left(\sum_{i=1}^{n-1}\left(\frac{y_n - y_{n-i}}{x_n - x_{n-i}}\right)\right)}{\boldsymbol{nsubfunz}}$$

$$\hat{\boldsymbol{q}} = \frac{\sum_{n=2}^{nsubfunz}\left(\sum_{i=1}^{n-1}\left(\left(\frac{y_n - y_{n-1}}{x_n - x_{n-1}}\right)x_n - y_n\right)\right)}{\boldsymbol{nsubfunz}}$$

5. Errore di imprecisione

E' naturale avere un **margine d'errore**, soprattutto quando si sta lavorando con una funzione regressiva modellata da punti e sub-funzioni di **scarsa qualità analitica** (che quindi forniscono risultati molto incerti).
Per questo è necessario tenere conto dell'errore che la funzione porta con se'.
A livello grafico, **l'errore non è altro che la differenza dei quadrati delle y del punto d'allenamento e del punto teorico** (punto fornito dalla retta regressiva avendo stessa X del punto di allenamento).

$\boldsymbol{Errore = (\hat{y} - y)^2}$**. Dove $\hat{y}$ è il valore teorico e y il valore reale**



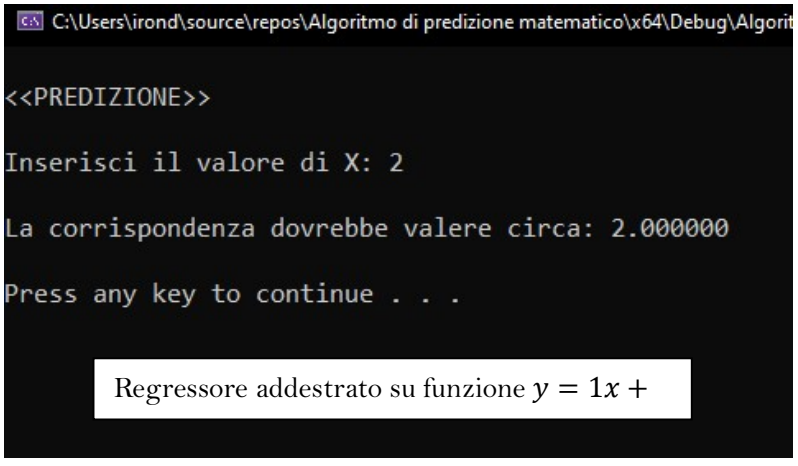
E' importante che sia tutto elevato al quadrato per il semplice fatto che il calcolo potrebbe essere intralciato da valori della y che potrebbero risultare negativi (ponendo tutto al quadrato, è come se si mantenesse l'errore costantemente positivo).
Per calcolare **l'imprecisione** bisognerà evolvere la formula in:

$$\boldsymbol{Imprecisione} = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\boldsymbol{n}}$$

Infatti, per ottenere **l'imprecisione indicativa** della retta regressiva semplice, l'algoritmo dovrà **sommare l'errore di n punti ed infine dividerlo per n punti stesso**. In questo modo si ottiene un indice d'errore indicativo che permette di valutare l'efficacia della regressione creata.

6. Regressore Matematico Lineare semplice creato

A questo punto il nostro regressore semplice **è stato allenato** ed è pronto per essere utilizzato. L'utilità come ho già detto prima, è quella di ottenere una **"predizione"** aritmetica di ogni x che inserisco in base ad una funzione appresa dal modello.
Il problema principale di un regressore lineare è appunto il fatto che sia lineare e basta. Lo scopo del mio programma è quello di andare oltre alla teoria e cercare di raggiungere il valore reale che potrebbe essere. Per questo ho creato il **regressore Matematico "Complesso".**

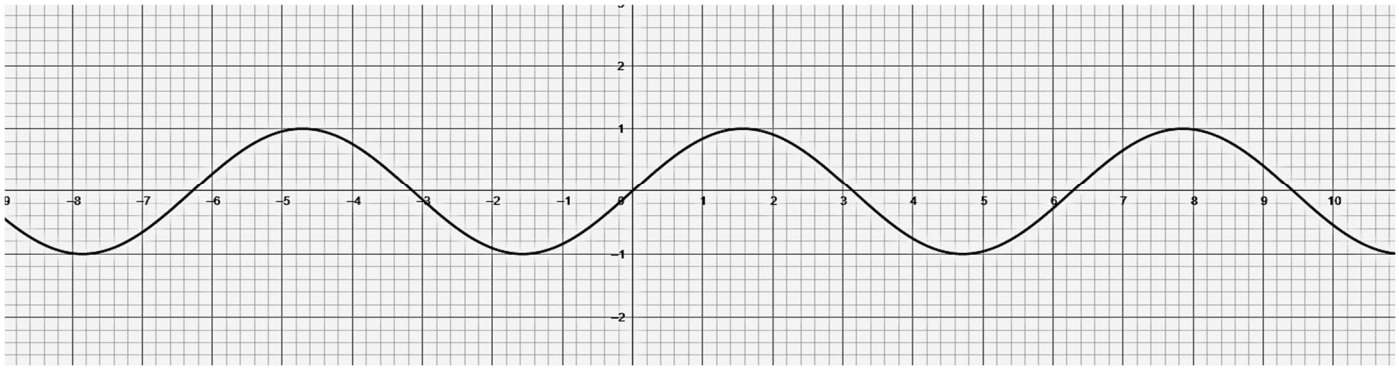


Algoritmo Regressore Matematico "Complesso"

Cosa cambia?

Lo scopo di questo algoritmo è quello di "aggiustare il tiro" del Regressore Semplice aggiungendo una sorta di "**Oscillazione d'errore**". Se infatti l'idea principale del regressore semplice è quello di ottenere una linea dritta e assolutamente poco efficiente che mi interroghi le y in modo lineare, l'idea del Regressore Complesso è quello di fornire alla mia linea una distorsione. Questa distorsione sarà in grado di fornire **Ampiezza** e **Frequenza** che aiuteranno a rendere più preciso l'allenamento del Regressore.

Per fare ciò sfrutteremo la funzione seno ovvero $\boldsymbol{y = \sin(x)}$**;**

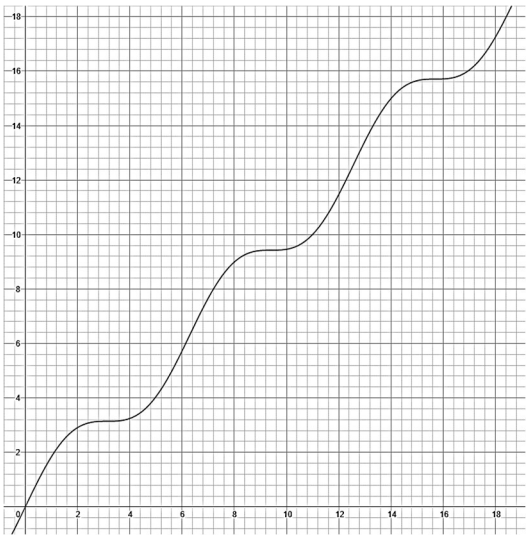


Da questo momento in poi, ci riferiremo alla nuova versione oscillante della funzione con il nome di "**Magnitudo**"

Infatti come si può ben vedere la funzione **y=sin(x) ha delle proprietà ottime per generare ampiezza e frequenza** in una linea. Per parametrizzare la funzione e, quindi, fornirgli le variabili la funzione si dovrebbe riscrivere sotto forma di $\boldsymbol{y = \rho \sin(\omega x)}$.

Dove il parametro "$\boldsymbol{\rho}$" indica il valore di **ampiezza** e "$\boldsymbol{\omega}$" il valore della **frequenza** che deve avere l'andamento**. Inoltre per fornire una inclinazione alla funzione seno, bisognerà aggiungere un incremento variabile X**, trasformando la funzione di prima in:

$$\boldsymbol{y = \rho \sin(\omega x) + x}$$



Infatti al variare della x, il risultato della funzione seno sarà incrementato.

Per fornire invece **una pendenza diversa da 1,** bisognerà aggiungere un **coefficiente angolare alla x** trasformando nuovamente la funzione in:

$$\boldsymbol{y = \rho \sin(\omega x) + mx}$$

In questo modo l'inclinazione della funzione sarà variabile e parametrizzabile nell'algoritmo successivo.

Vedremo successivamente come in realtà la funzione **si evolverà ulteriormente** e si aggiungeranno altri parametri importanti per l'apprendimento del modello.

La mia idea originale era quella di **ruotare rigidamente la funzione seno** secondo il sistema:

$$\begin{cases} \boldsymbol{x_{rot} = x\cos(a) - \sin(x)\sin(a)} \\ \boldsymbol{y_{rot} = x\sin(a) + \sin(x)\cos(a)} \end{cases}$$

Il grosso problema di questa funzione era il fatto che l'interrogazione della **Xrot** era impossibile da risolvere analiticamente e sarebbe servito un algoritmo ricorsivo che andasse a testare combinazioni di **x** e **y** ma ciò non sarebbe stata una soluzione efficacie poiché l'allenamento del modello avrebbe richiesto un tempo troppo eccessivo per essere completato, quindi l'idea della funzione seno rigidamente ruotata era inefficiente.

Adesso è il momento di vedere in che cosa consiste il nuovo algoritmo di **Regressione Complesso**.

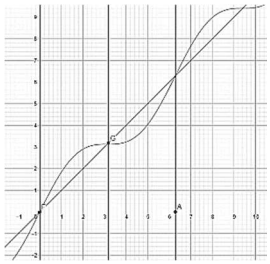
7. Costruire la funzione seno sulla funzione lineare

Il primo passo per il miglioramento dell'algoritmo è senza dubbio quello di impostare sopra la lineare la funzione seno.
Per fare ciò, è necessario inserire altri due parametri alla nostra magnitudo e questi due sono lo **Spostamento** e l'**Intercetta.**
Trasformando la funzione $\boldsymbol{y = \rho \sin(\omega x) + mx}$ in $\boldsymbol{y = \rho \sin(\omega x + s) + mx + q.}$
Per costruire sopra la funzione regressiva lineare questa versione della funzione seno, bisognerà assegnare la medesima pendenza (**m**) e medesima intercetta (**q**) della funzione regressiva.

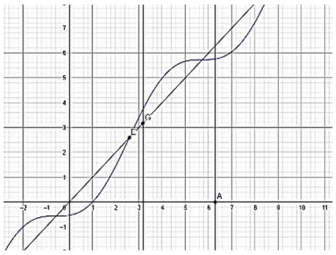
8. Calcolo spostamento massimo

Lo spostamento è forse uno dei parametri più importanti della magnitudo ed è importante prima comprenderne la natura.

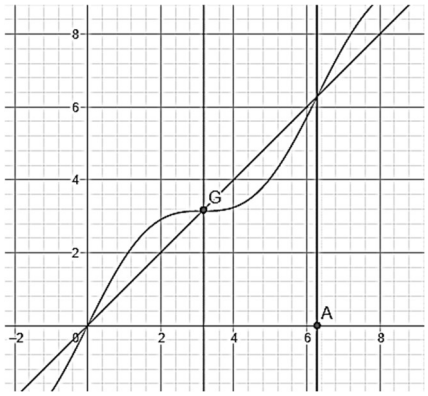
Il parametro di spostamento qui è settato a 0. Nel frattempo vengono messi dei punti di riferimento corrispondenti alle perpendicolari all'asse x che intersecano con il **mezzo periodo** e il **periodo** della funzione →



A questo punto si inizia a variare il parametro e si può notare come lo spostamento **avvenga seguendo la funzione lineare** avente stessa pendenza e intercetta (regressiva in questo caso) →



Come c'era s'aspettarsi, dopo che il parametro di spostamento avesse raggiunto il medesimo valore del periodo, **la funzione si è presentata nella stessa posizione di partenza** nonostante sia stata spostata.



Quindi è inutile far variare lo spostamento più del dovuto, perché non cambierebbe nulla, dunque l'unica soluzione è quella di assegnare uno **spostamento massimo** che equivale al **periodo della funzione**.
Per calcolarlo basta applicare la formula del calcolo del periodo della funzione seno:

$$\boldsymbol{SpostMax = T = \frac{2\pi}{|\omega|}}$$

Dove omega corrisponde alla **frequenza della magnitudo**.

Un trucco matematico per avere sempre il valore assoluto di un numero, quando si parla d'informatica, **è quello di calcolare la radice quadrata del quadrato del numero stesso** $\boldsymbol{\sqrt{n^2}}$ in questo modo (almeno nel contesto aritmetico) **si verrà sempre a considerare il valore positivo del numero**.

9. Calcolo ampiezza massima e ampiezza minima

Anche l'ampiezza è senza dubbio un parametro aggiuntivo molto importante che permette di generare l'oscillazione tra le relazioni ma **è necessario contenerne la variazione**. Infatti se l'ampiezza fosse esageratamente grande, **il rischio di sbagliare la predizione aumenterebbe** anziché di rendere la regressione più precisa quindi, bisogna calcolarne un'ampiezza massima e una minima.
L'ampiezza varia da **-n** a **+n** dove n è l'errore massimo generato da un punto (dato prima dell'allenamento) rispetto alla **Funzione Lineare regressiva.**
Con le formule scritte in precedenza si calcola ogni distanza del punto dalla retta e si salva la più grande. In questo modo si avrà impostato l'ampiezza massima.

```
double ampmax= AMPMASSIMA(finale,a,np);
```

10. Algoritmo di selezione

Questa è la fase **più importante** e la **più decisiva** dell'algoritmo del Regressore Complesso.
Ciò che si andrà a fare durante questa fase è il vero e proprio **allenamento del modello**, dove l'algoritmo inizierà a testare ogni configurazione possibile da applicare alla nostra funzione.

$$y = \rho \sin(\omega x + s) + mx + q$$

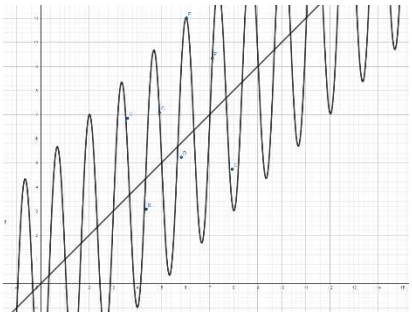
Il compito del regressore è semplice ora:
Per ogni frequenza applicata, sarà applicato uno **spostamento** e per ogni spostamento verrà settata anche una **ampiezza**, per semplicità, denominiamo i parametri.

$$y = amp \sin(frqx + spost) + pendx + interc$$

Facendo in questo modo, verrà sempre a crearsi una magnitudo unica nel suo genere che verrà ovviamente subito **testata.** L'algoritmo di selezione, come suggerisce il nome, ha lo scopo di selezionare la magnitudo testata che ha prodotto il **miglior risultato** dove con **Miglior risultato** si intende il "**punteggio errore**" calcolato quando viene comparata con le relazioni di allenamento (i punti).
Per fare ciò ricordiamo la formula per calcolare l'imprecisione e che si trasforma con gli input della magnitudo.

$$Imprecisione = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n} \quad \rightarrow \quad Imprecisione = \frac{\sum_{i=1}^{n} ((amp \sin(frqx + spost) + pendx + interc) - y_i)^2}{n}$$

Calcolata l'imprecisione, abbiamo ottenuto **una parte del punteggio**. Infatti per ottenere il "**punteggio d'errore**" bisogna applicare una ulteriore modifica, ovvero aggiungere ciò che ho definito **incertezza** nel regressore.
Infatti, anche se la configurazione ha ottenuto un buon punteggio iniziale, bisogna tenere conto di come lo ha raggiunto. (ricordo infatti che lo scopo del mio regressore è quello di ottenere una predizione che sia un buon compromesso con l'andamento dell'errore e la predizione corretta)



Serve notare come un grafico con un **alto tasso di ampiezza e frequenza è effettivamente sempre quello migliore** ma in un certo senso lo è "**barando**": una frequenza e ampiezza sballatissima **è potenzialmente capace di raggiungere perfettamente tutti i punti** di allenamento senza mantenere l'andamento quasi lineare che si sta cercando, creando relazioni per nulla simili a quelli della regressione come nell'immagine.
Quindi è necessario **aggiungere l'incertezza** capace di creare un **equilibrio tra oscillazione e la linearità** che equivale a:

$$Incertezza = (|(amp + frq) \times npunti|)^2$$

Si moltiplica la somma di ampiezza e **frequenza per il numero di punti** per arrivare al **compromesso** che più punti ci sono, meno ampiezza e frequenza **è accettata valida** ma al contrario, quando invece ci sono meno relazioni, l'andamento oscillante della magnitudo sarà più tollerato.
La formula del punteggio sarà allora la somma dell'imprecisione e dell'incertezza.

$$Punteggio\ errore = Imprecisione + Incertezza$$

Arrivato a questo punto, il regressore avrà individuato la configurazione con il **Punteggio errore** più basso e l'avrà selezionata generando il modello.

11. Regressore Matematico Lineare Complesso creato

Il regressore adesso è **allenato e perfettamente utilizzabile**. Per la predizione ovviamente sfrutterà la funzione della magnitudo che ha generato dopo l'allenamento. In realtà il programma chiederà in anticipo, prima dell'allenamento, se si vuole eseguire un allenamento **PRECISO** o **SUPER PRECISO**. Difatti la differenza è la sensibilità del campionamento che verrà effettuato ovvero, se si sceglierà Preciso, i valori di spostamento, frequenza e ampiezza **varieranno da quello successivo di 0.1 unità, invece il super preciso di 0.01 unità.**
Il guadagno del super preciso è la precisione, invece **la perdita** è l'aumentare del tempo d'allenamento (**10 volte più lento rispetto al preciso**).

Guida all'utilizzo del regressore

1. Crea nuova relazione

La prima opzione fornita dal regressore è ciò che permette di acquisire in input le relazioni d'allenamento.
Inizialmente sarà richiesta la variabile indipendente (x) e successivamente il regressore chiederà la variabile dipendente (y) associata.
NOTA: si ricorda che è un regressore che allena il modello su punti teoricamente appartenenti ad una proporzione lineare quindi, nel caso si inserissero punti di funzioni esponenziali, funzione seno o altri, la predizione fallirebbe.

2. Visualizza le relazioni

Permette di visualizzare tutte le relazioni inserite nel dataset.

3. Cancella una relazione precisa

Permette di visualizzare tutte le relazioni inserite e scegliere una relazione specifica da cancellare dal dataset.
Inserendo -1, la cancellazione verrà annullata.

4. Resetta dataset

Permette di eliminare completamente tutte le relazioni presenti nel dataset.

5. Allena modello

Opzione che permette di allenare effettivamente il modello lineare.
Verrà chiesto, prima dell'allenamento, di specificare il livello di precisione. A prescindere dalla decisione, il modello verrà allenato sia su modello di regressore semplice e complesso. L'unica cosa che varia, è la qualità di campionamento (indicato nel punto 11 dell'algoritmo).

6. Imprecisione lineare

Permette di visualizzare il numero di unità che differiscono il valore y delle relazioni fornite per l'allenamento e la y teorica dalla quale si ricaverebbe la predizione. Queste imprecisioni si basano sul modello lineare semplice.

7. Predizione

Permette di ottenere la predizione dopo aver allenato il modello.
La predizione fornisce il risultato teorico della funzione lineare regressiva (regressore semplice) e il risultato teorico basato sull'oscillazione d'errore (regressore complesso).

8. Andamento logico funzionale

Permette di visualizzare l'andamento delle y su una successione di x interrogate in automatico.
Prima di tutto, verranno richieste il numero di predizione che si vorranno effettuare e successivamente il regressore chiederà l'intervallo di interrogazione delle X nonché di quanto varieranno l'una dalle altre.
Una volta inseriti i dati, il regressore interrogherà in automatico il proprio modello e restituirà tutte le predizioni.

9. Informazioni sul modello

Permette di visualizzare il valore di tutti i parametri del modello e della funzione magnitudo di cui ho parlato sopra. Inoltre vengono mostrate ulteriori informazioni come numero di tentativi, punteggio errore e livello di precisione.

10. Chiudi

Termina l'esecuzione del programma.

NOTA: l'autenticità del software è garantita dalla costante presenza del Cognome e Nome Scapellato Davide all'interno dell'interfaccia utente del programma.

11. Per provare…

Se si vuole testare il regressore, consiglio di inserire, ad esempio, relazioni sul moto rettilineo uniforme nonché il classico esperimento che si svolge in un laboratorio di fisica.
Il tempo va inserito nelle variabili indipendenti x invece lo spostamento totale nelle y. Una volta allenato il modello, il regressore in base al tempo inserito, riuscirà a prevedere quanta distanza ha percorso l'oggetto che si sta testando.

Se si vuole inserire una grande quantità di dati:

1. Chiudere il programma se in esecuzione.
2. Andare su Excel e scrivere nella colonna A il valore delle X e nella colonna B la Y associata.
3. Una volta fatto ciò basterà fare un copia e incolla all'interno di un file di testo .txt classico.
4. Fare in modo che ci sia soltanto una linea vuota alla fine delle relazioni nel file di testo, non prima o nel mezzo delle relazioni.
5. A questo punto rinomina il file in "dati.txt" e salvalo nella stessa cartella dell'eseguibile.
6. E' importante che il nome del file sia come scritto prima, in caso contrario il file non sarà rilevato.

In caso di successo, al momento del lancio in esecuzione del programma, tutte le relazioni saranno caricate in automatico.