INFSCI 2915: Machine Learning
Extension to Linear Regression Model & Comparison with KNN regression

Mai Abdelhakim

School of Computing and Information

610 IS Building

Spring 2018

Objectives of this Unit

- Calculate R^2 metric
- Include quantitative features in linear regression
- Relax the additive assumption of the linear model
- Relax the linear assumption on the linear regression model – Polynomial regression
- Compare Linear Regression to KNN regression

Assessing Model Accuracy

- R^2 metric is a number between [0,1]

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

  - TSS is the total sum of squares $\mathrm{TSS} = \sum (y_i - \bar{y})^2$

- R^2 measures the proportion of variability in Y that can be explained using feature (X)
- Higher R^2 metric is desired

R^2 metric Calculations

Calculate the R^2 metric of OLS using the training in the table

Training Index *(i)*	Target (y_i)	Feature (x_i)
1	5	6
2	7	9
3	8	10
4	10	12
5	11	13
6	13	16

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

$$\mathrm{RSS} = e_1^2 + \ldots + e_n^2 = \sum_{i=1}^{n} [\hat{y}_i - y_i]^2 = \sum_{i=1}^{n} [\hat{\beta}_0 + \hat{\beta}_1 x_i - y_i]^2$$

$$\mathrm{TSS} = \sum (y_i - \bar{y})^2$$

http://www.uv.es/uriel/2%20Simple%20regression%20model%20estimation%20and%20properties.pdf

R^2 metric Calculations

Calculate the R^2 metric using the training in the table

Training index	Target (y_i)	Feature (x_i)	$y_i x_i$	x_i^2
1	5	6	30	36
2	7	9	63	81
3	8	10	80	100
4	10	12	120	144
5	11	13	143	169
6	13	16	208	256
sum	54	66	644	786

n=6 $\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} = 9$ $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = 11$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2 - \bar{x}\sum_{i=1}^{n} x_i} \Rightarrow \hat{\beta}_1 = \frac{644-9x66}{786-11x66} = 0.83$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x} \Rightarrow \hat{\beta}_0 = 9 - 0.83x11 = -0.13$$

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i = -0.13 + 0.83\, x_i$$

http://www.uv.es/uriel/2%20Simple%20regression%20model%20estimation%20and%20properties.pdf

R^2 metric Calculations

Calculate the R^2 metric using the training in the table

n=6 $\bar{y} = \frac{\sum_{i=1}^{y} y_i}{n}$ = 9 $\bar{x} = \frac{\sum_{i=1}^{y} x_i}{n} = 11$

Training index	Target (y_i)	Feature (x_i)	$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ $= -0.13 + 0.83\, x_i$
1	5	6	4.85
2	7	9	7.34
3	8	10	8.17
4	10	12	9.83
5	11	13	10.66
6	13	16	13.15
sum	54	66	

RSS (on training data) = $\sum_{i=1}^{n} [\hat{y}_i - y_i]^2 = 0.334$

TSS = $\sum_{i=1}^{n} [\bar{y} - y_i]^2 = 42$

Training R^2 metric in this example = (TSS-RSS)/TSS=0.99

Note that we access the model using test data, which can be evaluated in a similar manner but using a test dataset

http://www.uv.es/uriel/2%20Simple%20regression%20model%20estimation%20and%20properties.pdf

R^2 in python

- Using the score method:
  Fitted_model.score(X_test,Y_test)

- Or use metric module in sklearn
  from **sklearn.metrics** import **r2_score**
  predicted_target= fitted_model.**predict**(X_test)
  r2score=**r2_score**(Y_test, predicted_target)

Regression with Qualitative Features

- Some features may take discrete values (qualitative)
  - Examples: gender, ethnicity, marital status

- How to model qualitative features this problem?
  - **Define a dummy variable** based on the qualitative features

Regression Model with Qualitative Features

- Example: investigate difference in credit card balance between females and males
  - Here the feature has two possibilities only
- For example, to represent the gender feature, you can define a dummy variable

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male} \end{cases}$$

- The model becomes:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

β_0 is the average credit card balance among males
$\beta_0 + \beta_1$ is the average credit card balance among females
β_1 is the average difference in credit card balance between females and males

- Dataset can be found here: http://www-bcf.usc.edu/~gareth/ISL/data.html
  - Dataset also includes balance, gender, income, card limit, age, and other features
- P vale of the dummy variable is high, which suggests that gender has no significant impact on the credit card balance

	Coefficient	Std. Error	t-statistic	p-value
Intercept	509.80	33.13	15.389	< 0.0001
gender[Female]	19.73	46.05	0.429	0.6690

estimate of β_0 *estimate of* β_1 ➔ meaning that the females have 19.73 additional debt

High p-value, gender is insignificant feature

Question: What happens if the dummy variable is 0 for females and 1 for males? What will the new coefficients estimate be?

- Dataset can be found here: http://www-bcf.usc.edu/~gareth/ISL/data.html
  - Dataset also includes balance, gender, income, card limit, age, and other features
- P vale of the dummy variable is high, which suggests that gender has no significant impact on the credit card balance

	Coefficient	Std. Error	t-statistic	p-value
Intercept	509.80	33.13	15.389	< 0.0001
gender[Female]	19.73	46.05	0.429	0.6690

estimate of β_0 *estimate of* β_1 ➔ meaning that the females have 19.73 additional debt

Question: What happens if the dummy variable is 0 for females and 1 for males? What will the new coefficients estimate be?

Answer: *estimate of* β_0 = 509.80 + 19.73, *estimate of* β_1= -19.73

Other Coding Schemes for Qualitative Variables

- The choice of the code is arbitrary and has no effect on the regression fit
  - But changes the interpretation of the coefficients

- Another way to model the previous example, is to define

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ -1 & \text{if } i\text{th person is male} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 - \beta_1 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

- Final **predictions for the credit balances will be the same regardless of the coding scheme** used to model the qualitative variable.

Qualitative and Quantitative

- Suppose we have both **gender** and **income** as features:

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male} \end{cases}$$

- Regression model for predicting credit card balance is (assume error term = 0):

$$Y_i = \beta_0 + \beta_1 income_i + \beta_2 x_i \;= \begin{cases} \beta_0 + \beta_1 income_i & male \\ \beta_0 + \beta_2 + \beta_1 income_i & female \end{cases}$$

Qualitative variables with more than two levels

- We define **number dummy variables** = number of levels - 1
- For example, for ethnicity (Asian, Caucasian, African American [AA]) we can create two dummy variables

$$x_{i1} = \begin{cases} 1 & \text{if } i\text{th person is Asian} \\ 0 & \text{if } i\text{th person is not Asian} \end{cases}$$

$$x_{i2} = \begin{cases} 1 & \text{if } i\text{th person is Caucasian} \\ 0 & \text{if } i\text{th person is not Caucasian} \end{cases}$$

- Model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is Asian} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } i\text{th person is Caucasian} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is AA.} \end{cases}$$

	Coefficient	Std. error	t-statistic	p-value
Intercept	531.00	46.32	11.464	< 0.0001
ethnicity[Asian]	-18.69	65.02	-0.287	0.7740
ethnicity[Caucasian]	-12.50	56.68	-0.221	0.8260

- In this example, dummy variables have **high p-values** = > **week association** with credit card balance

- Python: encoding of qualitative variables
  http://scikit-learn.org/stable/modules/preprocessing.html#encoding-categorical-features

Assumption of the Linear Regression Model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_p x_p$$

- Two main assumptions
  - Additive assumption
  - Linear Assumption

How to relax these assumptions?

Additive Assumption

- Additive assumption: the change in the response due to one-unit change in feature i is *constant* (β_i), and is *independent* of other features
  - Example: we assumed that the sales increases with TV budget regardless of the amount spent on radio

$$\widehat{\texttt{sales}} = \beta_0 + \beta_1 \times \texttt{TV} + \beta_2 \times \texttt{radio} + \beta_3 \times \texttt{newspaper}$$

- In practice, the impact of a feature on the response may be affected by the other features

- Examples:
  - The increase of spending on radio advertising (X_1) may increase slop of TV (X_2) with Sales (Y)
  - Factory productivity (Y) increases with assembly lines (X_1) depends on number of workers (X_2)

How to Relax the Additive Assumption

- Include an **interaction term** to **relax the additive assumption**

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 (X_1 X_2)$$
$$Y = \beta_0 + (\beta_1 + \beta_3 X_2)\, X_1 + \beta_2 X_2$$

- Adjusting X_2 will change the impact of X_1 on Y

Example: Productivity of a Factory

- Measure productivity of a factory with features: number of workers, number production lines

- Additive assumption:

$$\text{productivity} = \hat{\beta}_0 + \hat{\beta}_1\, lines + \hat{\beta}_2\, workers$$

  - Increasing the number of production line increases the productivity, regardless of the number of workers
    - This is not accurate, since increasing the production lines may not be productive unless the are more workers to operate them

- Relax additive assumption by including interaction term

$$\text{productivity} = \hat{\beta}_0 + \hat{\beta}_1\, lines + \hat{\beta}_2\, workers + \hat{\beta}_3\, lines \,.\, workers$$

- *Adding new line will increase productivity by ($\hat{\beta}_1$ + $\hat{\beta}_3$ workers)*
- Having more workers, the increasing the assembly line will be more effective

Example: Advertising

- Include: Radio, TV, and interaction term TV x Radio in the advertising dataset

To code in python use: `model=smf.ols('Sales ~ TV+Radio+TV*Radio', AdvertisingData)`

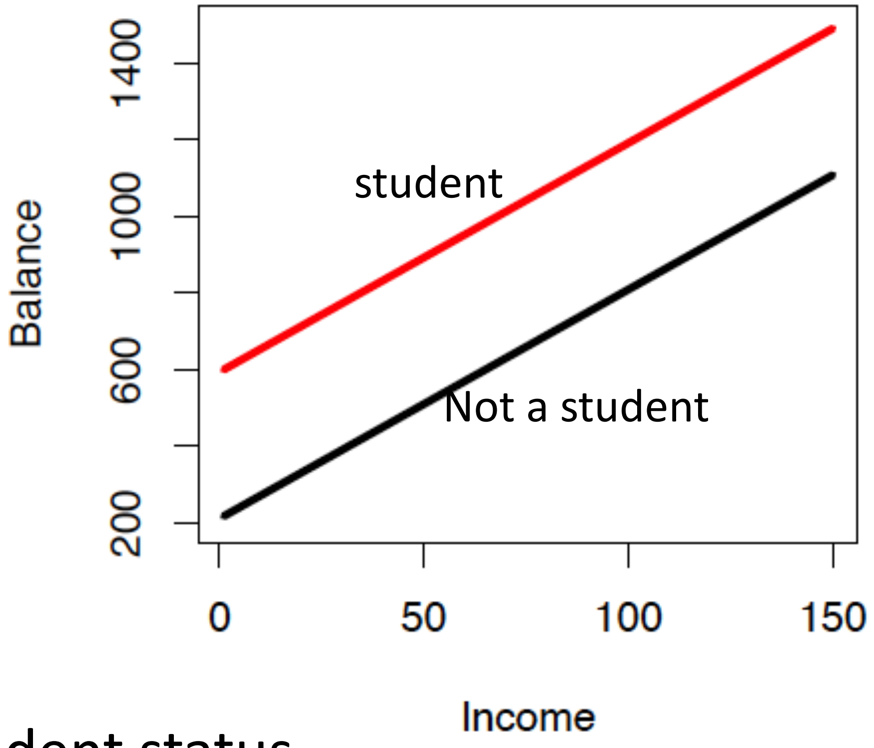
	Coefficient	Std. error	t-statistic	p-value
Intercept	6.7502	0.248	27.23	< 0.0001
TV	0.0191	0.002	12.70	< 0.0001
radio	0.0289	0.009	3.24	0.0014
TV×radio	0.0011	0.000	20.73	< 0.0001

- **Interaction term (TV x Radio) has low p-value indicating that the actual relationship is not additive**
  - Increase spending on radio advertising increase slop of TV => this is called Synergy effect in marketing

Example: Credit Card Balance
Interaction Between Qualitative and Quantitative Features

- Predict credit card balance as function of income (quantitative) and whether the card holder is student or not (qualitative).
- One can have a model: no interaction term

$$\begin{aligned}
\texttt{balance}_i &\approx \beta_0 + \beta_1 \times \texttt{income}_i + \begin{cases} \beta_2 & \text{if } i\text{th person is a student} \\ 0 & \text{if } i\text{th person is not a student} \end{cases} \\
&= \beta_1 \times \texttt{income}_i + \begin{cases} \beta_0 + \beta_2 & \text{if } i\text{th person is a student} \\ \beta_0 & \text{if } i\text{th person is not a student} \end{cases}
\end{aligned}$$



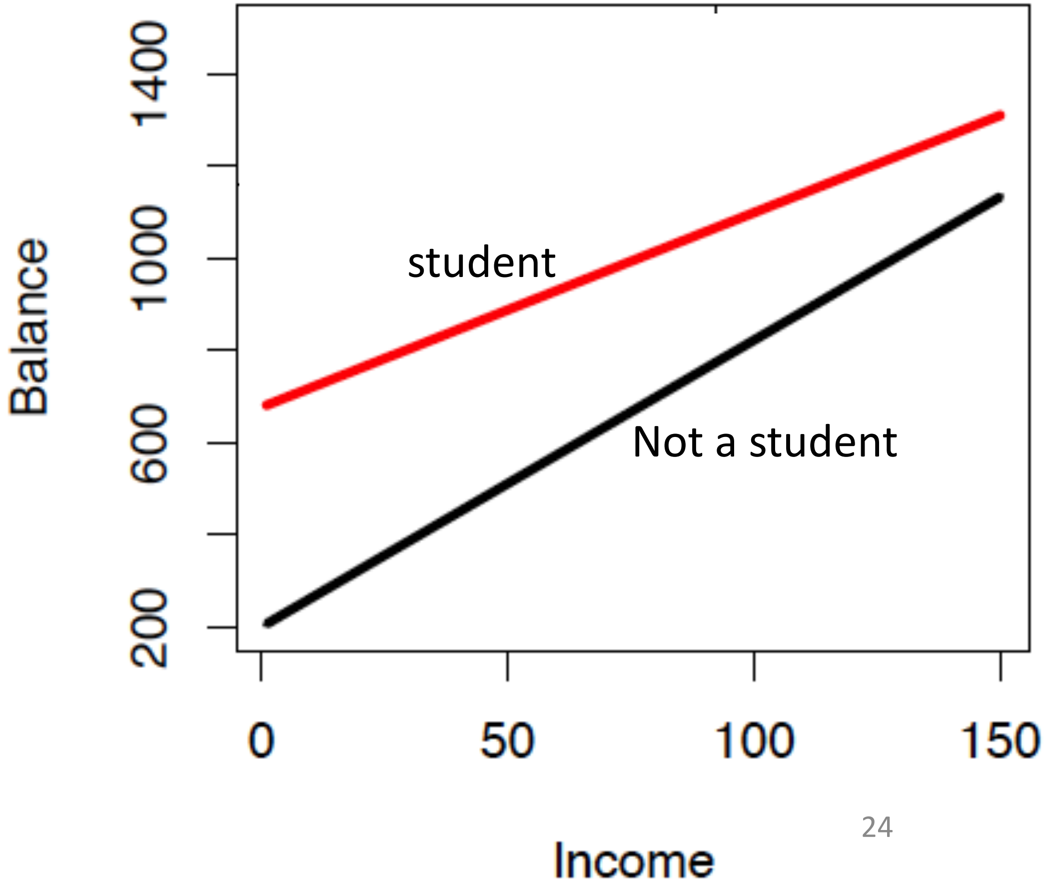
With this model we have same slop different intercept for student status

- By including the interaction term, the model will be:

$$\texttt{balance}_i \approx \beta_0 + \beta_1 \times \texttt{income}_i + \begin{cases} \beta_2 + \beta_3 \times \texttt{income}_i & \text{if student} \\ 0 & \text{if not student} \end{cases}$$

$$= \begin{cases} (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \times \texttt{income}_i & \text{if student} \\ \beta_0 + \beta_1 \times \texttt{income}_i & \text{if not student} \end{cases}$$

- Both the intercept and the slope are different
- Slope of students is lower!
    - Note that coefficients can be negative



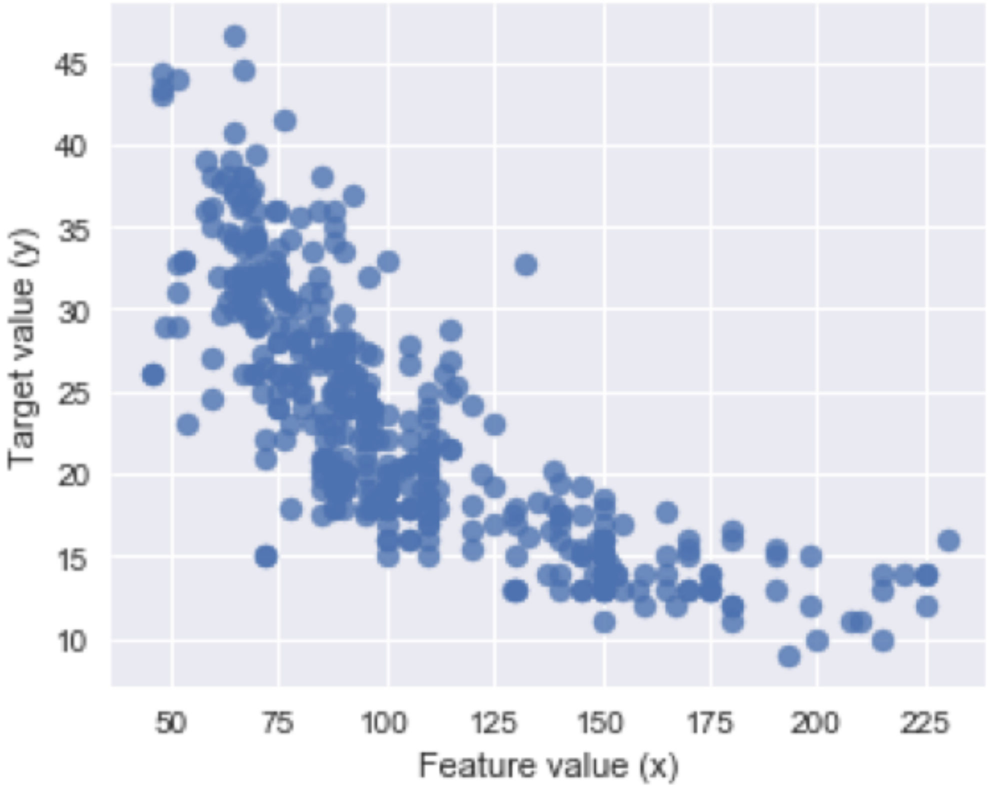
Common practice

- If the interaction term (e.g. $X_1 X_2$) is important (has low p-value), then we also include the individual terms (X_1) and (X_2) regardless of their p-value: hierarchy principle

- The interactions are hard to interpret in a model without main effects.

Linear Assumption

- The linear model assumed that there is a **linear relationship** between the response and the features
- Actual relationship may not be linear



Polynomial Regression

- Linear assumption can be relaxed to include non-linear relationship
    - **Still with a linear regression model!**

- A simple approach to incorporate non-linear relationships to a linear model is to **include transformed versions of the predictors** into the model

This is called **polynomial regression**

Example

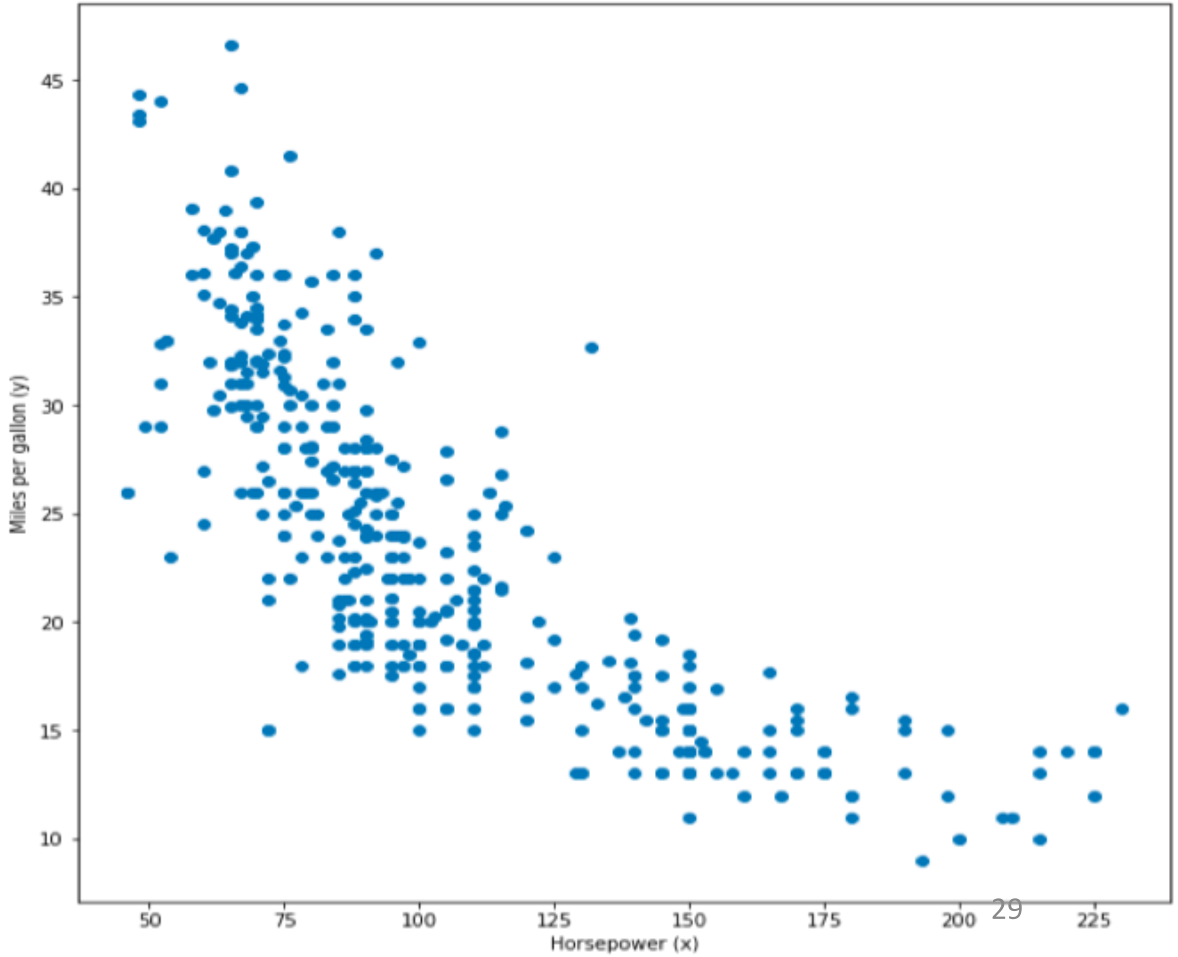
- Quadratic relationship: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2$
  - This is still a linear model: Y= $\beta_0 + \beta_1 X_1 + \beta_2 X_2$
    - But set $X_2 = X_1^2$

- Cubic relationship:
  We can define Y= $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$
  - Let $X_2 = X_1^2$ and $X_3 = X_1^3$ ➔ Y = $\beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3$

Polynomial Regression with Auto Dataset

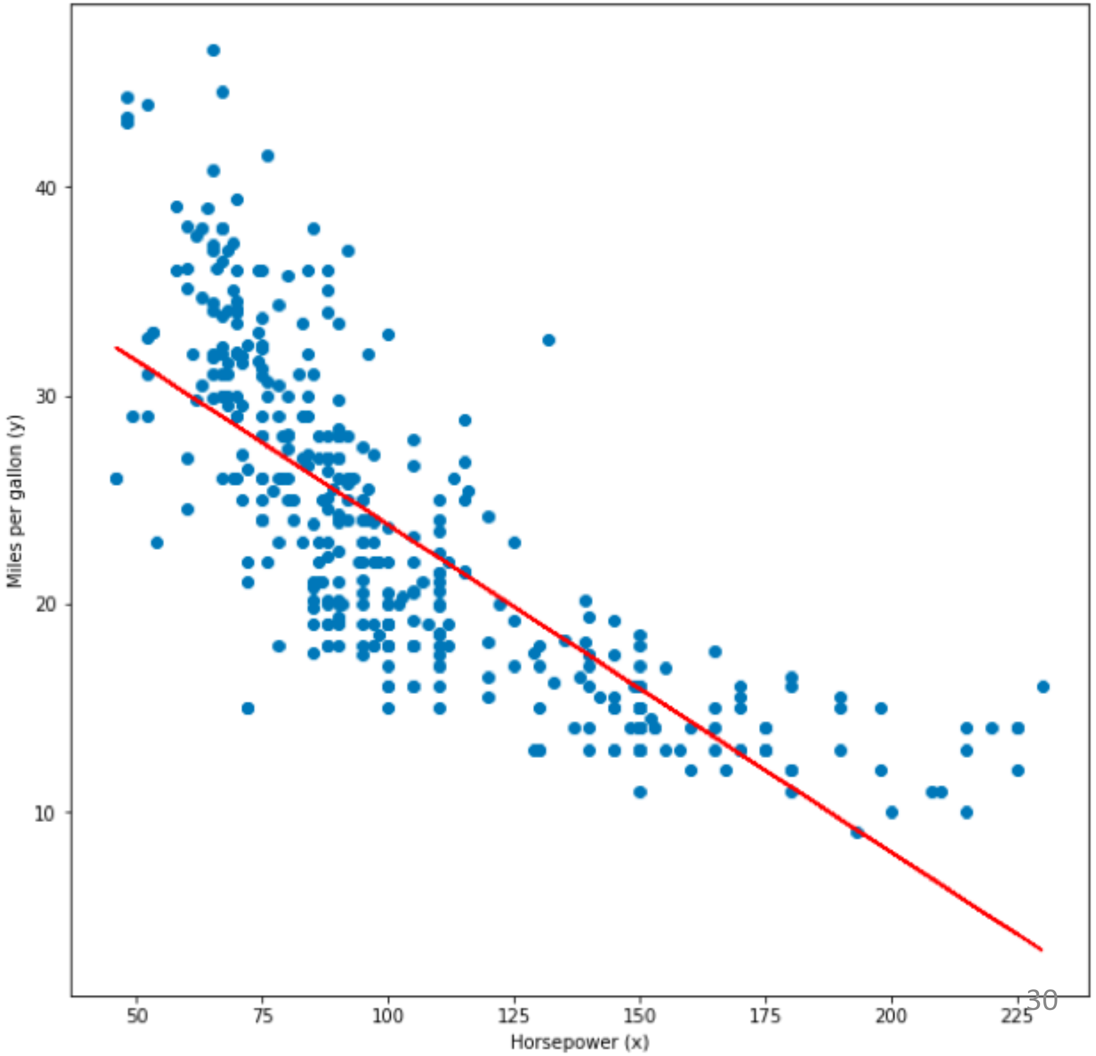
- Auto dataset includes the miles per gallon (mpg) and horse power for a number of cars http://www-bcf.usc.edu/~gareth/ISL/data.html
- It is clear that relationship is not linear

Miles per gallon



- If we fit linear model with only horsepower feature, we get

$$\mathtt{mpg} = \beta_0 + \beta_1 \times \mathtt{horsepower} \ + \epsilon$$



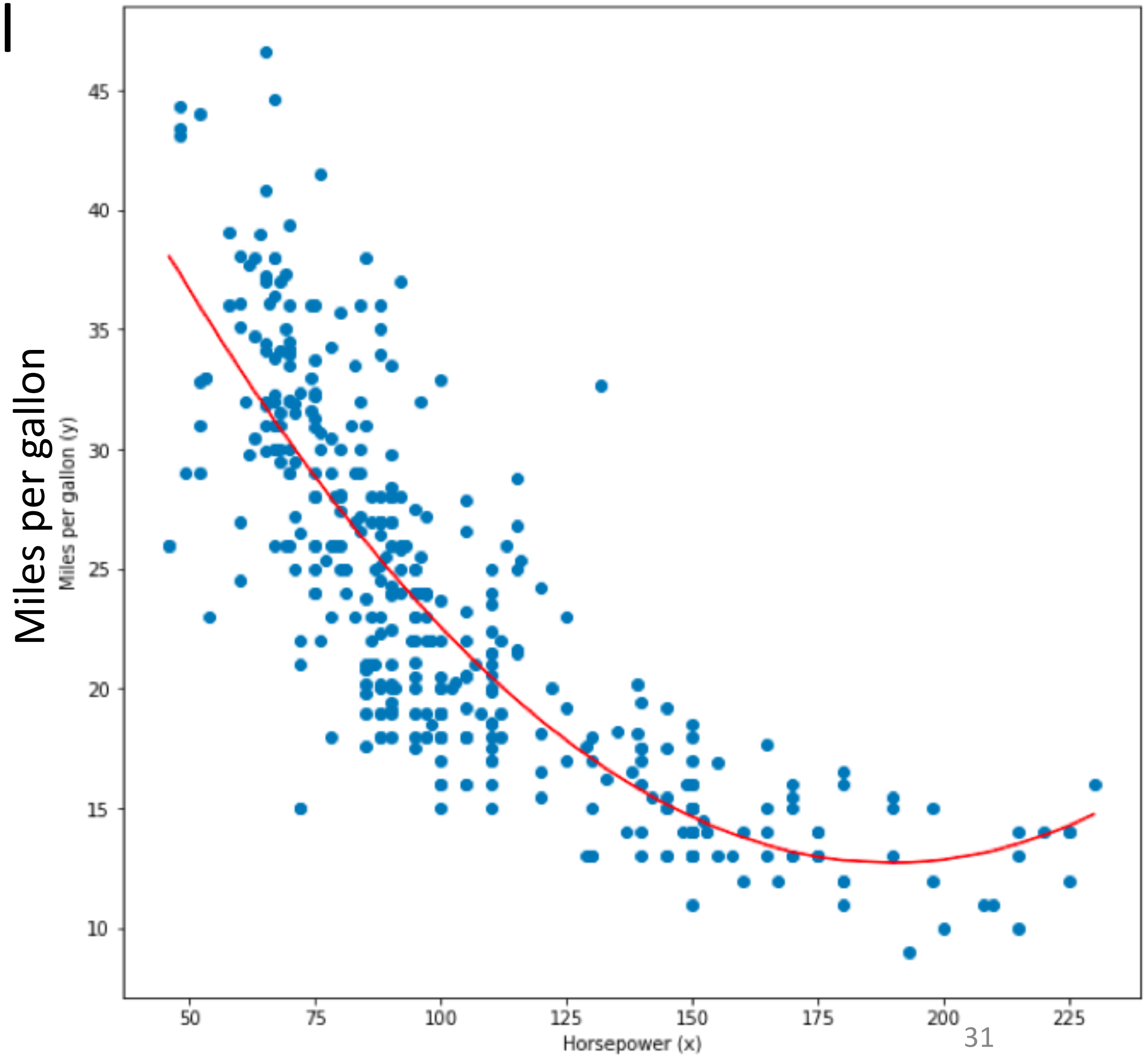
- R^2 metric is 0.6

Polynomial Regression with Auto Dataset

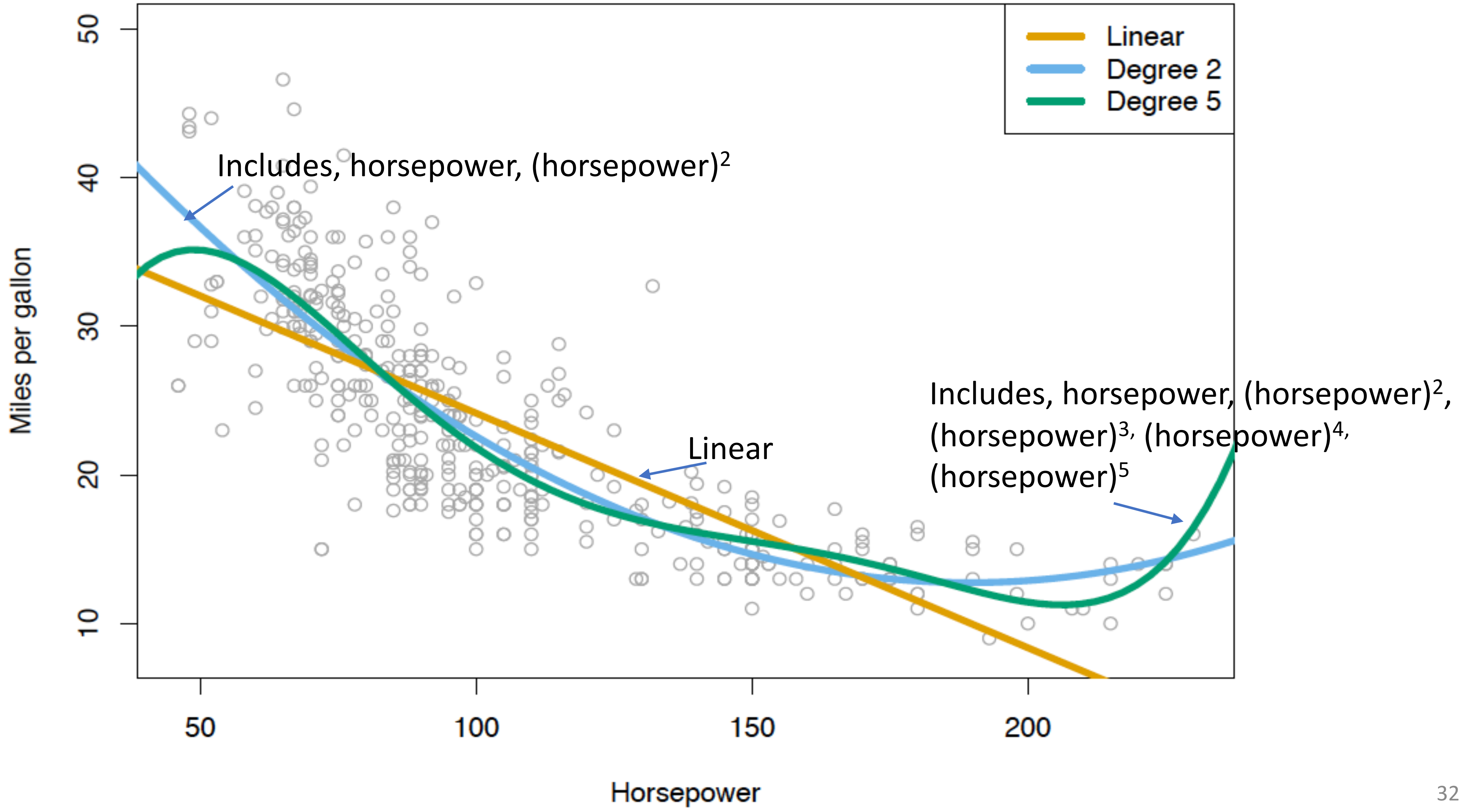
- Adding quadratic term to the linear model

$$\texttt{mpg} = \beta_0 + \beta_1 \times \texttt{horsepower} + \beta_2 \times \texttt{horsepower}^2 + \epsilon$$

	Coefficient	Std. Error	t-statistic	p-value
Intercept	56.9001	1.8004	31.6	< 0.0001
horsepower	-0.4662	0.0311	-15.0	< 0.0001
$\texttt{horsepower}^2$	0.0012	0.0001	10.1	< 0.0001



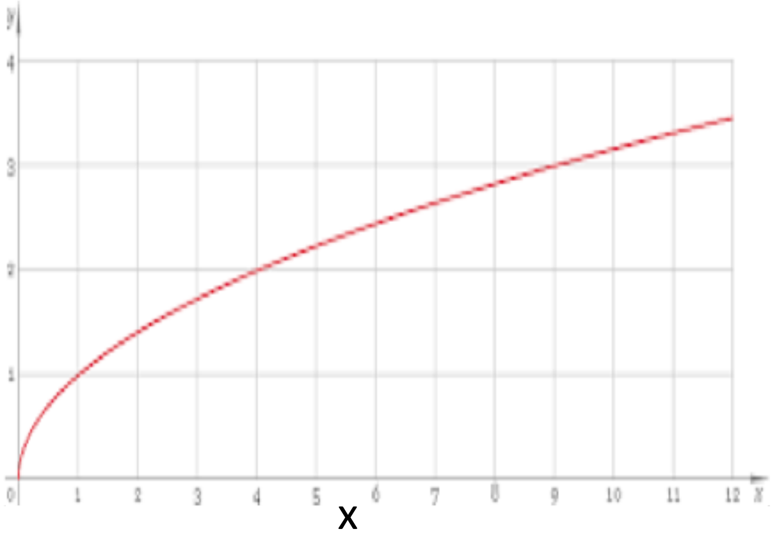
- You can add more terms, i.e. increase the degree of the polynomial
- Examine the output, and make sure to avoid overfitting!



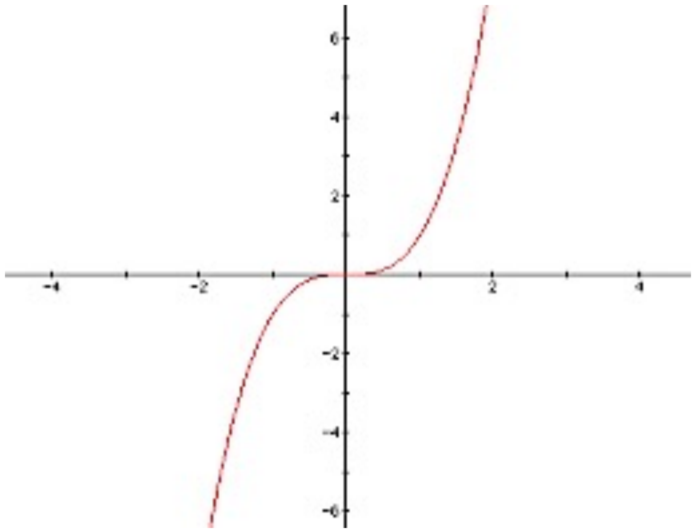
Polynomial Regression

- Other transformations: log(x), $\sqrt{x}$

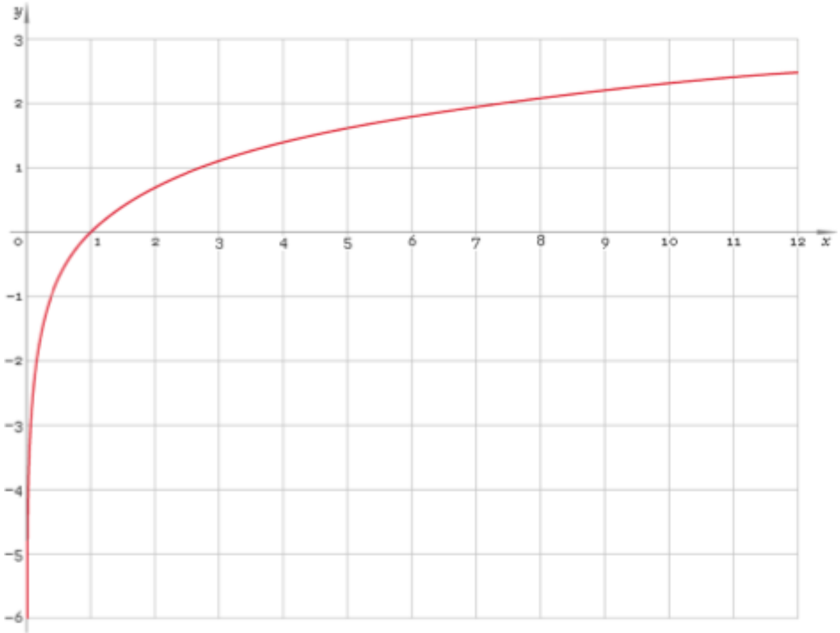
Square root function



Cubic function



Logarithmic function



- Y= $\beta_0 + \beta_1 X_1 + \beta_2\ X_2$
  - $X_2 = \sqrt{X_1}$ ➔ Y = $\beta_0 + \beta_1 X_1 + \beta_2 \sqrt{X_1}$

Parametric vs Non-Paramteric Regression

- Parametric
  - Make strong assumption about f(x). e.g. linear model
  - Easy to fit and understand

- Non-paramteric methods
  - Do not assume any form of f(x), hence are more flexible, e.g. KNN

- Typically, the parametric approach outperforms the non-parametric one if the model selected is close to the true one

KNN Regression

- One of the simplest non-parametric methods is the K-nearest neighbors regression (KNN regression)
- First identifies the K training observations that are closest to the new observation point (x_o) – denote these neighbors by $\mathcal{N}_o$.
- Then estimate f (x_o) as the average of all the training responses in N_o.

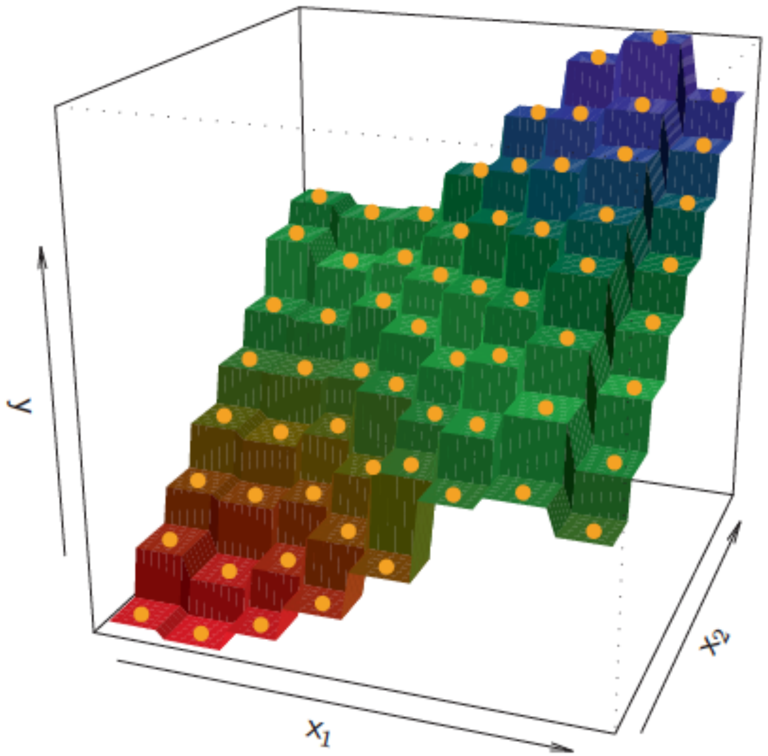
$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_0} y_i$$

In KNN classification, we use majority .. With regression we use average

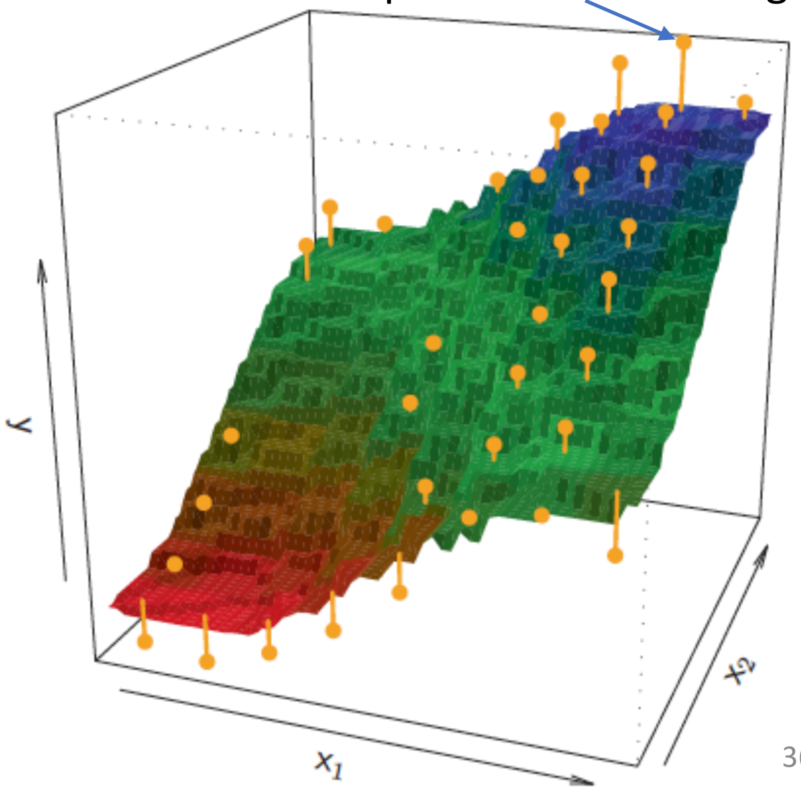
- K=1: output takes the form of steps
  - Output depends on a single observation
- K=9: Smoother function due to the averaging
- Optimal K depends on **bias-variance trade-off**
  - Small k => high variance
  - Large k => high bias and low variance

Dots represents the training points

K=1



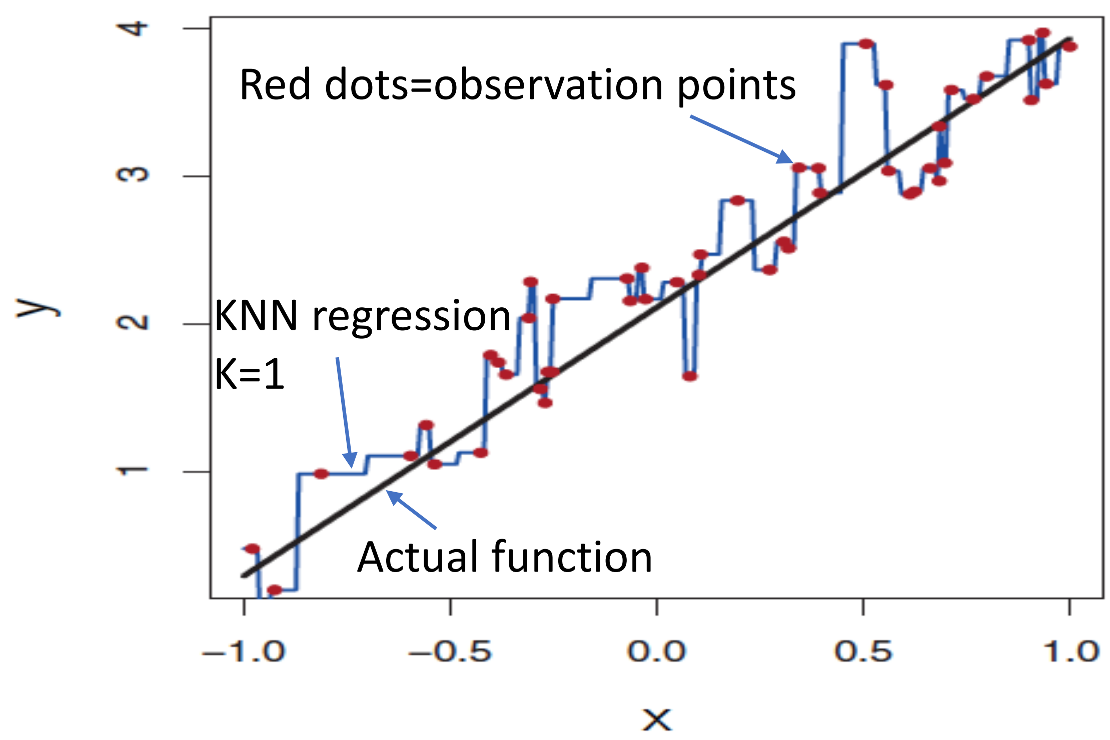
K=9



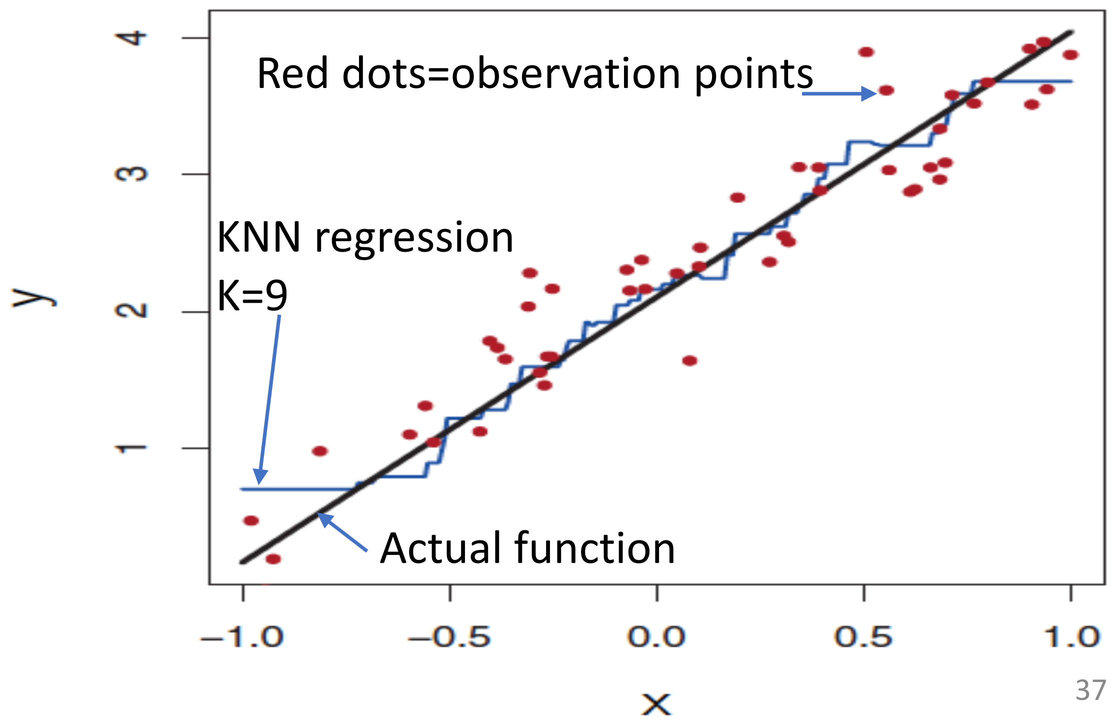
Regression with One Feature

- Assume one feature (1-D),
- data (100 observations) are drawn from linear function, the KNN regression fit with k=1, and k=9

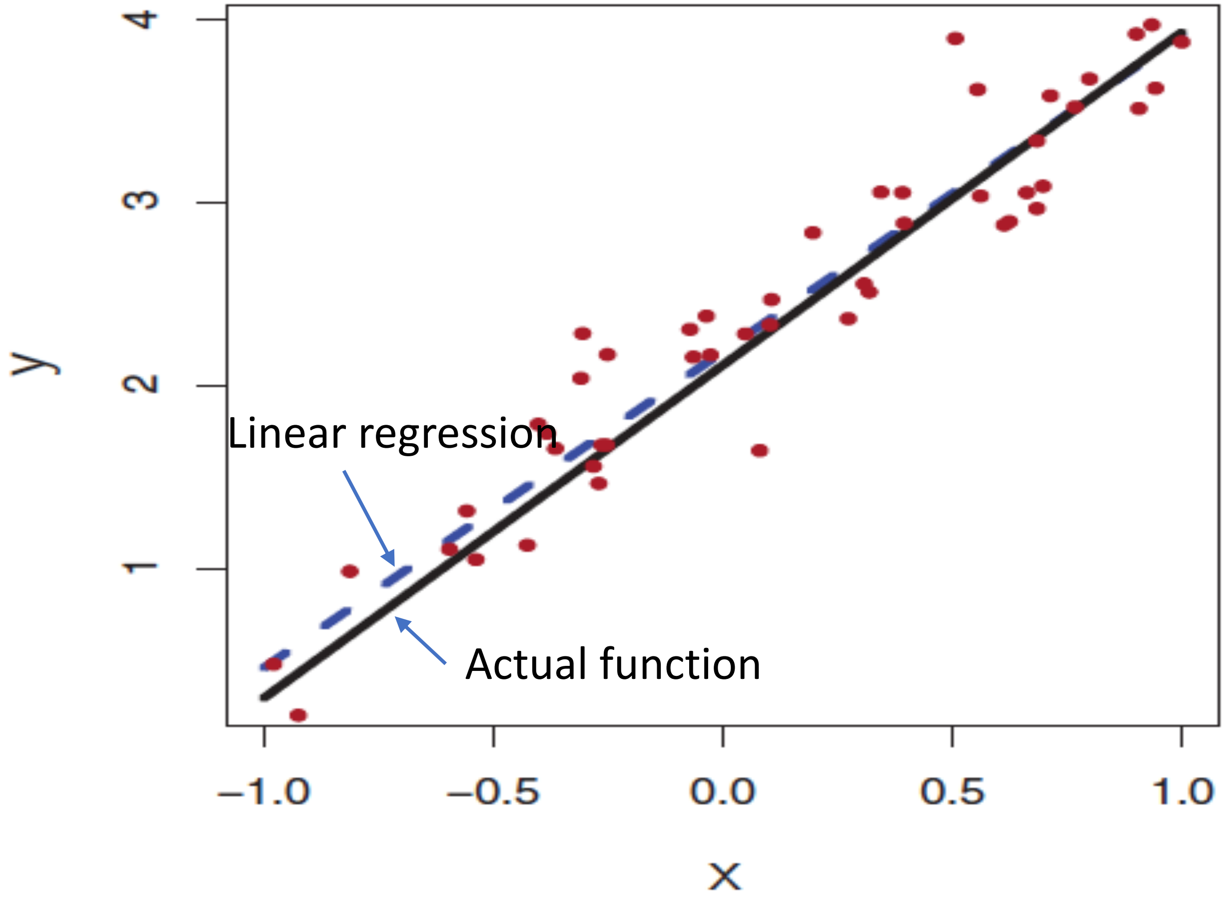
K=1



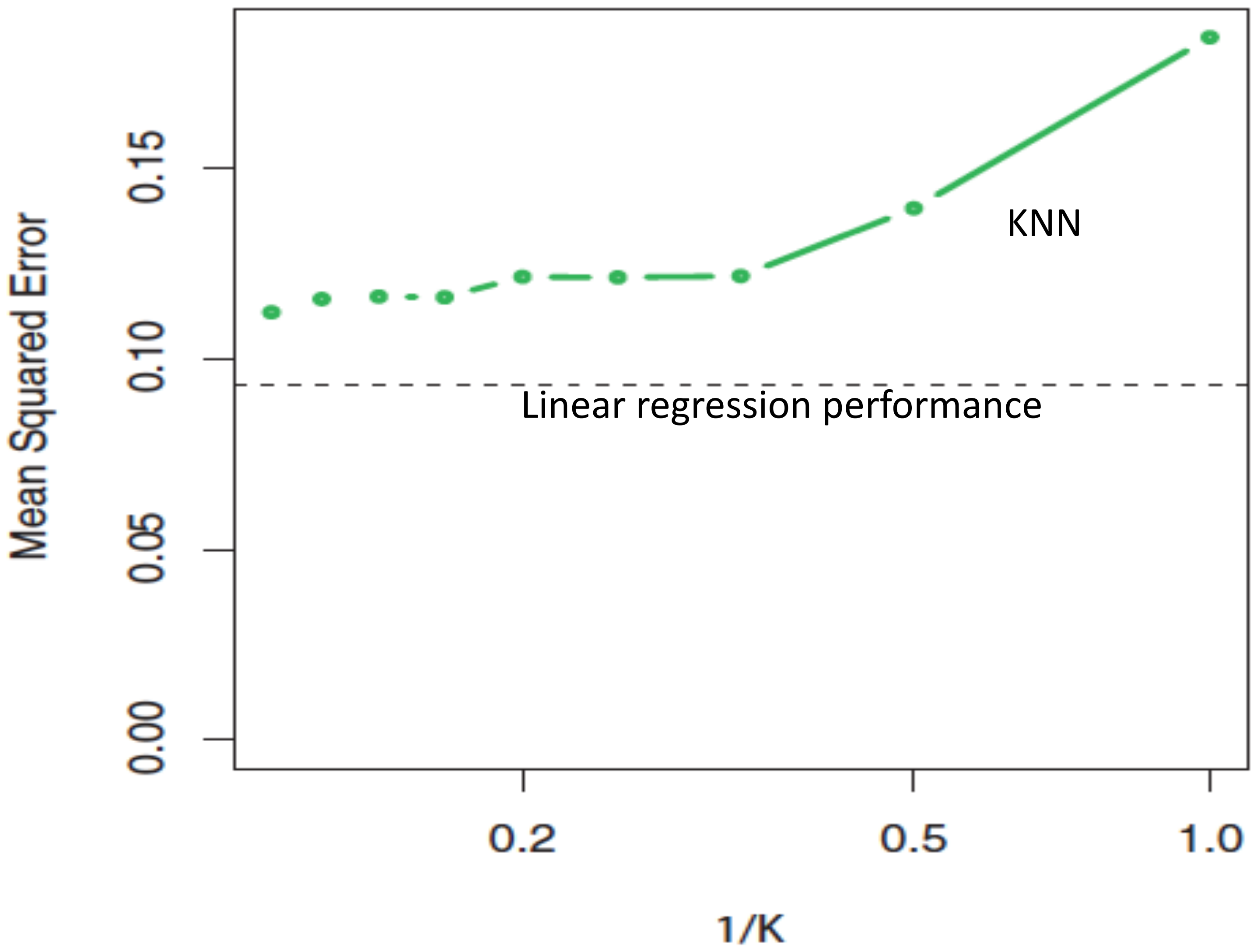
K=9



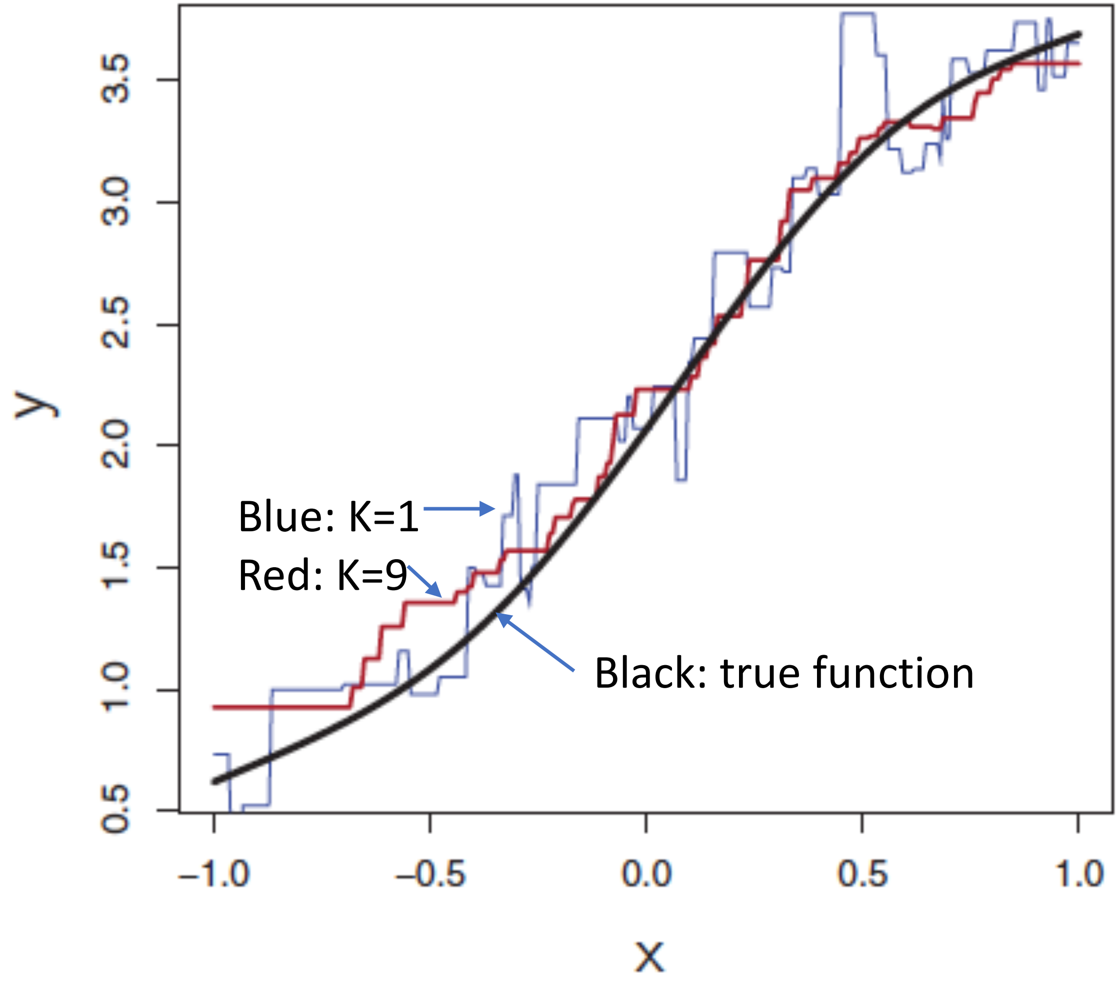
- The parametric approach will outperform the nonparametric approach if the parametric form that has been selected is close to the actual form

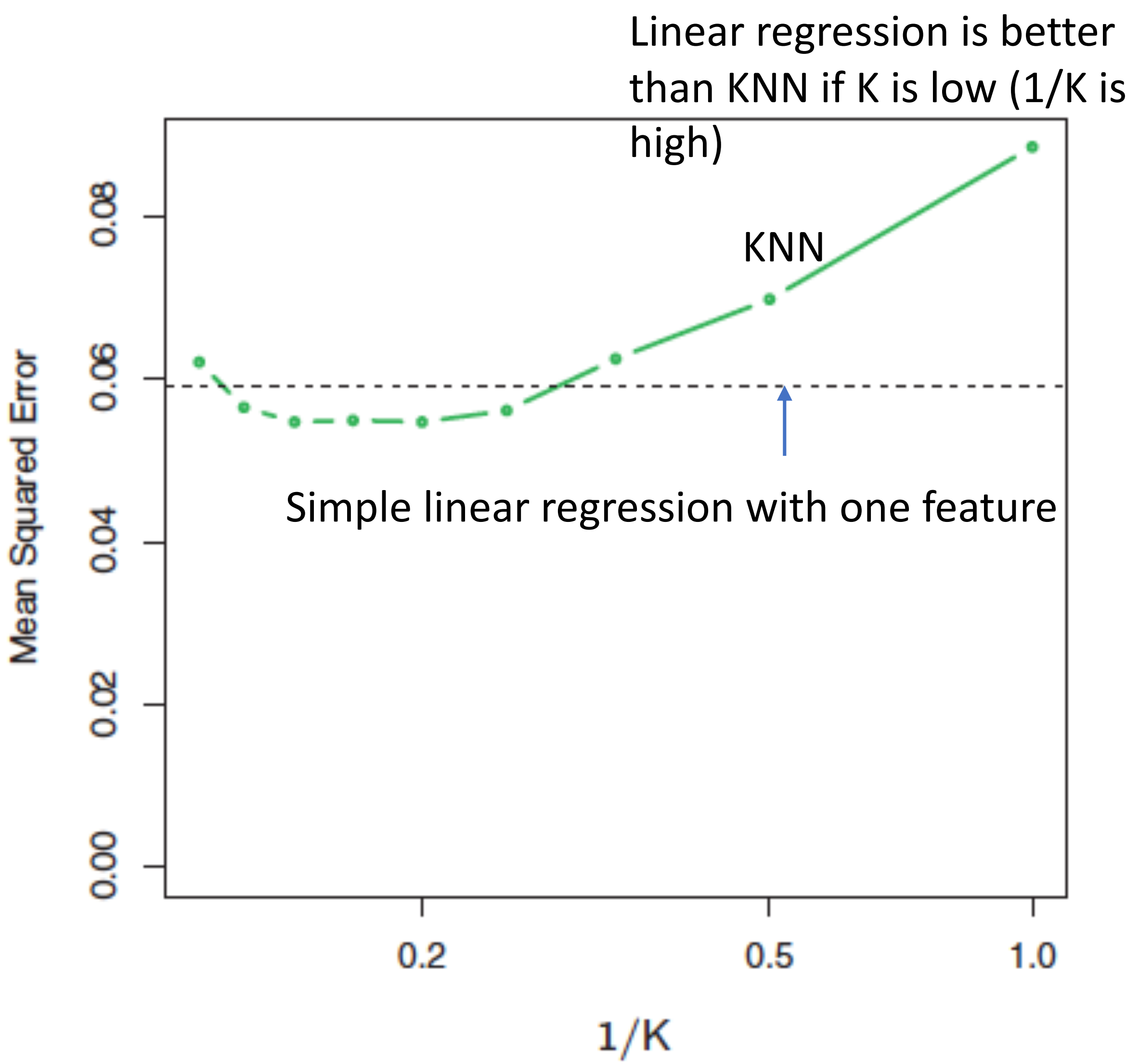


- Since the actual function is linear in this example, linear regression performs better than KNN

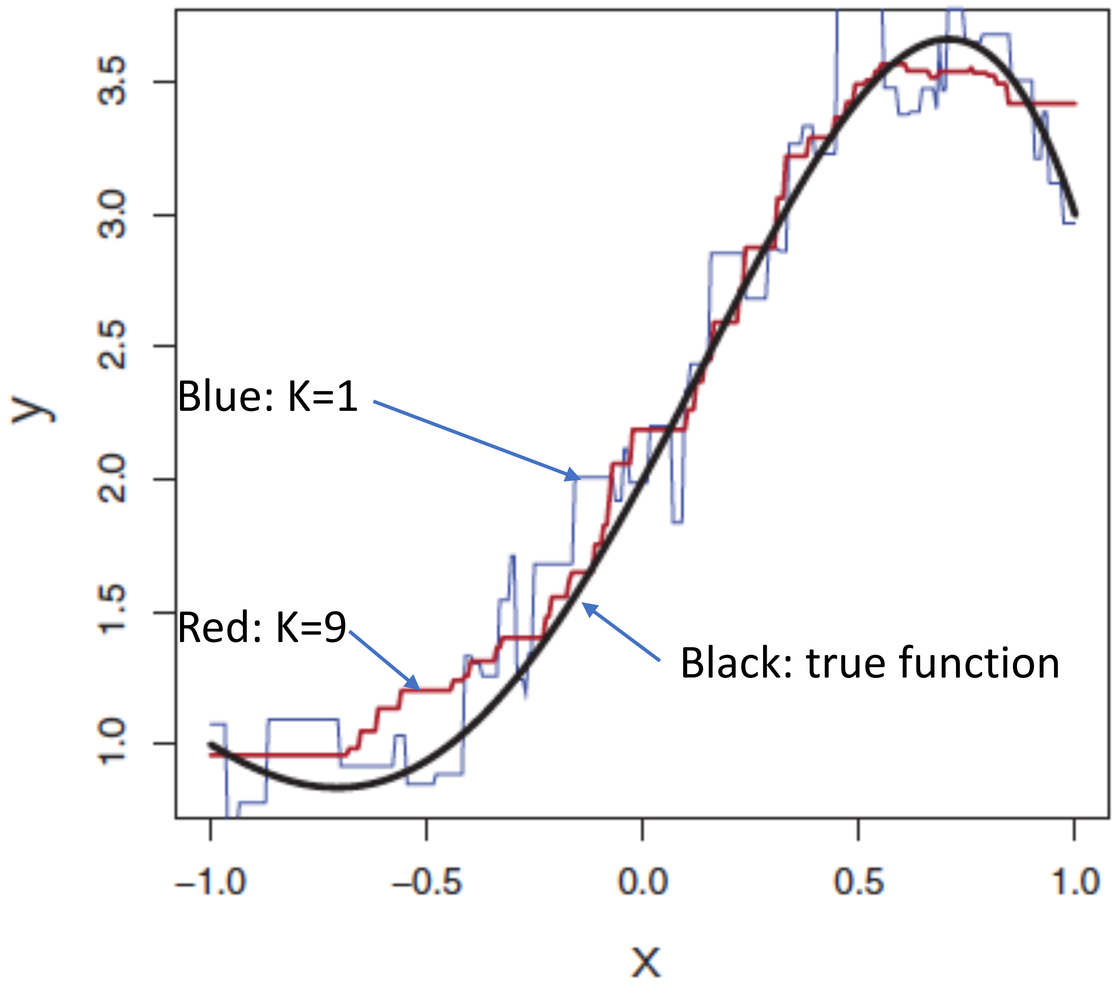


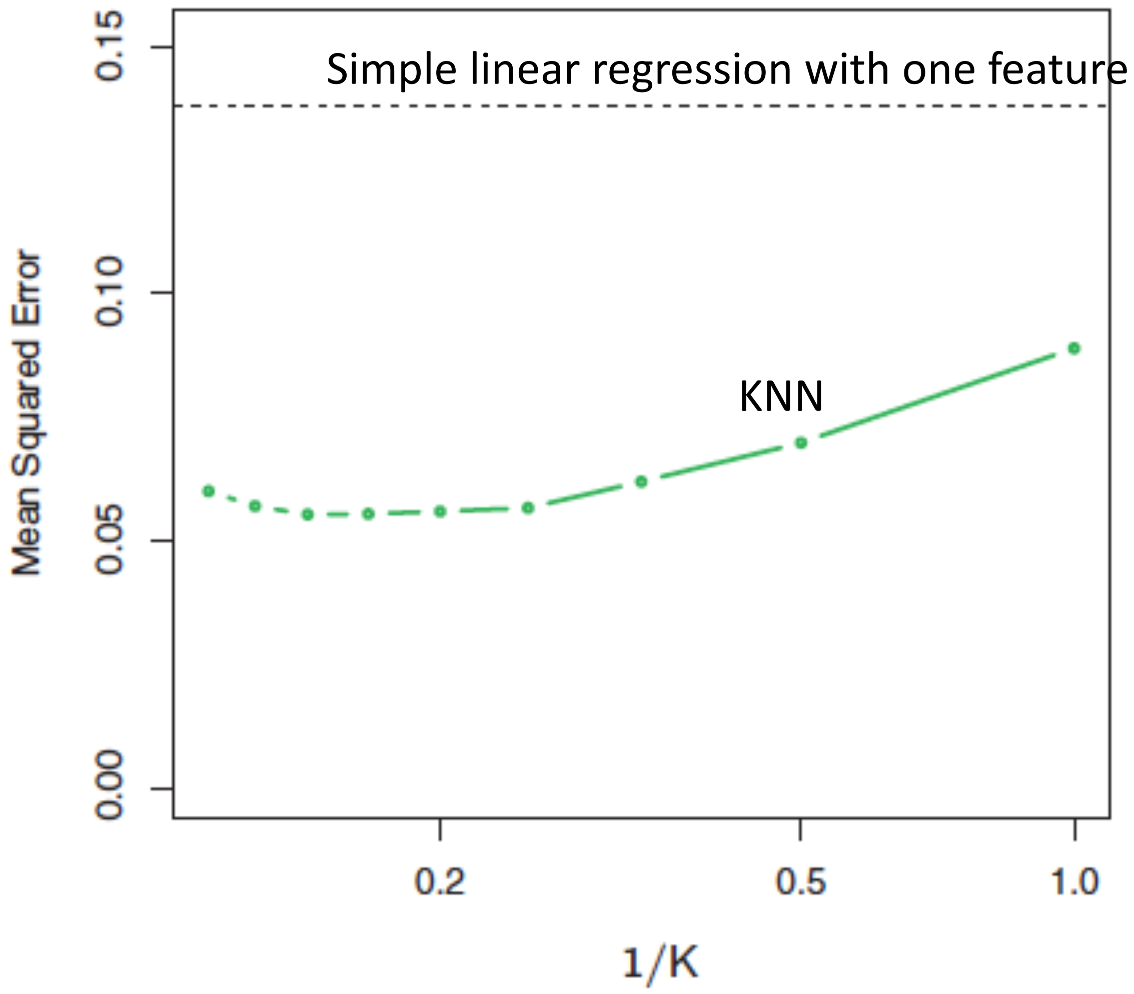
• Another example, where the actual function is not linear





Linear assumption of simple linear regression model works poorly as the data is nonlinear





More Features – Higher Dimensions

- Assume same non linear relationship between feature and response as in the previous slide
- As the number of features increases the KNN performance degrades (common problem)

Here we have **100 observations**

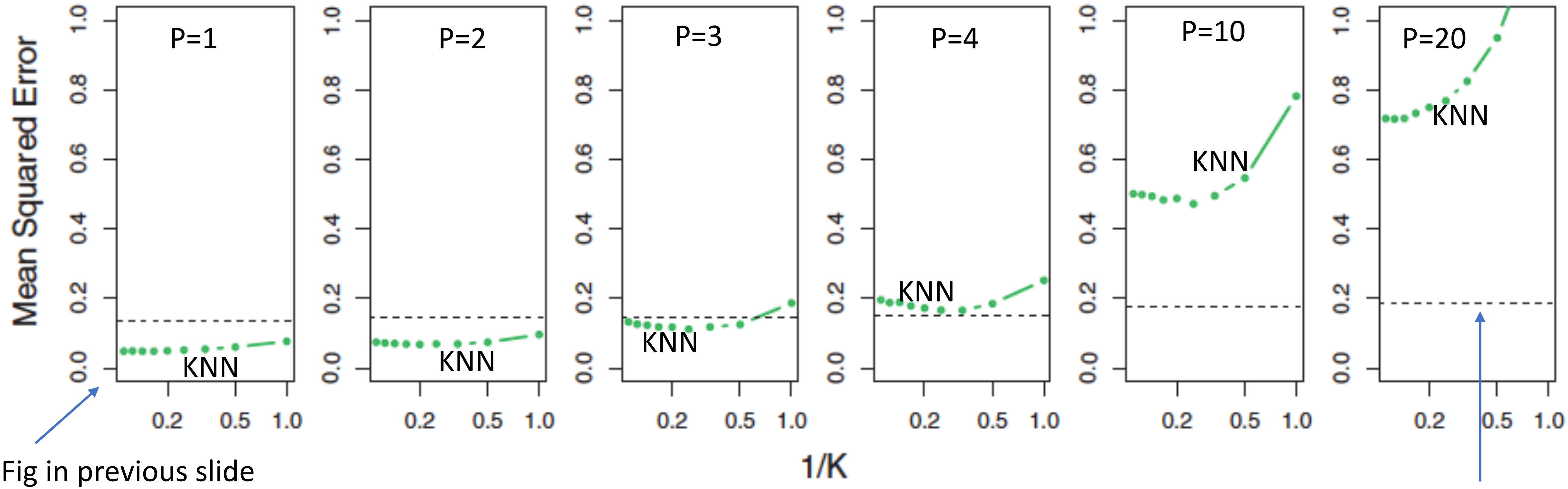


Fig in previous slide

Horizontal black dashed line in all figures is linear regression (assume linear function with all predictors)

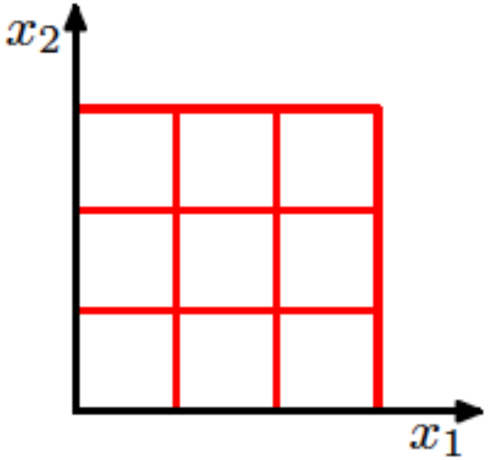
Curse of Dimensionality

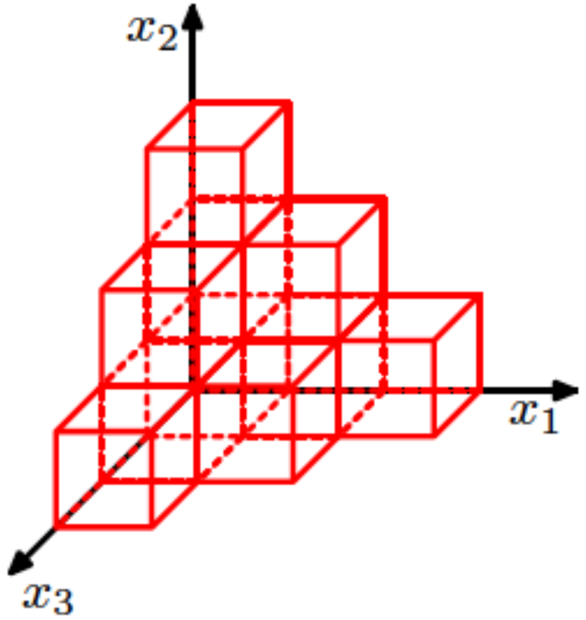
- **Curse of dimensionality**: Number of **training** data needed grows **exponentially** with the number of **features**.
  - Having large number of features and insufficient number of training leads to poor accuracy
  - In previous example:
    - One feature, 100 observations provides sufficient information for estimation
    - 20 features, 100 observations are not sufficient

Curse of Dimensionality

- For KNN regression: the K observations that are nearest to the the new observation (x_o) may be very far from x_o in p-dimensional space when p (number of features) is large
  - More training samples are needed to keep the accuracy

Large number of training data is needed to ensure that regions (in features space) are w





Reference: Bishop, Pattern Recognition

Exercise

The exercise (including dataset) is on courseweb

- Use auto dataset, and fit a linear model to predict the miles per gallon (mpg) from horsepower
  - Hints:
    - Use: **from sklearn.linear_model import LinearRegression**
    - Create model object using: **model=LinearRegression()**
    - Fit the model: **fitted_model=model.fit(X_train,Y_train)**
- Find the R^2 metric: **fitted_model.score(X_test,Y_test)**

Exercise – cont.

- Polynomial regression: Find the R^2 metric when we include both the horsepower feature and $(\text{horsepower})^2$

    - Hint:
        - You can use **numpy.concatenate** to define feature vector

        https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.concatenate.html

        Don’t forget to import numpy

- Optional: increase the degree of the polynomial to 3,then 4, then 5 and check the accuracy in each case

Exercise – cont.

- Repeat using KNN regression, with K=7. That is, find R^2 metric in the following cases

1. One feature: Horsepower only
2. Two features: horsepower and $(\text{horsepower})^2$

  - Hint: In python, create KNN regression object using **neighbors.KNeighborsRegressor:**

    from sklearn import neighbors
    knnRegression = **neighbors.KNeighborsRegressor**(n_neighbors=7)

    Then use the **.fit** and **.score** methods as before

- COMMENT on your results: which model performs better? How does performance change when adding the quadratic feature?