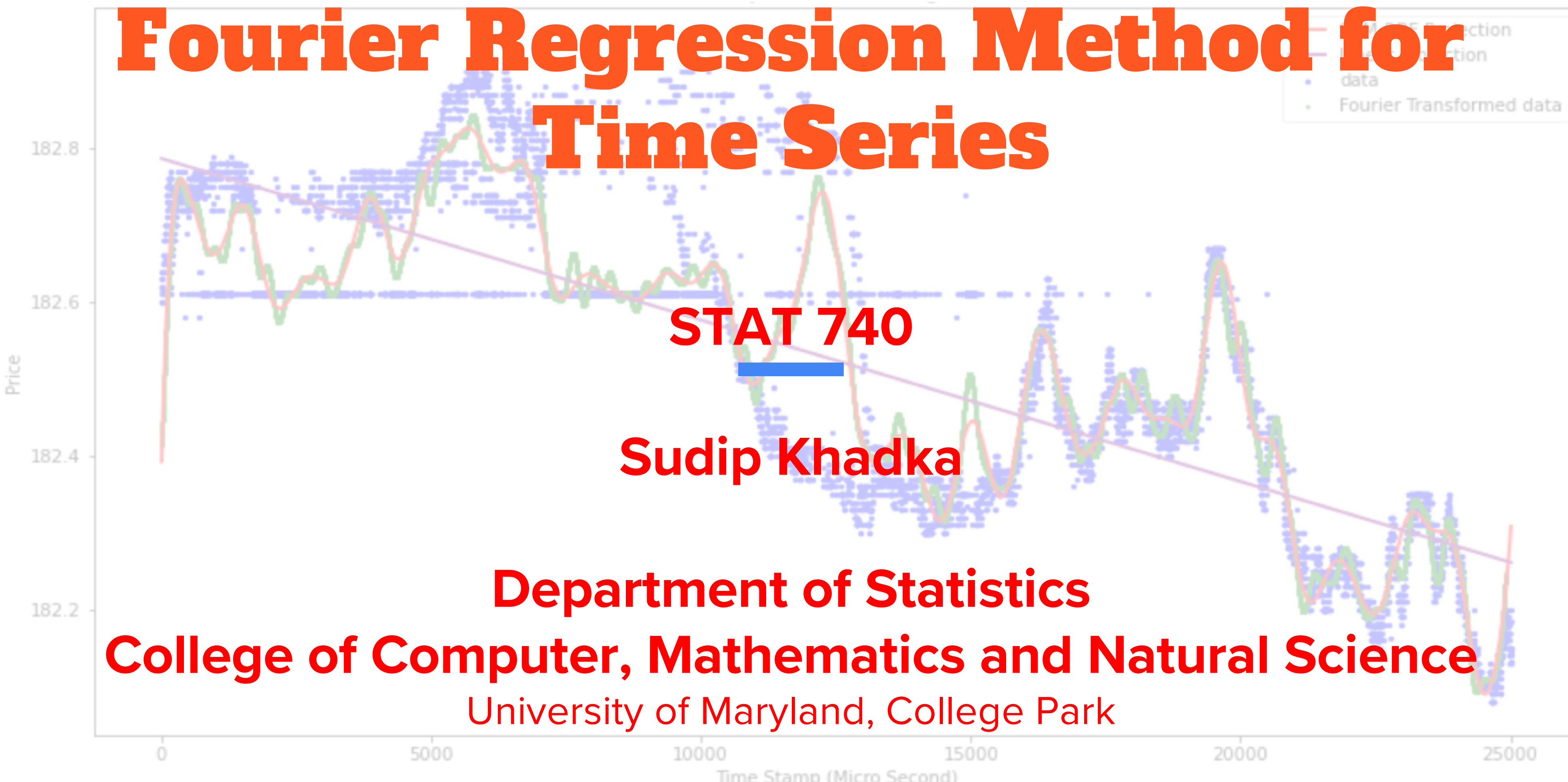
# Fourier Regression Method for Time Series




## STAT 740


**Sudip Khadka**

**Department of Statistics**

**College of Computer, Mathematics and Natural Science**

University of Maryland, College Park

# Objectives:

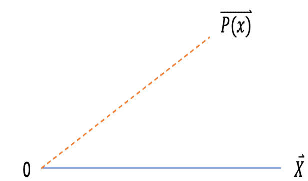
- Dataset is the high frequency price of some asset collected from several exchanges .

- Predicting Price onto time:
  - What would be the price at a certain  time period ?
    - Price is the independent Variable (y)
    - Time is the dependent Variable ( time index is assumed as a dependent variable) ’X’

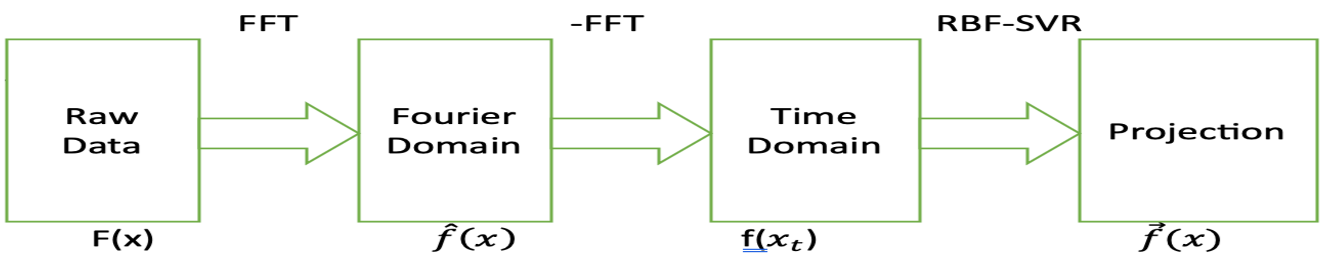
- Used Linear Regression method and Non-Parametric method for projecting the price of the asset.

# Agenda:

- Naive Linear Regression Method:
  - Projecting vector $P_x$ onto model space *C(X)*
  - If X has a full rank, $\boldsymbol{P_x} = \boldsymbol{H} = \mathbf{X(X'X)^{-1}X'}$
    - The projection matrix $\boldsymbol{P_x}$ is symmetric and idempotent
  - Model fit $\hat{\mu} = p_x y = \hat{\beta} x$ is the orthogonal projection of $y$ into *C(X)*
    - $\boldsymbol{y = \hat{\mu} + (y - \hat{\mu}) = p_x y + (I - p_x) y}$




- Fourier Transformed Radial Basis Support Vector Regression:
  - Model Assumption:
    - Instead of assuming SSE is due to some omitted variable, here I am assuming SSE is due to the Hibbert space noise and used Fourier transformation to filter out the noise and extract the true signal.



# Linear Regression method

**Regression Problem**

- T = $\{(x_{1,}\ y_1), \ldots.,(x_{i,}\ y_n)\}$, where $\boldsymbol{x_1 \epsilon R^n}$, $\boldsymbol{y_i \epsilon y = R}$, ***i* = 1,2,...n**

- Find the real function f(x) in $R^n$ .

- When *f(x)* is restricted to be a linear function then the corresponding problem is defined as linear regression problem:

  ***y* = *f(x)* = *xβ* + $\epsilon$**

- y|x ~ *N*(μ(x), $\sigma^2 I$)

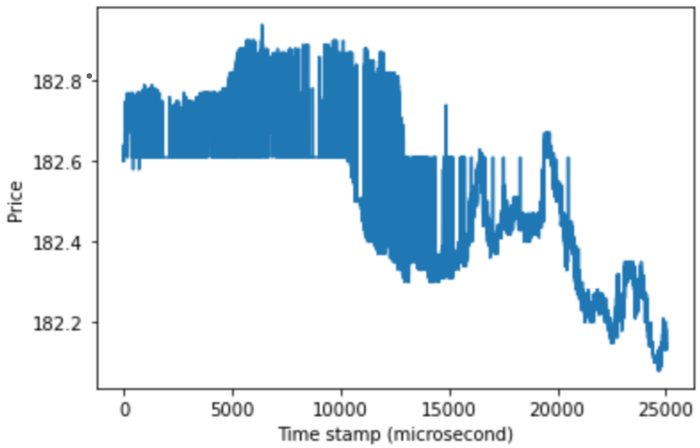
# Cont:

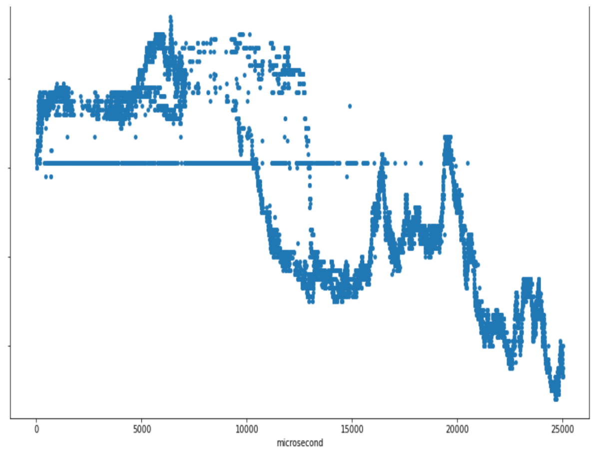
- Objective: Minimize SSE i.e $||y - x\beta||^2$

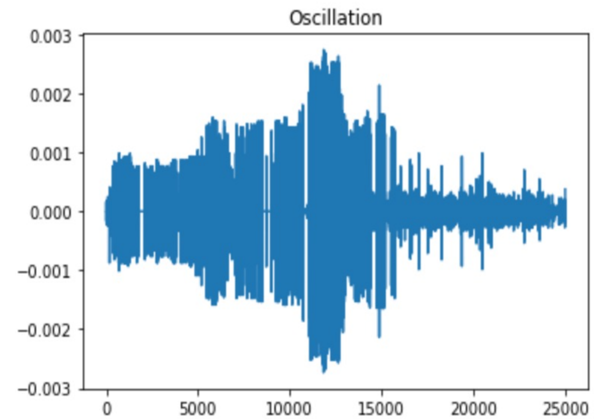
where,

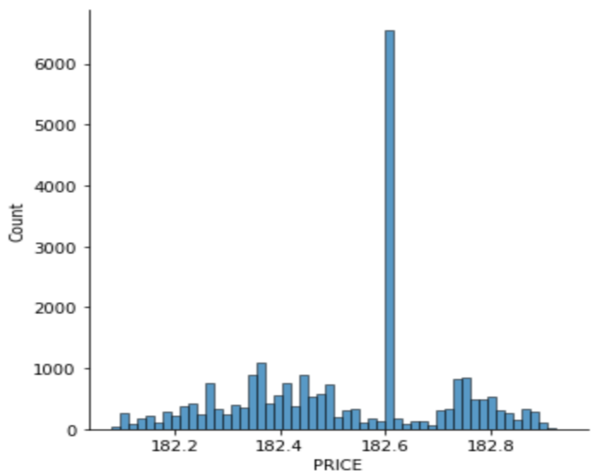
- $\hat{\beta} = (x'x)^{-1} x'y$
- $E(\hat{\beta}) = \beta$
- $Var(\hat{\beta}) = \sigma (x'x)^{-1}$
- Accuracy : $R^2 = \frac{\sum(y_i - \bar{y})^2 - \sum(y_i - \hat{y_i})^2}{\sum(y_i - \bar{y})^\wedge 2}$ **i.e** $\frac{SST - SSE}{SST}$
- Prediction error: **RSME** = $\sqrt{\frac{\sum(y_i - \hat{y}_i)^\wedge 2}{N}}$

# Observation:

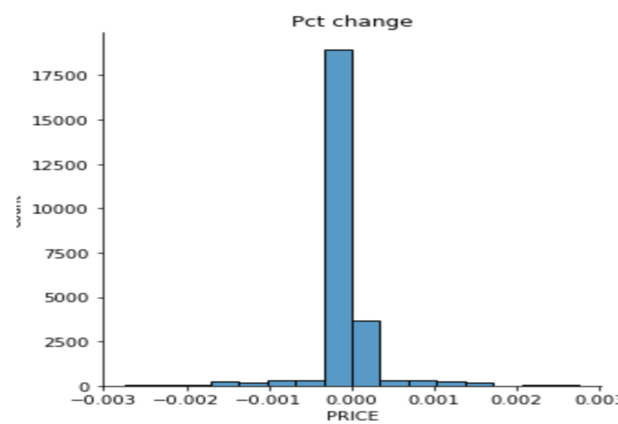



| | |
|---|---|
| count | 25011.000000 |
| mean | 182.523932 |
| std | 0.191680 |
| min | 182.080000 |
| 25% | 182.370000 |
| 50% | 182.590000 |
| 75% | 182.610000 |
| max | 182.940000 |



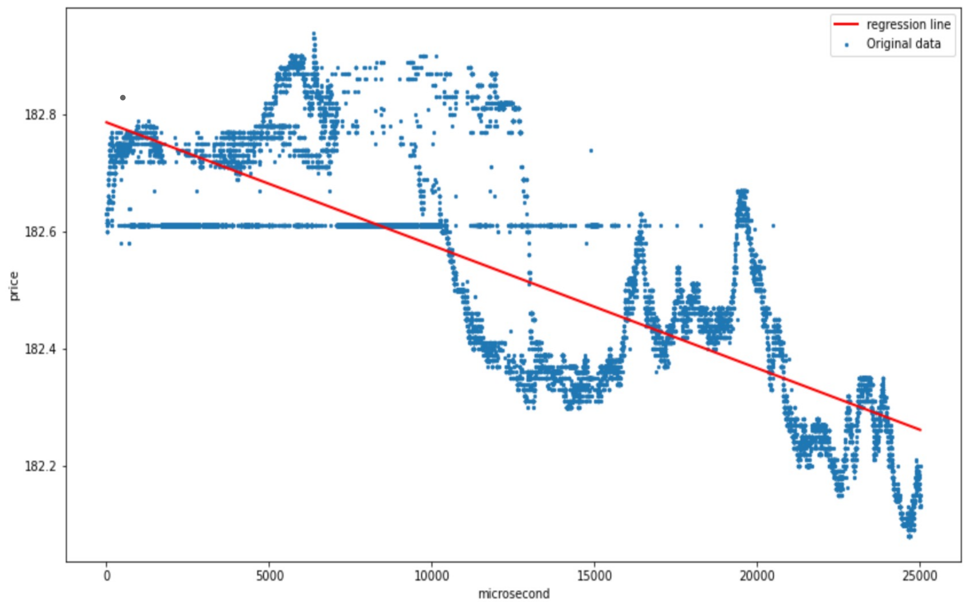
# Observation Cont:

- **For simplicity I assume normal distribution**
  i.e. the f(x) is continuous and normally distributed.

- Under normality, $\widehat{\boldsymbol{\beta}}_{lm}$ = $\widehat{\boldsymbol{\beta}}_{Glm}$ , even normal distribution happened to be exponential family distributions. (proof at appendix)

- Running an OLS and GLM is identical, However OLS is computationally cheaper.
  - $\mathbf{y} = \alpha + x\beta + \epsilon$

# Model Result:




```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.626
Model:                            OLS   Adj. R-squared:                  0.626
Method:                 Least Squares   F-statistic:                 4.182e+04
Date:                Thu, 23 Nov 2023   Prob (F-statistic):               0.00
Time:                        22:41:14   Log-Likelihood:                 18120.
No. Observations:               25011   AIC:                        -3.624e+04
Df Residuals:                   25009   BIC:                        -3.622e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        182.7865      0.001   1.23e+05      0.000     182.784     182.789
x1           -2.1e-05   1.03e-07   -204.503      0.000   -2.12e-05   -2.08e-05
==============================================================================
Omnibus:                     1474.165   Durbin-Watson:                   0.390
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1747.252
Skew:                           0.646   Prob(JB):                         0.00
Kurtosis:                       2.902   Cond. No.                     2.89e+04
==============================================================================
```

$\hat{\beta} = (x'x)^{-1} x'y$

$\alpha$

$$\frac{\sum(y_i - \bar{y})^2 - \sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})\text{^}2}$$

Root Mean Squared Error: 0.11725419080350061

# Is beta coefficient Biased?

- Null : $\boldsymbol{H_0}$ = $\mathbf{E}(\widehat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$
- Alternative: $\boldsymbol{H_1}$ = $\mathbf{E}(\widehat{\boldsymbol{\beta}}) \neq \boldsymbol{\beta}$
- T-test : $\boldsymbol{t} = \frac{\overline{\boldsymbol{y}}-\boldsymbol{\mu}}{\frac{\boldsymbol{s}}{\sqrt{\boldsymbol{n}}}} \sim \boldsymbol{t_{n-1}}$

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 182.7865 | 0.001 | 1.23e+05 | 0.000 | 182.784 | 182.789 |
| x1 | -2.1e-05 | 1.03e-07 | -204.503 | 0.000 | -2.12e-05 | -2.08e-05 |

- Larger values of $|t|$ provide stronger evidence against $H0$.
- At a 5% significance level, we reject the null hypothesis, indicating that there is evidence to suggest bias in the estimated coefficient $\beta$. We express 95% confidence in this conclusion.
- Can it be better?

## Extending linear regression;

- **From Linear Regression problem** T = $\{(x_{1,}\ y_1),\ldots.,(x_{i,}\ y_n)\}$

- **Introduce:**
  - transformation *x = Ø(x)*
  - kernel *k(x, x') = (Ø(x) . Ø(x'))*
  - decision function = *(w.x)* + *b*

- **Objective:**
  - maximize flatness
  - minimize deviation

## Linear Support Vector Regression:

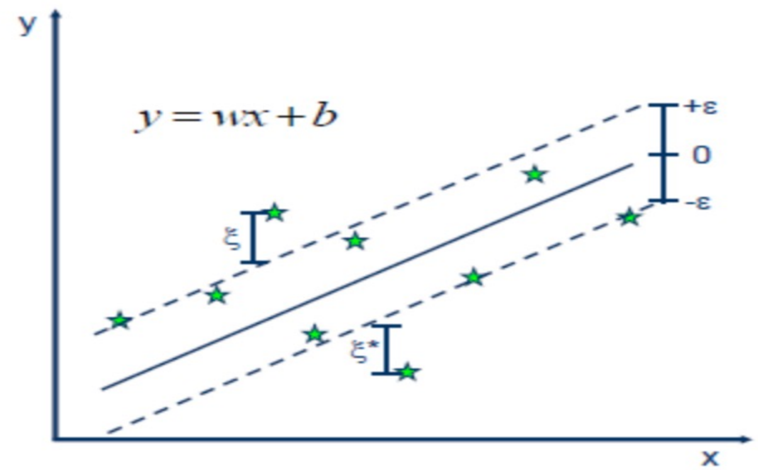
- **Established by converting linear regression problem to linear classification problems** (refer appendix).
- **Find a real function for training set** $T = \{(x_{1,}\ y_1), \ldots,(x_{i,}\ y_n)\}$, **where** $x_1 \epsilon R^n$, $y_i \epsilon\, y = R$, $i = 1,2,\ldots n$

- **When g(x) is restricted to be a linear function** ***y = g(x) = (w.x) + b*****, the corresponding problem is the linear regression problem.**

- **In the naive linear regression model, the objective is:**
  - *Minimize* $||y - x\beta||^2$
- **However, support Support Vector provides some flexibility.**
  - Hard Margin Hyper plain (Similar to the naive linear regression model)
  - Soft Margin Hyper plain (violates the naive linear regression model)

# Linear Support Vector Regression

- **Soft-margin Hyper plain Linear Support Vector Regression:**
  - Introduce 3 parameters;
    - $\xi^* = (\xi_1, \xi_1^*, \dots\dots\dots, \xi_i, \xi_j^*)'$
    - C is a penalty for violation
    - $\in$ - insensitive loss function:
      - $l_\in(\mathrm{y}, \mathrm{wx}) = \begin{Bmatrix} 0 & , if\ |y_i - wx_i| \leq \varepsilon \\ |y_i - wx_i| - \varepsilon, & otherwise \end{Bmatrix}$




  - The primal problem of linear soft vector regression can be written as:
    - $\min_{w,b,\xi^*} = \frac{1}{2}||w||^2$ + c $\sum_{i=1}(\xi_i + \xi^*{}_i)$
      - Subject to;
      - $y_i - wx_i - b \leq \varepsilon + \xi_i$ where, $\iota = 1,2, \dots. n$
      - $wx_i + b - y_i \leq \varepsilon + \xi^*$ where, $\iota = 1,2, \dots. n$
      - $\xi^* \geq 0\ where, i = 1,2 \dots., n$

# Cont:

- After obtaining the solution to the primal problem we get regression function:
    - **y = g(x) = ($\bar{w}.b$) + $b$**
    - Note: $\xi^*$ does not exists in the solution for the primal problem

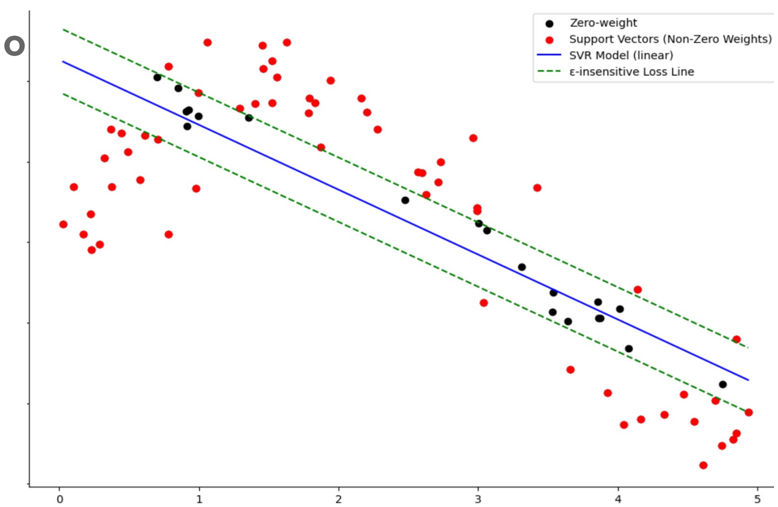
- In order, to address the above problem we can derive it as a dual problem,
    - we introduce the Lagrange multiplier function.
    - $$L(w, b, \xi^*, \alpha^*, \beta^*) = \frac{1}{2}||w||^2 + c\sum_{i=1}(\xi_i + \xi^*{}_i) - \sum_{i=1}(\beta_i\xi_i + \beta_i^*\xi^*{}_i) - \sum_{i=1}\alpha_i(\varepsilon + \xi_i + x_k - (w.x_i) - b) - \sum_{i=1}\alpha^*(\varepsilon + \xi_i^* + x_k - (w.x_i) + b)$$
        - *Where,*
          $\alpha^* = (\alpha_1\alpha_1^*, \ldots\ldots\ldots, \alpha_i\alpha_j^*)$ *are Lagrange multiplier vector*
          $\beta^* = (\beta_1\beta_1^*, \ldots\ldots\ldots, \beta_i\beta_j^*)$ *are Lagrange multiplier vector*

# Cont:

- **Substituting the derivatives of** *w, b,* $\xi^*$ **to the Lagrange function while maximizing** $\alpha^*, \beta^*$ **we get,**
  - $\max_{\alpha^*,\beta^*} = -\frac{1}{2}\sum_{i,j}^{n}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\,(\boldsymbol{x_i}\,,\ \boldsymbol{x_k}) - \varepsilon\sum_{i=1}^{n}(\alpha_i^* - \alpha_i) - \sum_{i=1}^{n} y_i(\alpha_i^* - \alpha_i)$

    Subject to:
    - $\sum_{i=1}^{n} y_i(\alpha_i^* - \alpha_i) = 0$
    - C - $\alpha_i^* - \beta_i^* = \mathbf{0},\ where\ i = 1,2,3,\dots,n$
    - $\alpha_i^*, \beta_i^* \geq 0,\quad where\ i = 1,2,3,\dots,n$
    - Turns to the quadratic programing problem

- **If $\alpha^*$ is the solution to any problems, then w and b can be o**
  - w = $\sum_{i=1}(\alpha_i - \alpha_i^*).\ \boldsymbol{x_i}$
  - b = $x_k$- $\sum_{i=1}(\alpha_i - \alpha_i^*).(\boldsymbol{x_i}\,,\ \boldsymbol{x_k}) - \varepsilon$




- **Decision Function:**
  - y = g(x) = $\sum_{i=1}(\alpha_i - \alpha_i^*).(\boldsymbol{x_i}\,,\ \boldsymbol{x_k}) + b$

- **Linearity till holds**; *y = wx + b*

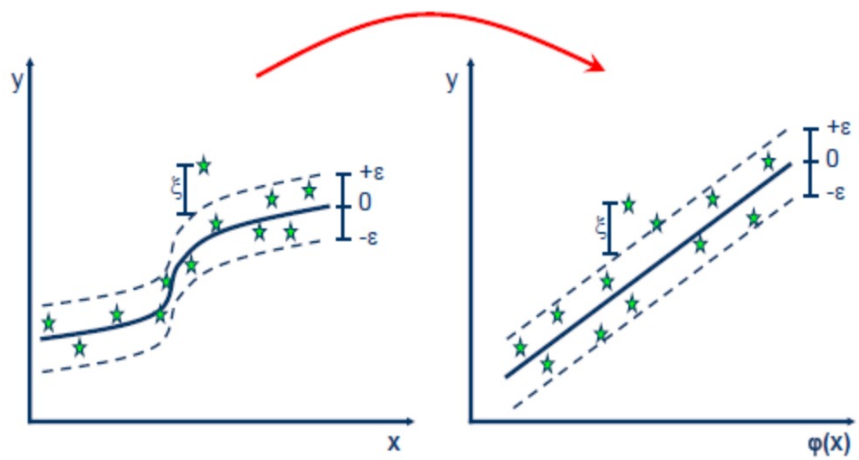
# Non-linear Transformation:

- Given : $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots \dots \dots, \boldsymbol{x_n}\} \subset \mathbf{x}$ plain

- Introduce a new space x = *Ø(x)* $and\ kernal\ k(x, x') = ($*Ø(x) . Ø(x')* )
    - Where,
        - K is a kernel function
- Then,
    - A = $\begin{pmatrix} \boldsymbol{k(x_1, x_1)} & \cdots & \boldsymbol{k(x_1, x_m} \\ \vdots & \ddots & \vdots \\ \boldsymbol{k(x_m, x_1)} & \cdots & \boldsymbol{k(x_m, x_m)} \end{pmatrix}$
    - Where,
        - All A∈ $R^n$
        - A is a positive definite
- Rewritten as ;
    - $\boldsymbol{k(x, x')}$ = $\sum_{i=1}$ ***(Ø(x) . Ø(x')*** , ***Ø : x → R***

# Cont:

- **The primal problem of non- linear soft vector regression can be written as:**
  - $\underset{w,b,\xi^*}{min} = \frac{1}{2}||w||^2 + c \sum_{i=1}(\xi_i + {\xi^*}_i)$
    - Subject to;
    - $y_i - w\phi(x_i) - b \leq \varepsilon + \xi_i$ where, $\iota = 1,2, \dots . n$
    - $w\phi(x_i) + b - y_i \leq \varepsilon + \xi^*$ where, $\iota = 1,2, \dots . n$
    - $(\xi_i, \xi^*) \geq 0\ where, i = 1,2 \dots . , n$




- Objective :
  - Maximize the flatness
  - Minimize the deviations

## Cont:

- We can derive dual problem from primal problem using the Langrage function.
  - $L(w, b, \xi^*, \alpha^*, \beta^*) = \frac{1}{2}||w||^2 + c\sum_{i=1}(\xi_i + \xi^*{}_i) - \sum_{i=1}(\beta_i\xi_i + \beta_i^*\xi^*{}_i) - \sum_{i=1}\alpha_i(\varepsilon + \xi_i + x_k - (w.\varphi(x_i)) - b)$
    $- \sum_{i=1}\alpha^*(\varepsilon + \xi_i^* + x_k - (w.\varphi(x_i)) + b)$

    *Where,*

    $\alpha^*, \beta^*$ *are Lagrange multiplier vector*

    ***Solving w.r.t w, b, $\xi^*$ while $\alpha^*, \beta^*$ we get,***

  - $$\max_{\alpha^*,\beta^*} = -\frac{1}{2}\sum_{i,j}^{n}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(\varphi(x_i),\ \varphi(x_k)) - \varepsilon\sum_{i=1}^{n}(\alpha_i^* - \alpha_i) - \sum_{i=1}^{n} y_i(\alpha_i^* - \alpha_i)$$

    Subject to:

    - $\sum_{i=1}^{n} y_i(\alpha_i^* - \alpha_i) = 0$
    - $C - \alpha_i^* - \beta_i^* = 0,\ where\ i = 1,2,3,\dots,n$
    - $\alpha_i^*, \beta_i^* \geq 0,\quad where\ i = 1,2,3,\dots,n$

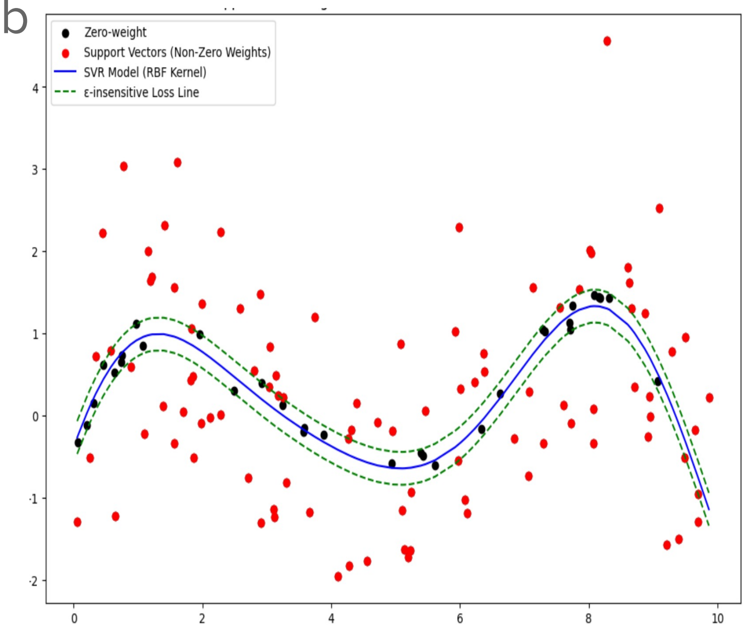
- In order to only include $\alpha^*$, we can further simplify the above convex quadratic problem,
  - $\min_{\alpha^*} = \frac{1}{2}\sum_{i,j}^{n}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(\varphi(x_i), \varphi(x_k)) - \varepsilon\sum_{i=1}^{n}(\alpha_i^* - \alpha_i) - \sum_{i=1}^{n} y_i(\alpha_i^* - \alpha_i)$

    Subject to:

    - $\sum_{i=1}^{n} y_i(\alpha_i^* - \alpha_i) = 0$
    - $0 \leq \alpha_i^* \leq C,\ i = 1,2,\dots,n$

# Cont:

- Similar to, linear, if $\alpha^*$ is know we can obtain w and b
  - w = $\sum_{i=1}(\alpha_i - \alpha_i^*).\ \boldsymbol{\varphi}(\boldsymbol{x_i})$
  - b = $x_k$- $\sum_{i=1}(\alpha_i - \alpha_i^*).(\boldsymbol{\varphi}(\boldsymbol{x_i}).\ \boldsymbol{\varphi}(\boldsymbol{x_k}))$ - $\varepsilon$




- Decision Function:

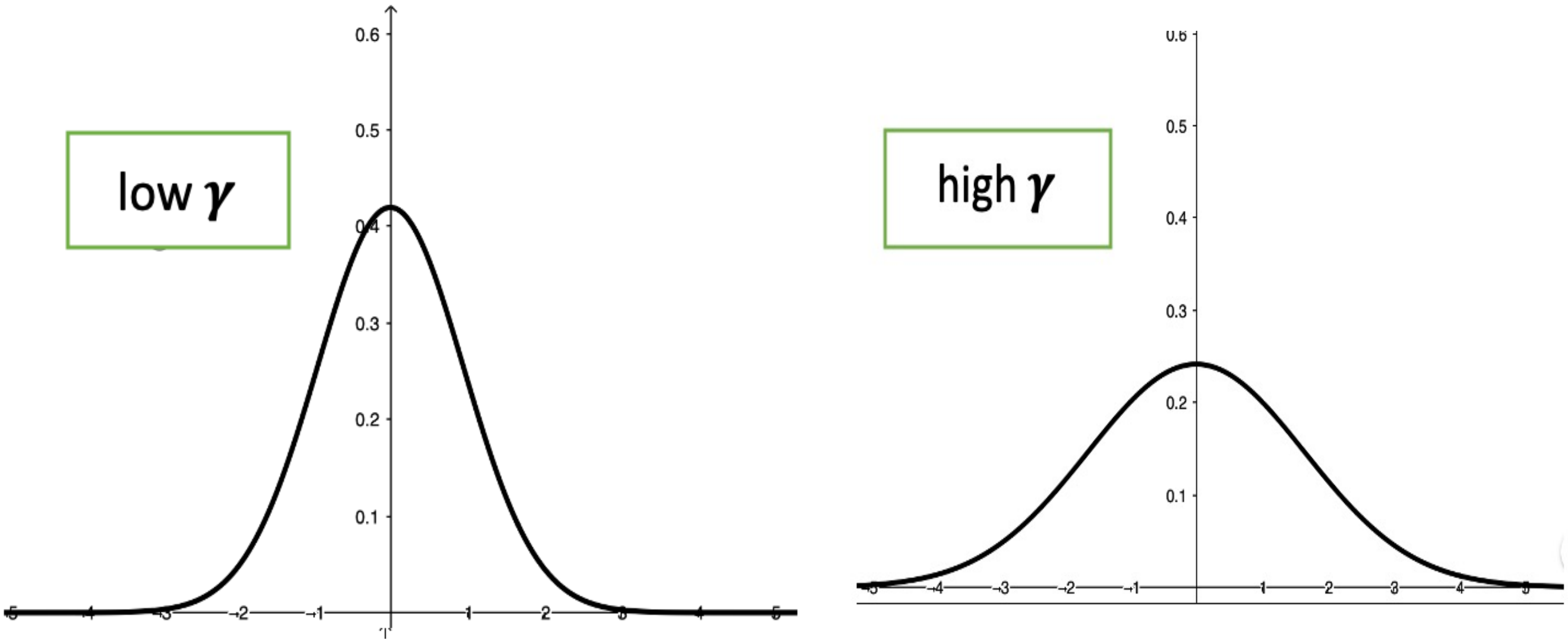
$$y = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \cdot \langle \varphi(x_i), \varphi(x) \rangle + b$$

$$\mathbf{y = g(x) =} \ \sum(\alpha_i \ - \ \alpha_i^*)\ k(x_i\,,\, x_k\,) + b$$

- For the propose of this project the kernel is Gaussian Radial Basis function (RBF)
  - $k(x_i\,,\, x_k\,) = exp\left(-\gamma||x_i - x_k||^2\right)$

    Where,
    - γ is a parameter that sets the “spread” of the kernel
    - $||x_i - x_k||$ is the Euclidean distance between points $x_i$ *and* $x_k$

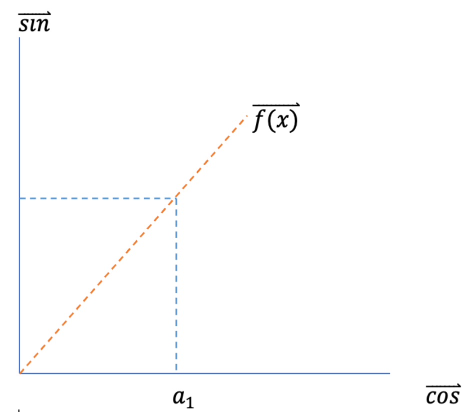
# Gaussian RBF Cont:



# Controlling Noise:

- Fourier Series Basis idea:
  - We can approximate *f(x)* with cos and sine of higher and higher frequencies.

    i.e. f(x) = $\frac{A_0}{2} + \sum_{k=1}^{\infty}(A_k \cos\left(\frac{2\pi kx}{L}\right) + B_k \cos\left(\frac{2\pi kx}{L}\right)\ )$
  - Coefficients are determined by taking the Hilbert space inner product of *f(x)* with their respective functions.




$A_k = \frac{2}{L}\int_0^L f(x)\, Cos\left(\frac{2\pi kx}{L}\right) dx.$ i.e. $\frac{1}{||\cos(kx)||^2} < f(x), \cos(kx) >$

$A_B = \frac{2}{L}\int_0^L f(x)\, Sin\left(\frac{2\pi kx}{L}\right) dx.$ i.e. $\frac{1}{||\sin(kx)||^2} < f(x), \sin(kx) >$

<u>In Complex Form:</u>

f(x) = $\sum_{k=-\infty}^{\infty} C_k e^{ikx}$

where,

$e^{ikx}$ = *cos(kx) + i sin(kx)* *{Euler formula that is obtained by solving the Taylor series Expansion}*

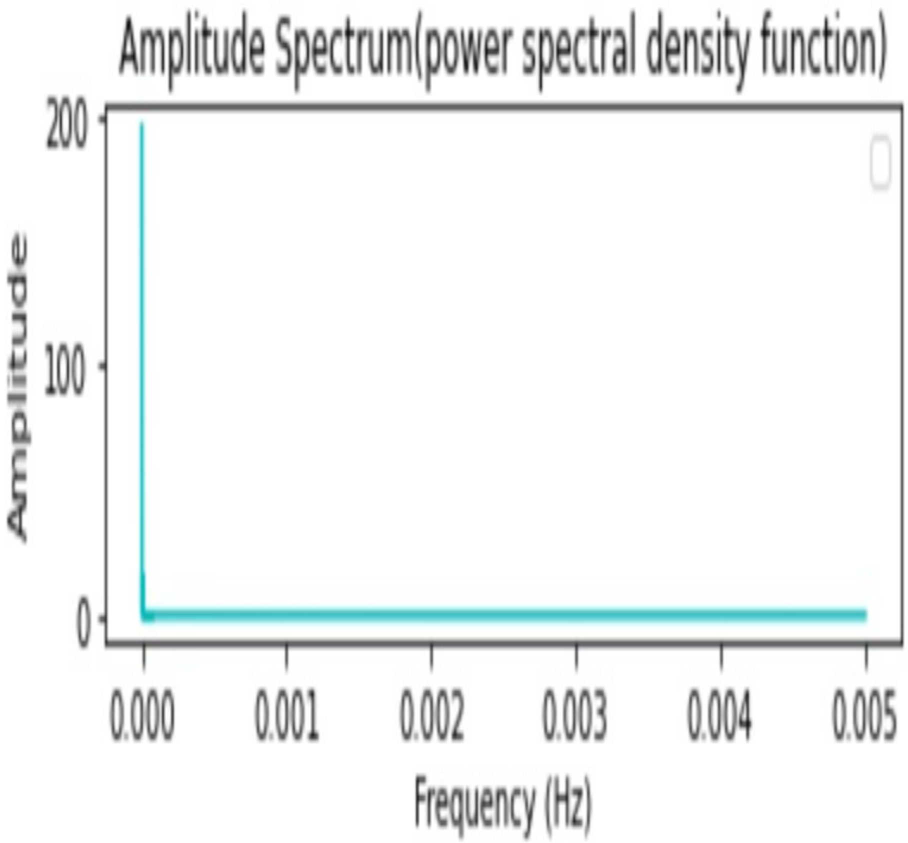
## Fourier Series to Fourier Transform:

- For Fourier Transform, L goes to infinity and does not remain periodic.
  - When we approximate Fourier complex for n discreet value, we obtain:
    - $\hat{f}_k = \sum_{j=0}^{n-1} f_j e^{-i2\pi jk/n}$ (complex form)
    - $f_k = (\sum_{j=0}^{n-1} \hat{f}_j e^{i2\pi jk/n})\frac{1}{n}$ (Converging back into time domain (real number))
      - Frequency$(\omega_n) = e^{-2\pi i/n}$ *where,* $i = \sqrt{-1}$

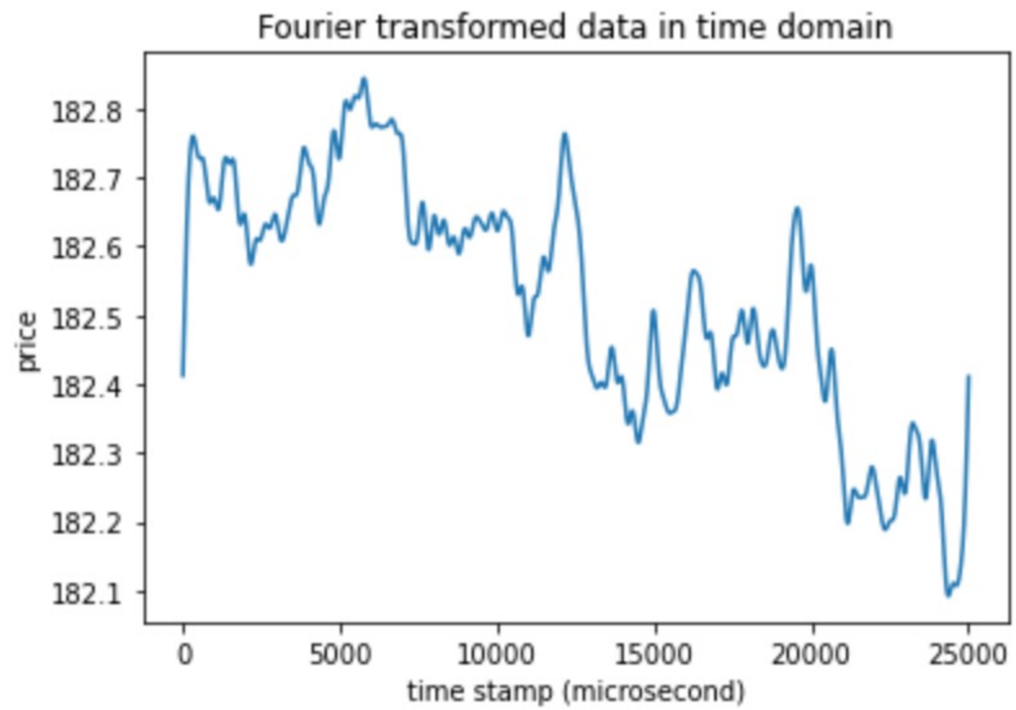
- Fast Fourier Transform*(FFT)* is an algorithm that implements Discreet Fourier Transform*(DFT).*
  - *Computationally cheap i.e less time complexity O(nlog),* compared to DFT $O(n^2)$

# FFT results:

- Fourier Domain




Time Domain



# Final Model:

- y = g$(x) = \sum(\boldsymbol{\alpha_i} - \boldsymbol{\alpha_i^*})\boldsymbol{k}\left(\boldsymbol{x_i^{-f}}, \boldsymbol{x_j^{-f}}\right) + \boldsymbol{b}$

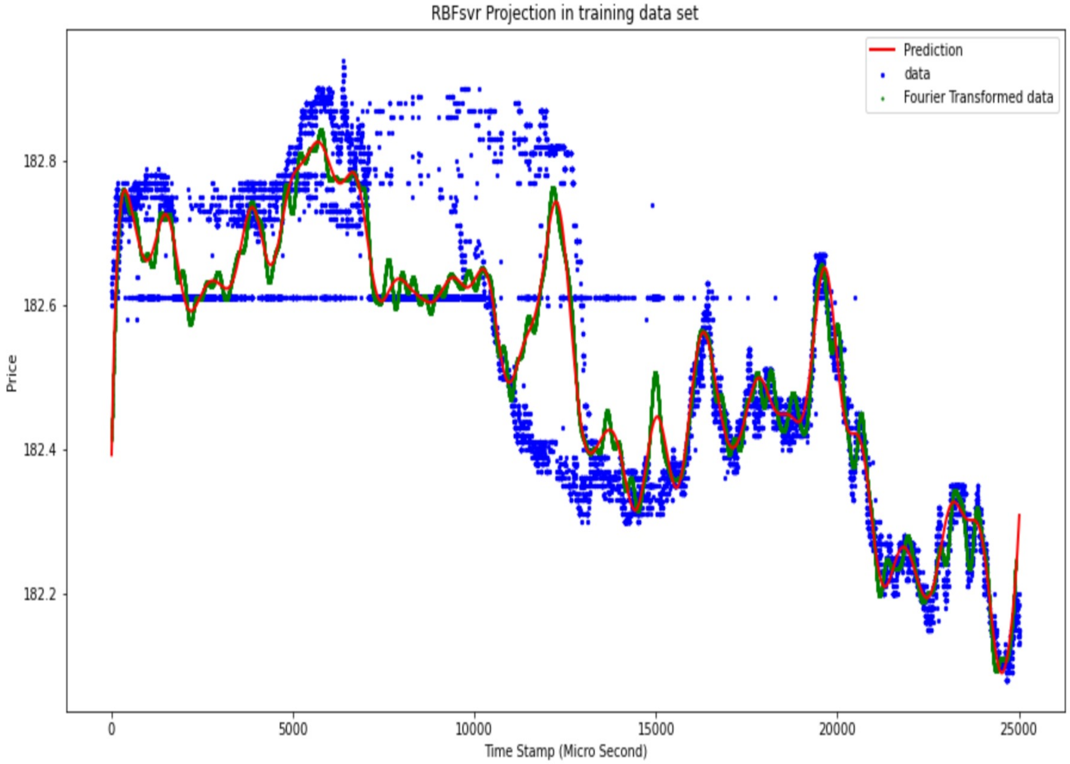
  Where,

  - **K is a Gaussian Radial Basis Function**

  - $\boldsymbol{x^{-f} is\ a\ Denoised\ data}$

# Fourier Transformed RBF SVR Training Result:

- Given $fourier\ transformed$ $f(x)$: $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots \dots \dots, \boldsymbol{x_n}\} \subset \boldsymbol{X}$ *plain*




R-Squared: 0.8503
Root Mean Squared Error: 0.0742
C: 100
gamma 1e-06
epsilon 7e-06

# Remarks:

- The projection accuracy on in sample using RBF –SVR on the noise filtered training data increased by 4% compared to RBF –SVR using raw training data.
- The assessment of the model's reliability and robustness is contingent upon its performance on the test dataset.
- I manually selected hyperparameters in order to  substantiate and provide a comprehensive overview of the model.
- The optimal value of the parameter can be determined through hyperparameter tuning employing cross-validation to avoid overfitting.

# THANK YOU!

# Apendix :

Proof:

**Generalized Linear Model**

- $\mu(x) = x'\beta$
- $g(\mu(x)) = x'\beta$ , where g is the link function
  - Link function makes two variable compatible.

GLM coefficient estimate:

- $E[y] = x\beta$

- $\varepsilon_t \sim N(0, \boldsymbol{\sigma}^2 I)$, where $\varepsilon$ is *IID*

- By the linear transformation of normal random variable, y is conditionally normal.
- The conditional PDF is given by:

  $f_y(y \mid x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2\sigma^2}(y - x\beta)^2)$

- $f(y \mid \beta, \boldsymbol{\sigma}^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2\sigma^2}(y - x\beta)^2)$

  Likelihood function;

  $L(\vartheta \mid y) = L(\beta, \boldsymbol{\sigma}^2 \mid y)$

  $= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2\sigma^2}(y - x\beta)^2)$

  $= \frac{1}{\sqrt{(2\pi\sigma^2)}} - \frac{n}{2} \exp(-\frac{1}{2\sigma^2} \sum_{i=1}^{n}(y - x\beta)^2)$

  Taking natural logarithm of the likelihood function we get,

  $= -\frac{n}{2}\ln(2\pi) - \frac{n}{2}(\boldsymbol{\sigma}^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n}(y - x\beta)'(y - x\beta)$

  take partial derivative wrt β ,

  $\frac{\partial L(\vartheta \mid y)}{\partial \beta} = -\frac{1}{2\sigma^2} \sum_{i=1}^{n}(y - x\beta)'(y - x\beta)$

  simplifying above equation we get,

  $\frac{\partial L(\vartheta \mid y)}{\partial \beta} = y'y - 2x'y + \beta'x'x\beta$

  $= -2xy' + 2x'x\beta$

  set the derivative equal to 0, we get

  $-2xy' + 2x'x\beta = 0$

  $-2x'x\beta = -2xy'$

  solving above equation we get,

  $\widehat{\boldsymbol{\beta}} = (\boldsymbol{x}'\boldsymbol{x})^{-1}\boldsymbol{x}'\boldsymbol{y}$

## Cont:

- $Y \mid X \sim \mathcal{F}$ from exponential family
- Exponential family is the generalization in which a lot of distribution lives.
- Distribution is said to be the Exponential family distribution if the probability of a vector y given a parameter vector $\vartheta$ is :
  - $P(y \mid \vartheta, \phi) = \exp\left(\frac{y\vartheta - b(\vartheta)}{a(\phi)} + c(y, \phi)\right)$

- Suppose if we were interested in the direction of the price move, where the direction is denoted by the binary value.
- $\varepsilon$ is *Independent and identically distributed*.
  - $\vartheta = \log(\frac{\pi}{1-\pi})$ , which is a logit link
  - $\phi = 1$
  - $b(\vartheta) = n \log(1+\exp(\vartheta))$
  - $c(y, \phi) = \log\binom{n}{y}$
  - This leads to the Bernoulli distribution.

Therefore, $p(1 \mid \vartheta) = \frac{\exp(\vartheta)}{\exp(\vartheta)+1}$ and $p(0 \mid \vartheta) = \frac{1}{\exp(\vartheta)+1}$

# Basic Idea Support Vector Machine :linear classification problems:

- Find Hyper plain / support line
- Two support line can be expressed as -v and v:
  - Let, w = $\frac{w}{v}$, b = $\frac{b}{v}$ ⟺ $(wx + b = 1 \text{ and } wx + b = -1)$
    - Then, expression become $(wx + b = 0)$
- Find separation line $(wx + b = 0)$ that maximizes the margin 'w':
  - this turns into optimization problem.
  - $\max_{w,b} = \frac{2}{||w||}$

    Subject to:

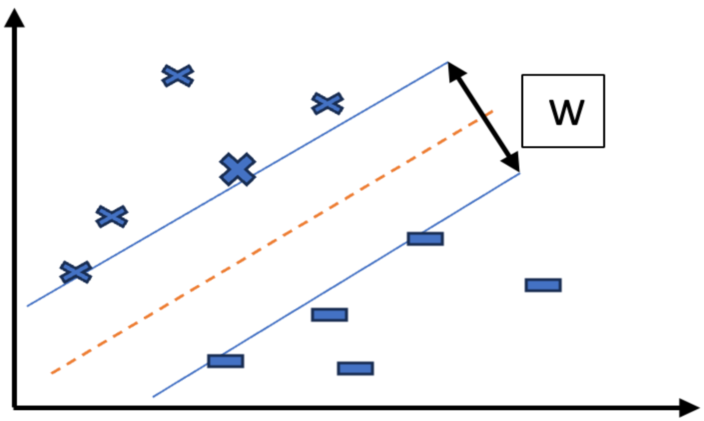
    $(wx + b \geq 1)$ , $y_i = 1$

    $(wx + b \leq -1)$ , $y_i = -1$

  Alternatively,

  $\min_{w,b} = \frac{1}{2}||w||^2$

  **subject to:** $y_i(wx_i) + b \geq 1, i = 1, 2, .., n$




- The above primal problem can be solved using the dual problem:
  - Introduce Lagrange function
  - $L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i \, (y_i(wx_i) + b - 1)$
  - Solving w.r.t w & b we get,
    - $\nabla_w L = w - \sum_{i=1}^{n} \alpha_i \, y_i \, x_i = 0$
    - $\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i \, y_i = 0$
  - Substituting the above equation to the Lagrange while maximizing $\alpha$ we get,
    - $L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \, \alpha_i \alpha_j x'_i x_j$
    - **max** $\alpha$ **s.t** $\alpha_i \geq 0$, where $i = 1, 2, \ldots, n$ & $\sum_{i=1}^{n} \alpha_i \, y_i = 0$
    - This turns out to be the quadratic programing problem

- Kernel Types:
  - Polynomial Kernel : k($x_i, x_k$) $= ((x_i, x_k) + 1)^d$ , where d = 1, 2,...,n
  - Gaussian Radial Basis Kernel : k($x_i, x_k$) $=$ exp( - $\frac{(x_i, x_k)^2}{\sigma^2}$ ), where $\sigma > 0$
  - Sigmoid Kernel : k($x_i, x_k$) $=$ tanh (b($x_i \cdot x_k$) $+$ c). , where b is slope , c is bias

# Citations:


- Agresti, A. (2022). *Foundations of linear and generalized linear models*. Wiley.

- Deng, N. (2013). *Support Vector Machines: Theory, algorithms, and extensions*. CRC Press.
- Brunton, S. L., & Kutz, J. N. (2022). *Data-driven science and engineering machine learning, Dynamical Systems, and Control*. Cambridge University Press.

- Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H.-T. (2012). *Learning from data: A short course*. AML Book.

- *Support Vector Machine - Regression (SVR)*. Support vector regression. (n.d.). https://www.saedsayad.com/support_vector_machine_reg.htm