University of Pittsburgh

ECE 2195: Special Topics – Computers Machine Learning

Extension to Linear Regression Model & Comparison with KNN regression

Mai Abdelhakim, PhD

ECE Department

Swanson School of Engineering

University of Pittsburgh

maia@pitt.edu

This unit

- 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

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
- To represent the gender feature, we 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-value of the dummy variable is high➔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

$estimate\ of\ \beta_0$

$estimate\ of\ \beta_1$ ➔ 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?

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 of 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:

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

Python command: model=smf.ols('Balance ~ Ethnicity ', credit)

	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
  - E.g.: we assumed that sales increases with TV budget regardless of 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
  - Increase of spending on radio advertising (X_1) may increase slop of TV (X_2) with Sales (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

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 – Relaxed model

- Relax additive assumption by including interaction term

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

- *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)

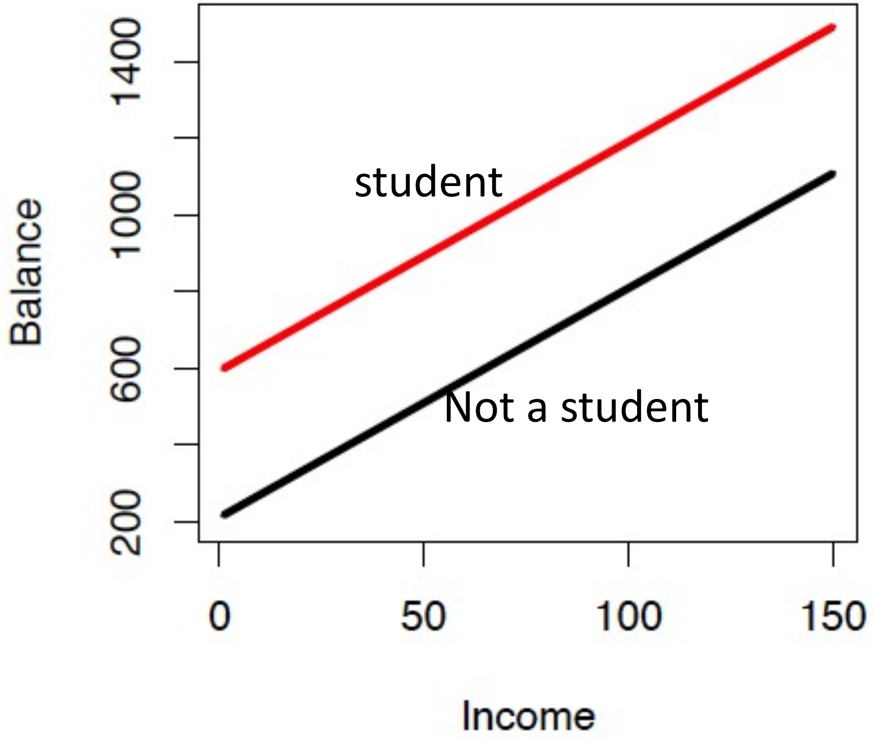
	Coefficient	Std. error	t-statistic	p-value
Intercept	6.7502	0.248	27.23	< 0.0001
TV	0.0191	0.002	12.70	< 0.0001
radio	0.0289	0.009	3.24	0.0014
TV×radio	0.0011	0.000	20.73	< 0.0001

- **Interaction term (TV x Radio) has low p-value indicating that the actual relationship is not additive**
  - Increase spending on radio advertising increase slope of TV

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

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

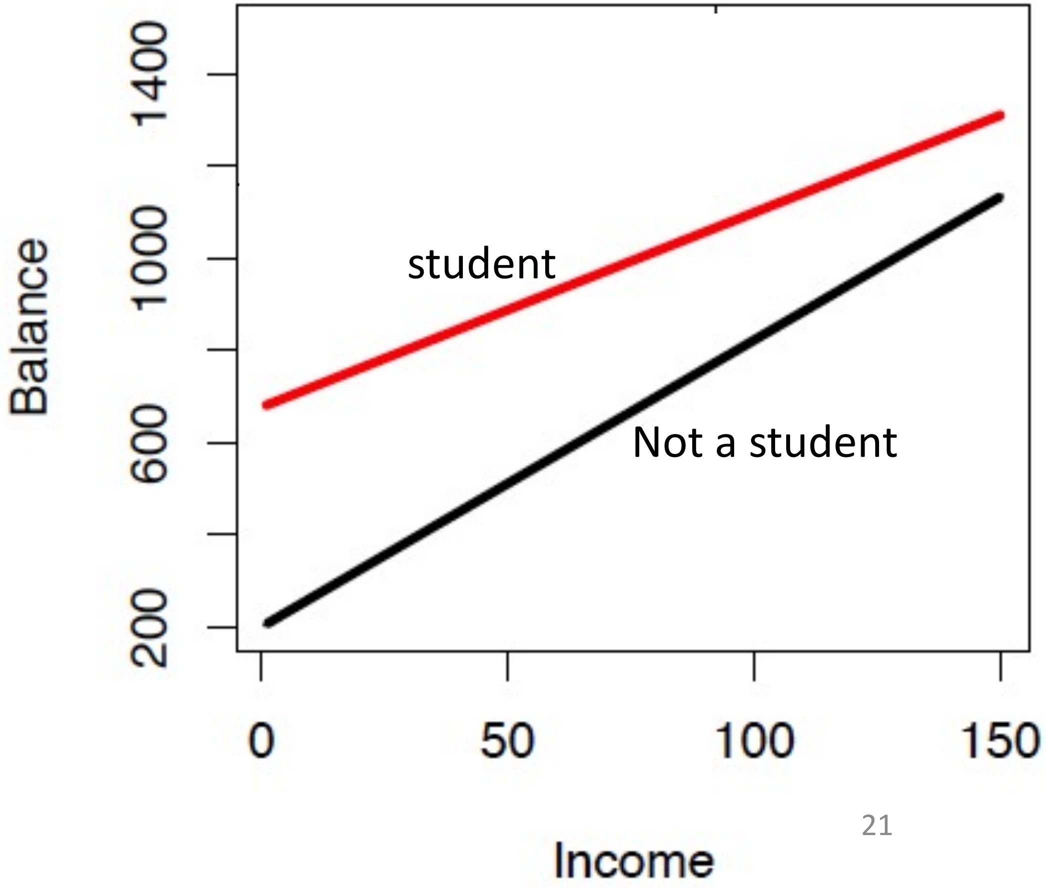


With this model we have same slope for income, but different intercept for student status

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

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

- 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.

Fit with multiple features including interaction terms

- Use the credit data set, fit OLS model to predict credit balance using all of the following features

  - Student
  - Income
  - Limit
  - Interaction term: Income*Student
  - Interaction term: Limit*Student

Find the p-values of all features. Are they all helpful in predicting the response?

```
model=smf.ols('Balance ~ Student+Income+Income*Student+Limit +Limit*Student', credit)
```

- To use qualitative feature in Scikit-Learn, you need to encode the qualitative variable
  After reading the credit data use using:

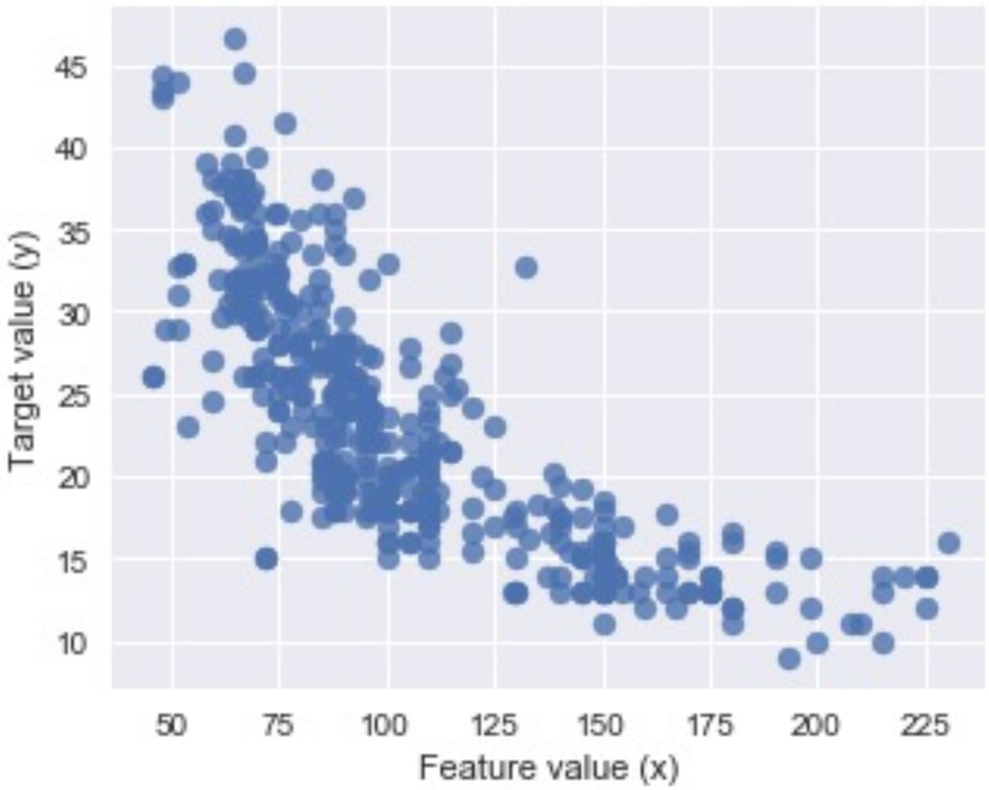
credit =read_csv('Credit2.csv’)

Add a column to the data set that is 0 for not student and 1 for students

credit['StudentEncode'] = credit.Student.map({'No':0 , 'Yes':1})

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

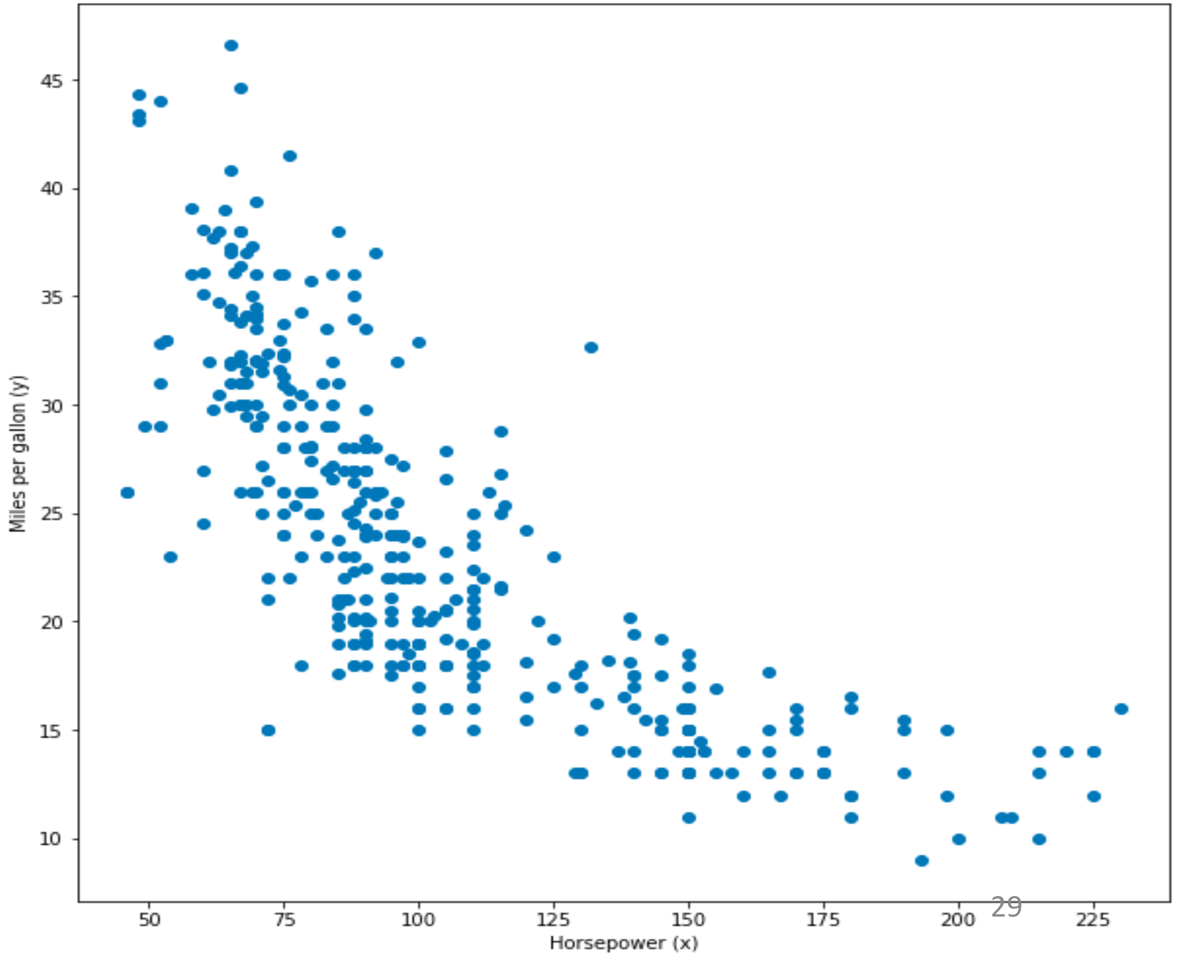
- Quadratic relationship: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2$
  - This is still a linear model: Y= $\beta_0 + \beta_1 X_1 + \beta_2\ X_2$
    - But set $X_2 = X_1^2$

- 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

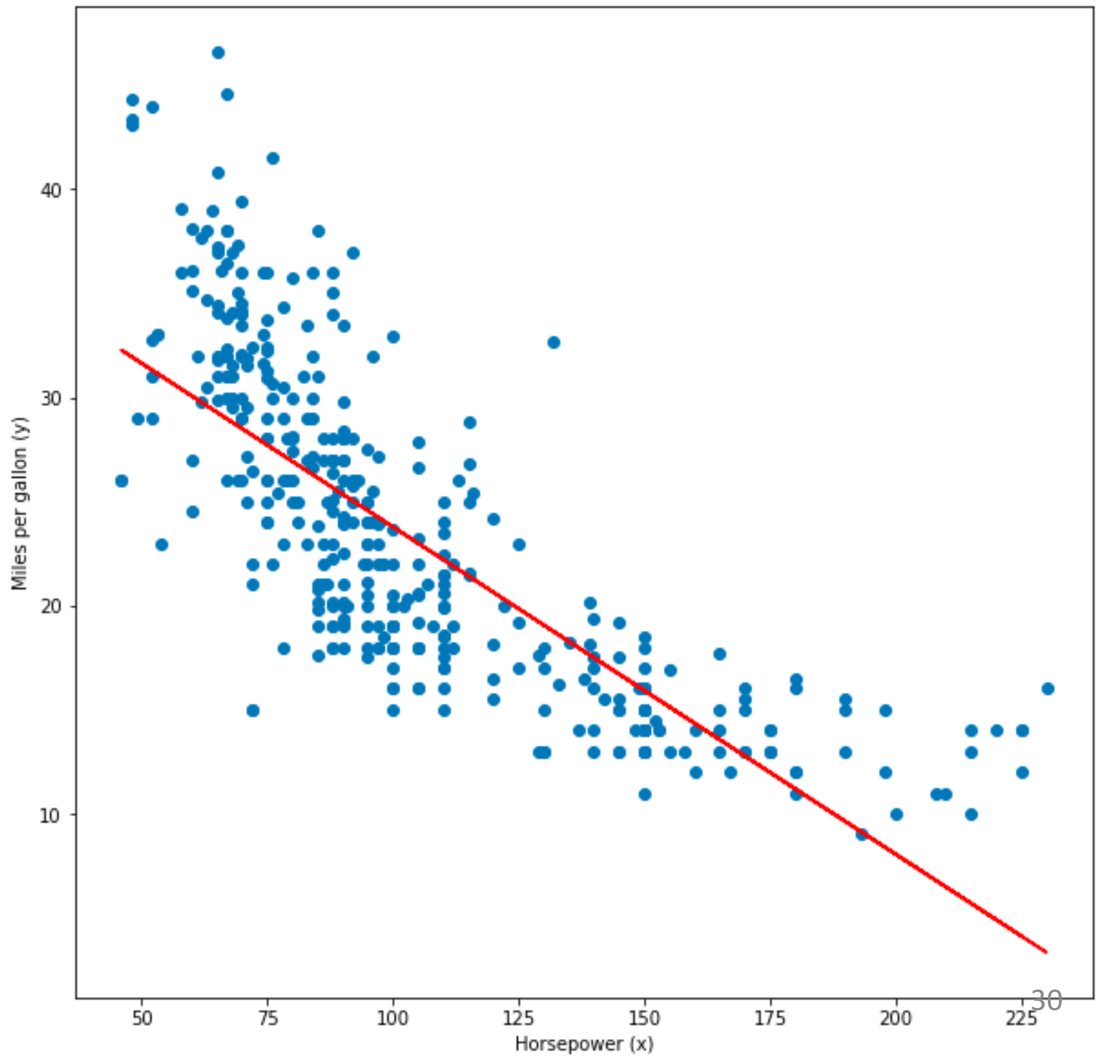
- Auto dataset includes the miles per gallon (mpg) and horse power for a number of cars http://www-bcf.usc.edu/~gareth/ISL/data.html
- It is clear that relationship is not linear

Miles per gallon



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

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



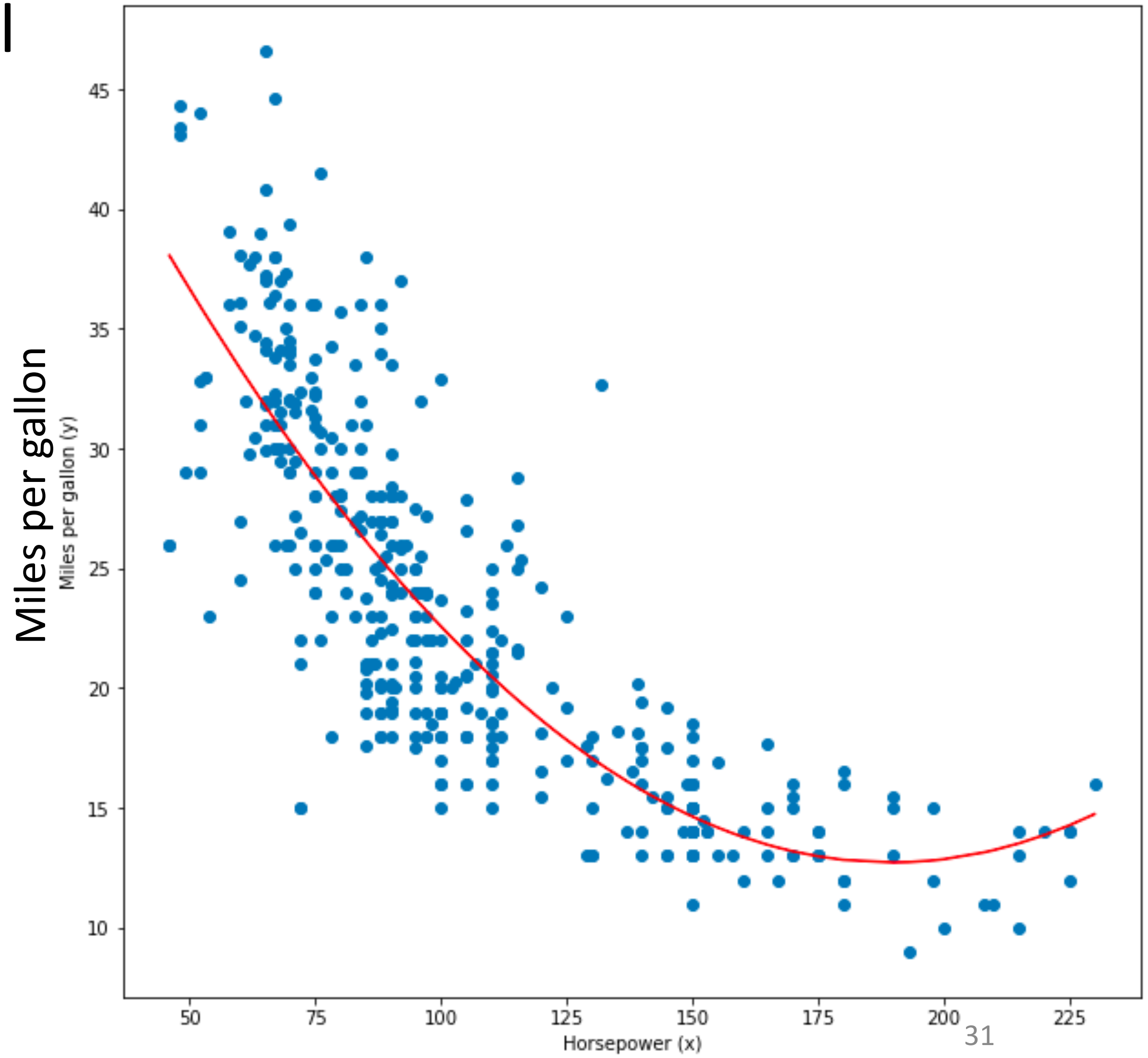
- R^2 metric is 0.6

Polynomial Regression with Auto Dataset

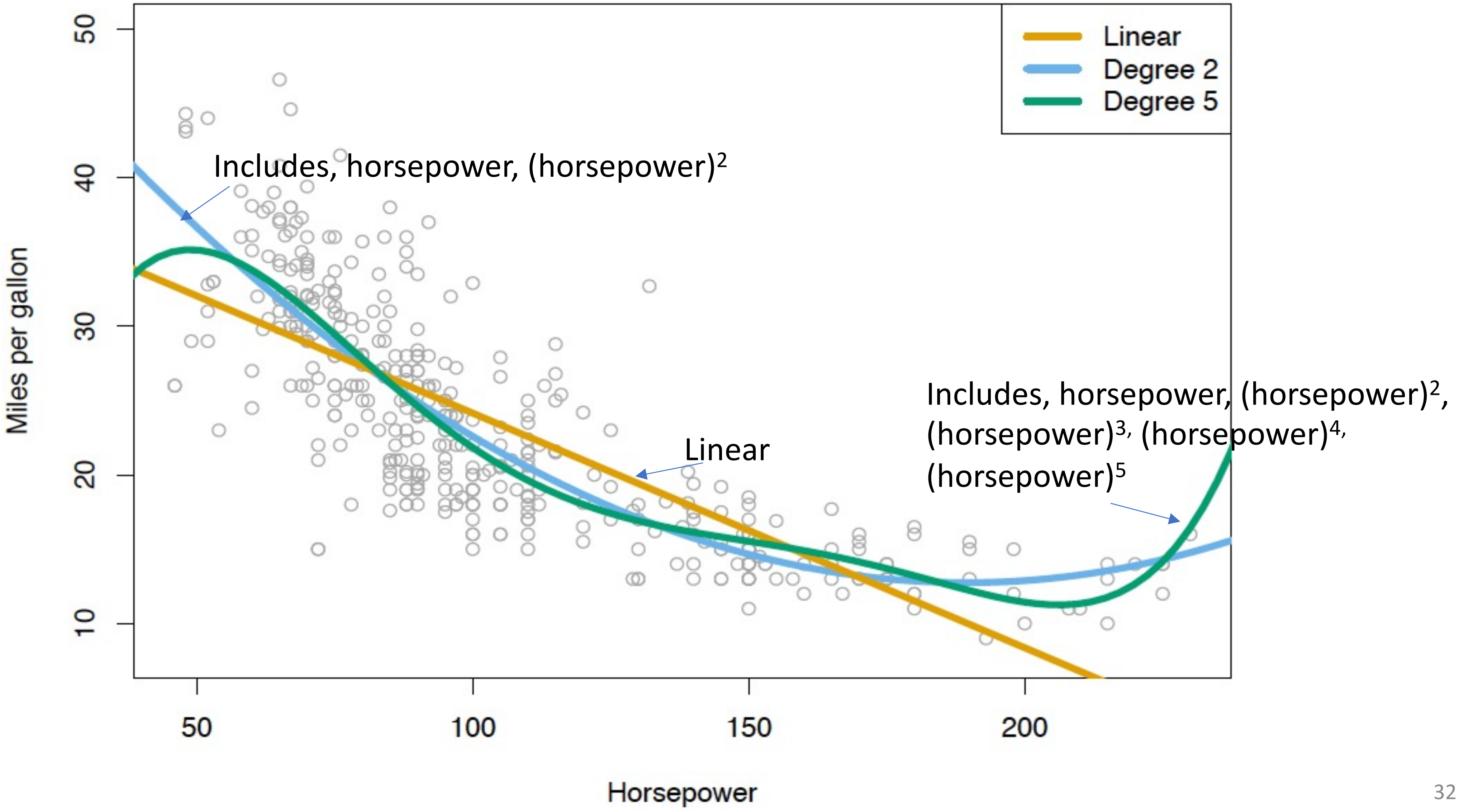
- Adding quadratic term to the linear model

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

	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



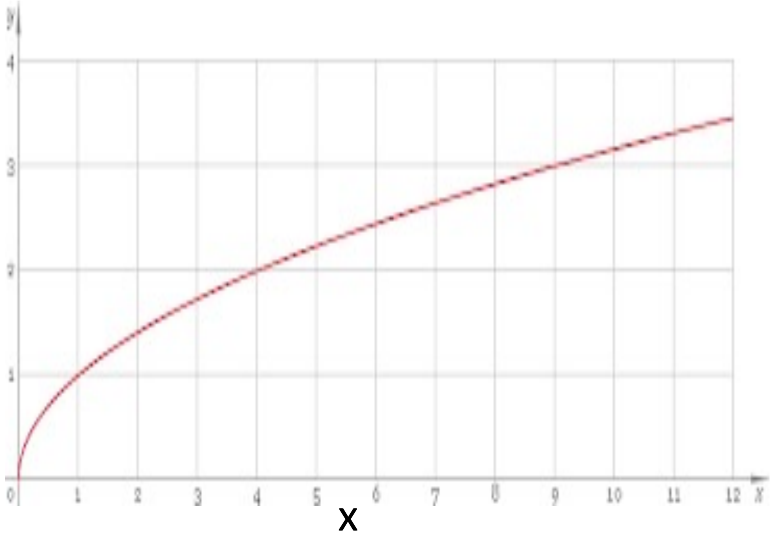
- You can add more terms, i.e. increase the degree of the polynomial
- Examine the output, and make sure to avoid overfitting!



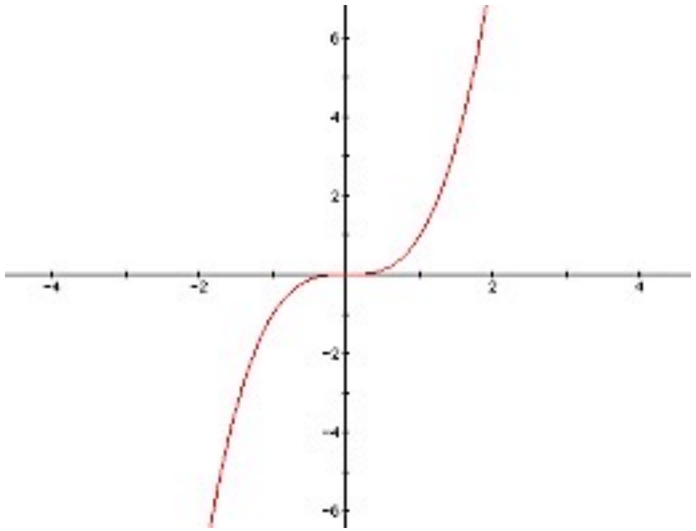
Polynomial Regression

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

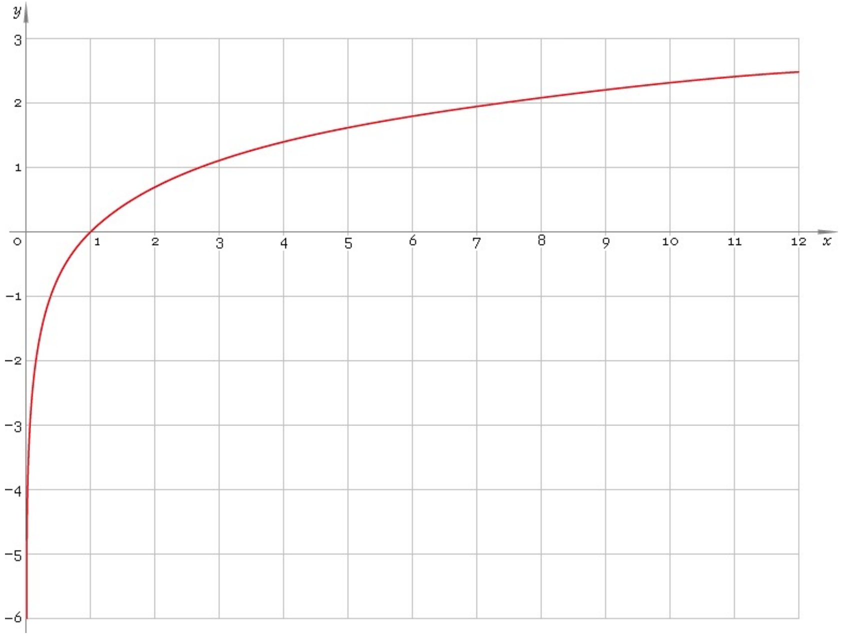
Square root function



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-Parametric Regression

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

- Non-parametric 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 <u>**if**</u> the model selected is close to the true one

KNN Regression

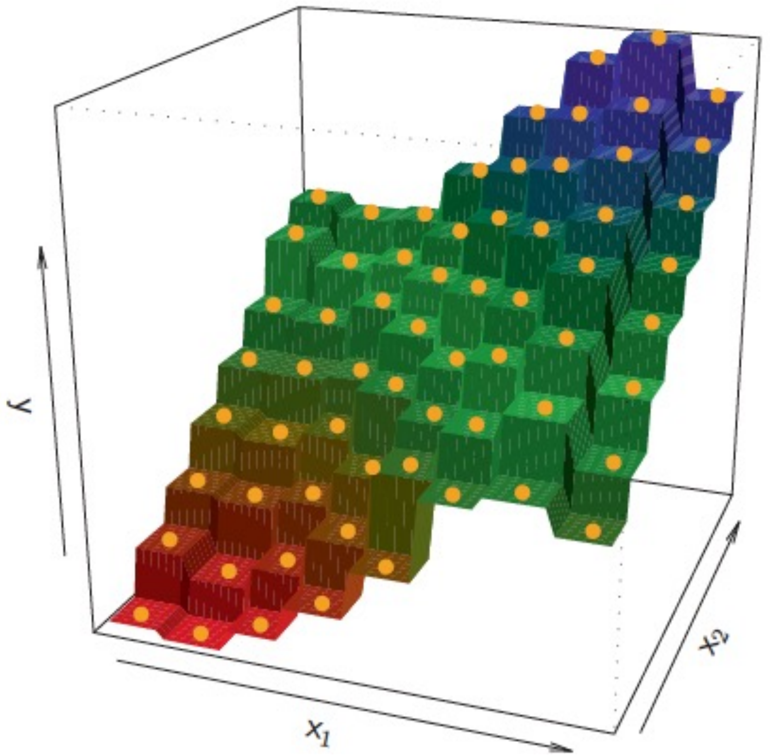
- One of the simplest non-parametric methods is the K-nearest neighbors regression (KNN regression)
- First identifies the K training observations that are closest to the new observation point (x_o) – denote these neighbors by $\mathcal{N}_o$.
- Then estimate f (x_o) as the average of all the training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

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

Note that in KNN classification, we use majority .. With regression we use average

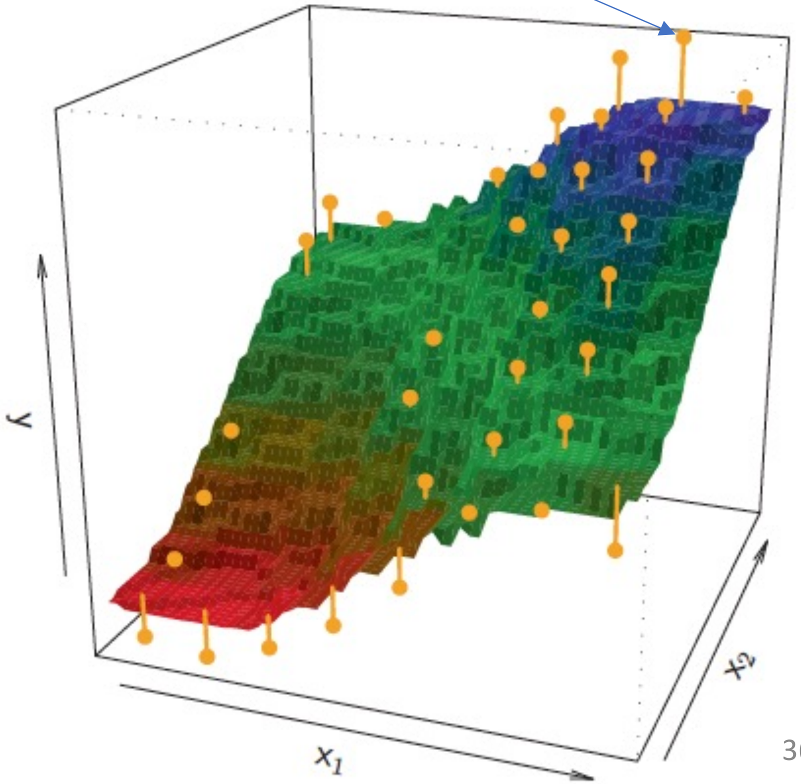
- K=1: output takes the form of steps
  - Output depends on a single observation
- K=9: Smoother function due to the averaging
- Optimal K depends on **bias-variance trade-off**
  - Small k => high variance
  - Large k => high bias and low variance

K=1



K=9

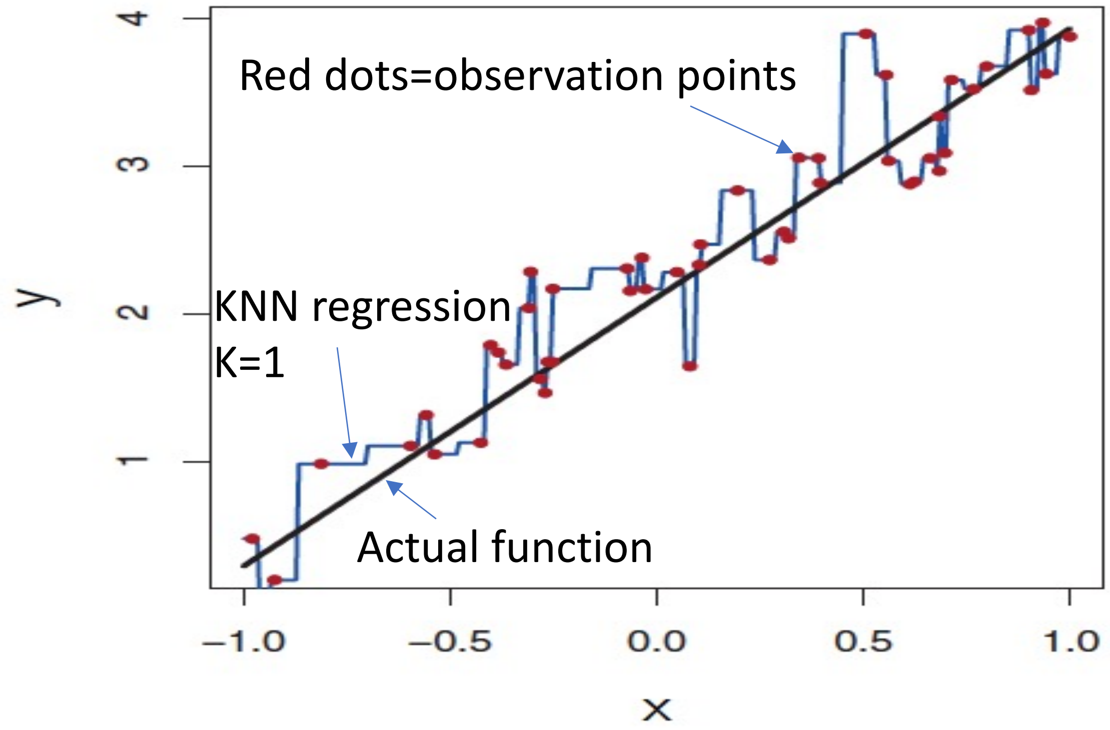
Dots represents the training points



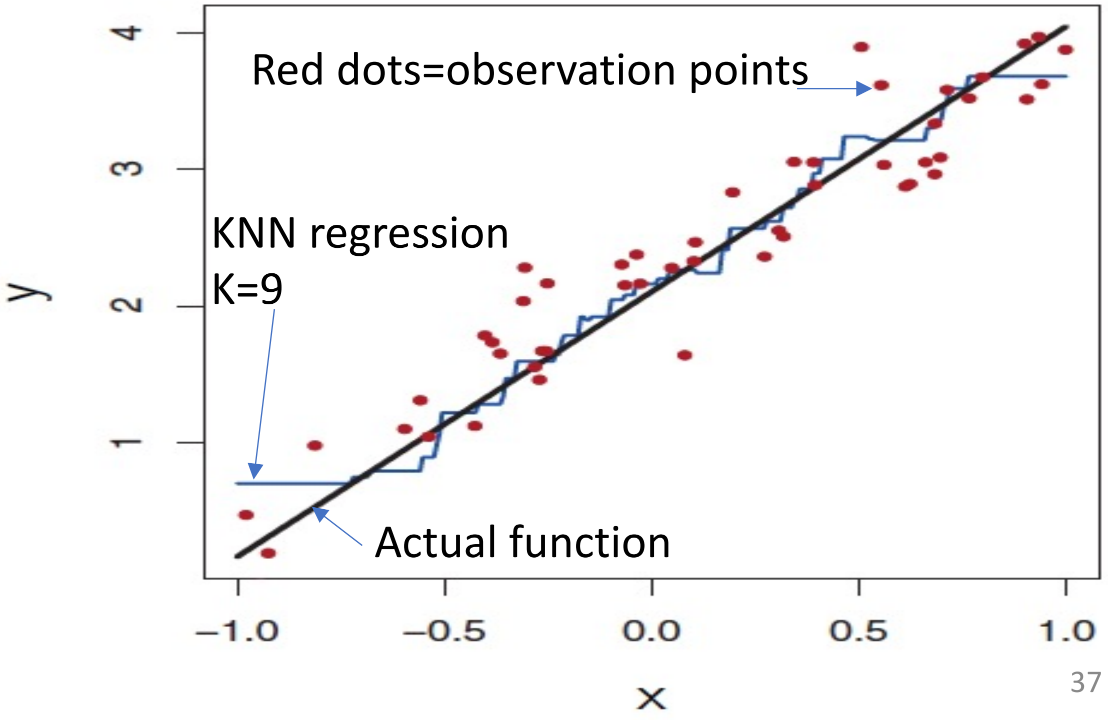
Regression with One Feature

- Assume one feature x (1-D),
- Data (100 observations) are drawn from linear function Y= β_0+$\beta_1 X_1$, the KNN regression fit with k=1, and k=9

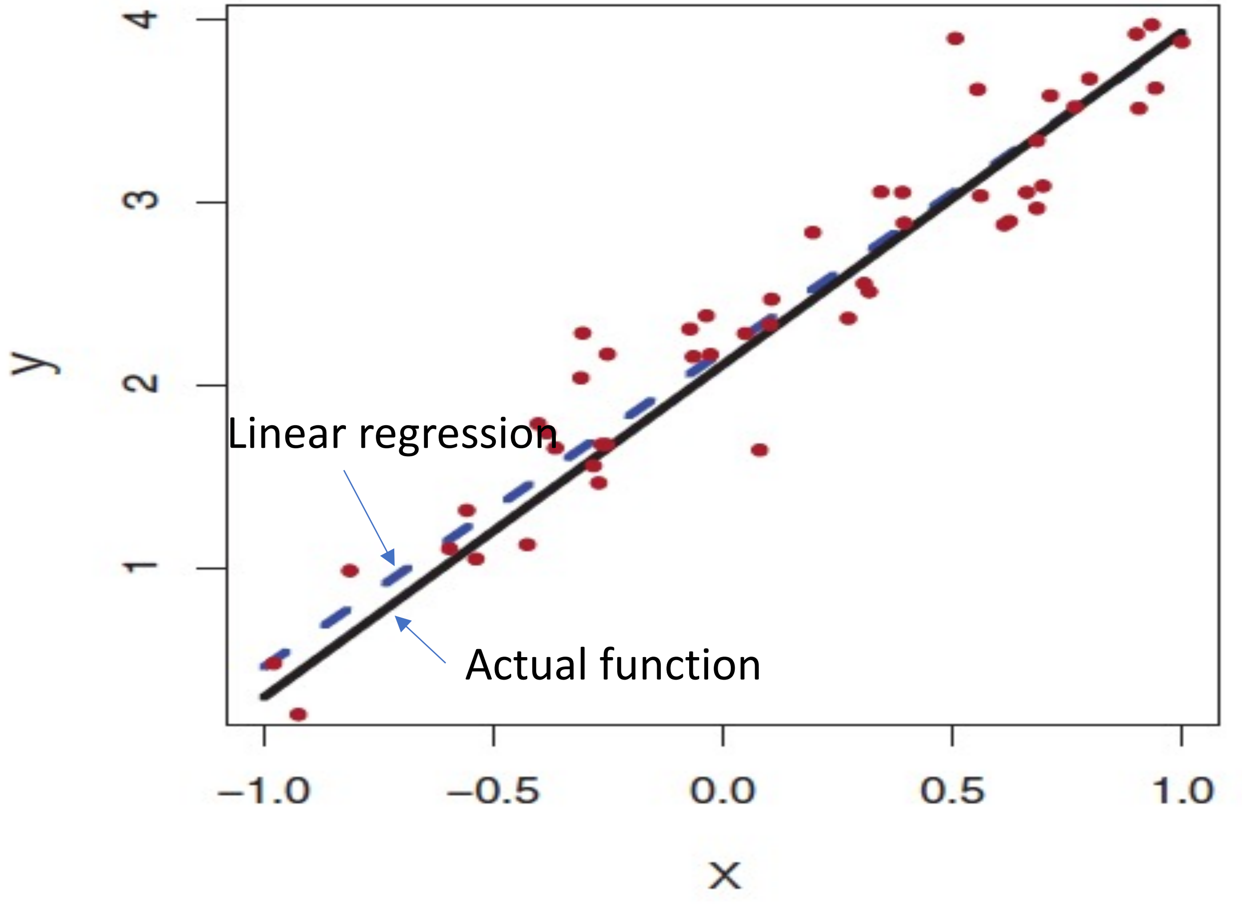
K=1



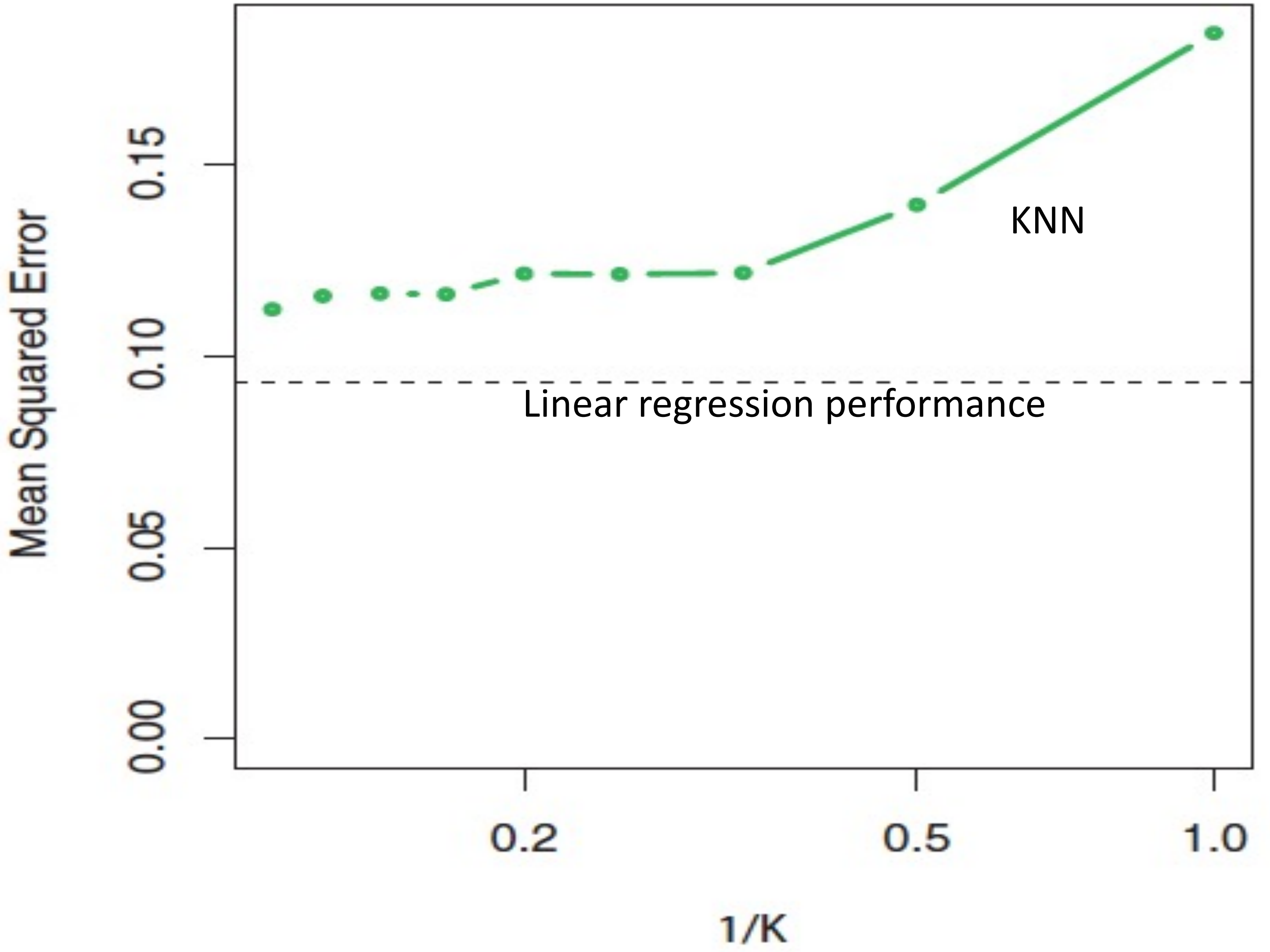
K=9



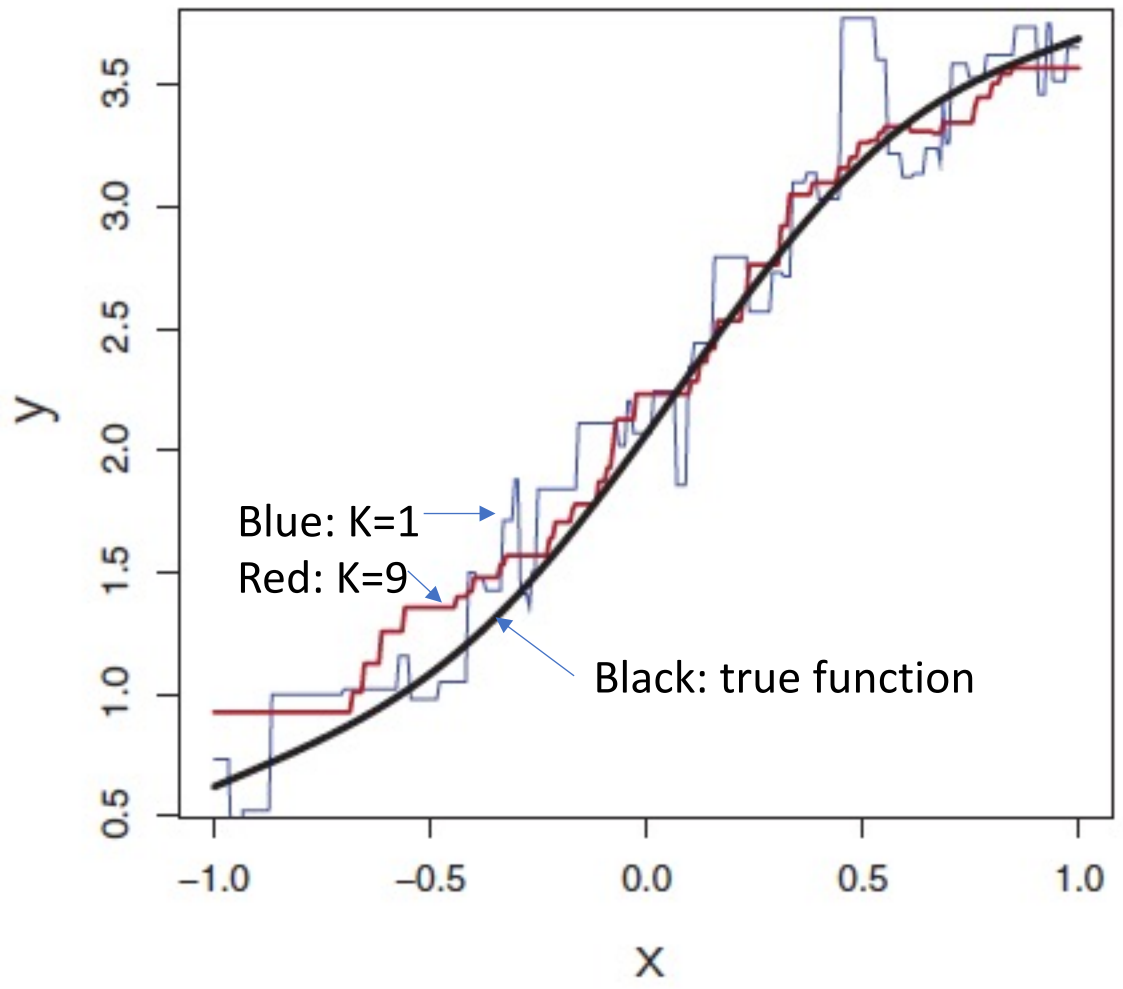
- The parametric approach will outperform the nonparametric approach if the parametric form that has been selected is close to the actual form

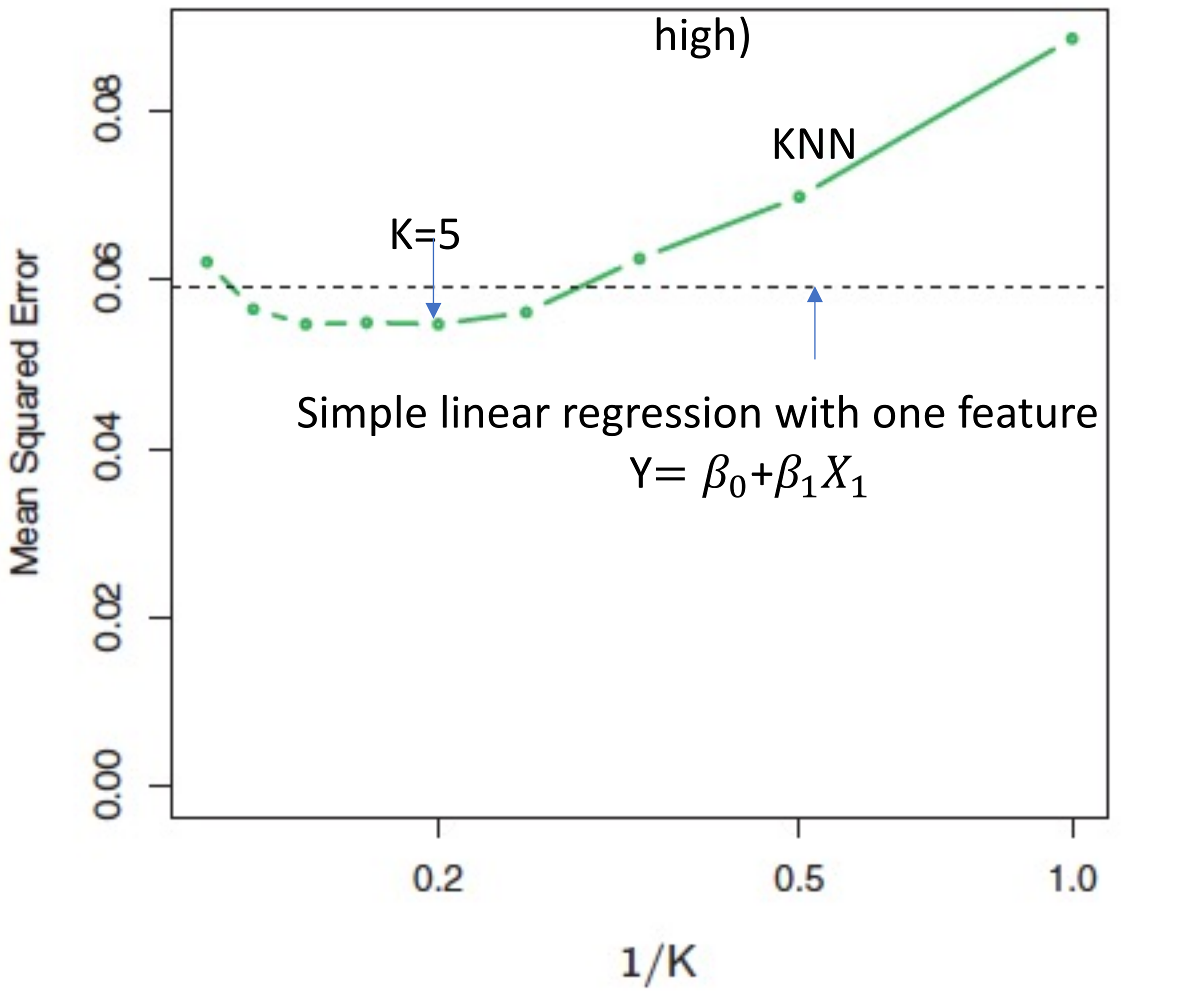


- Since the actual function is linear in this example, linear regression performs better than KNN

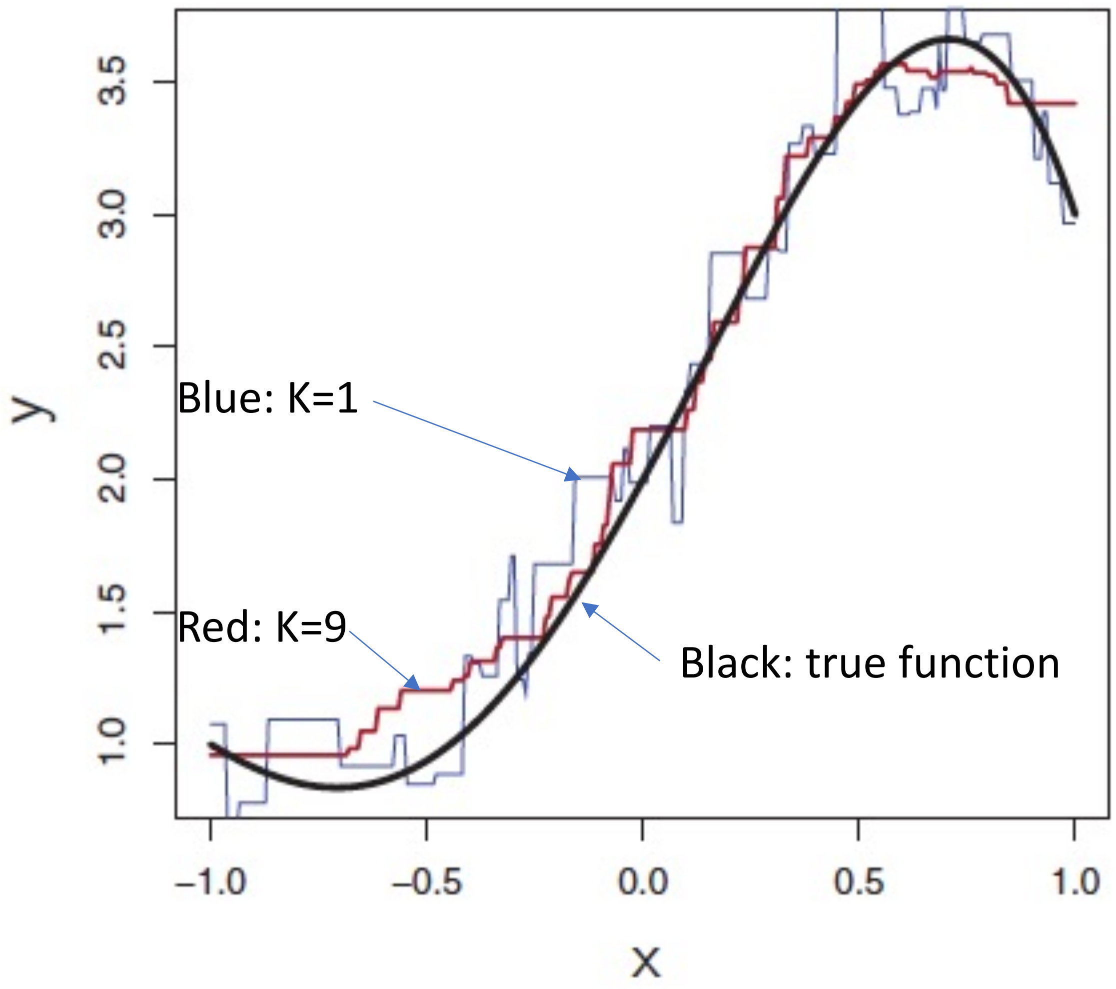


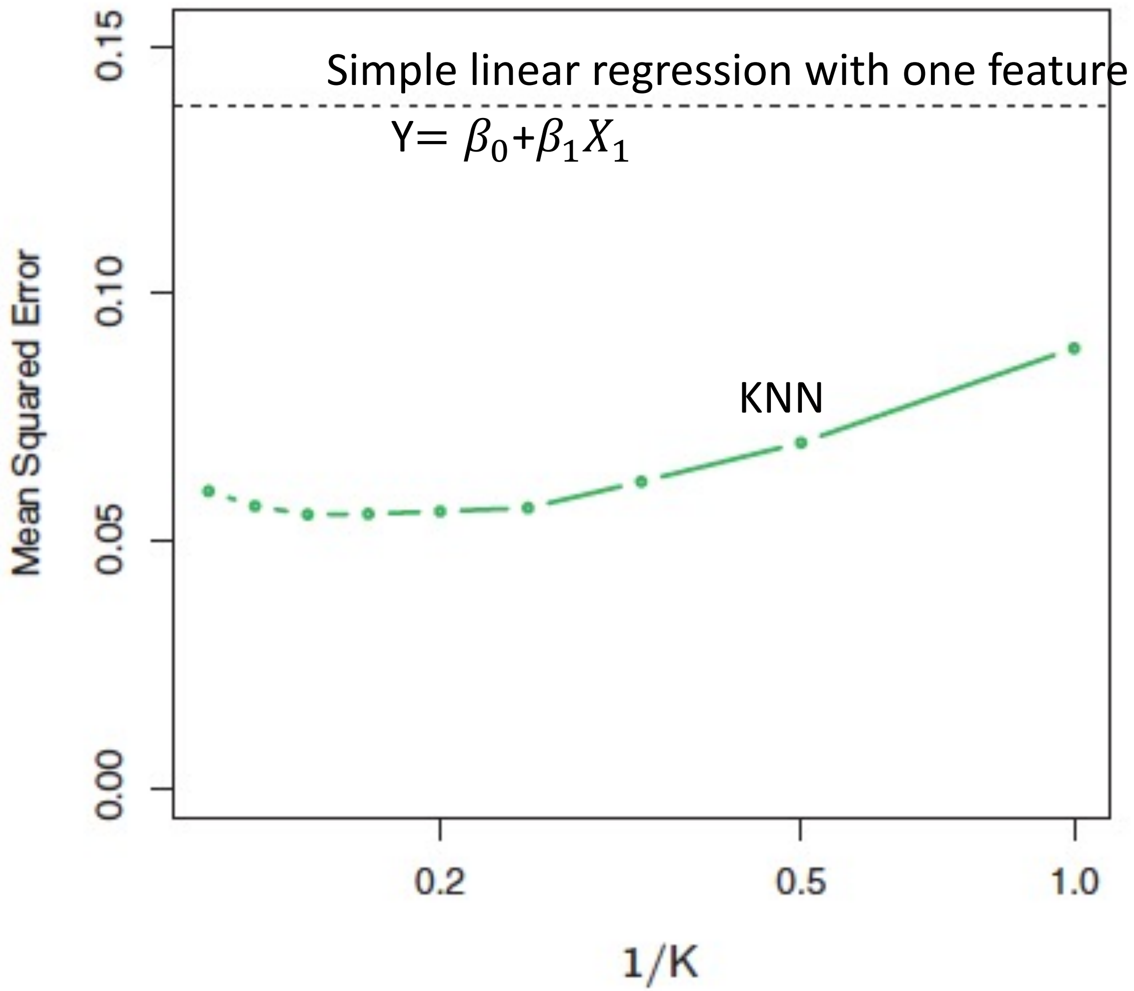
- Another example, where the actual function is not linear





Linear assumption of linear regression model ($Y=\beta_0+\beta_1 X_1$) 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, and we have 100 training observations each of p features
- As the number of features increases the KNN performance degrades (common problem)

Here we have **100 observations**

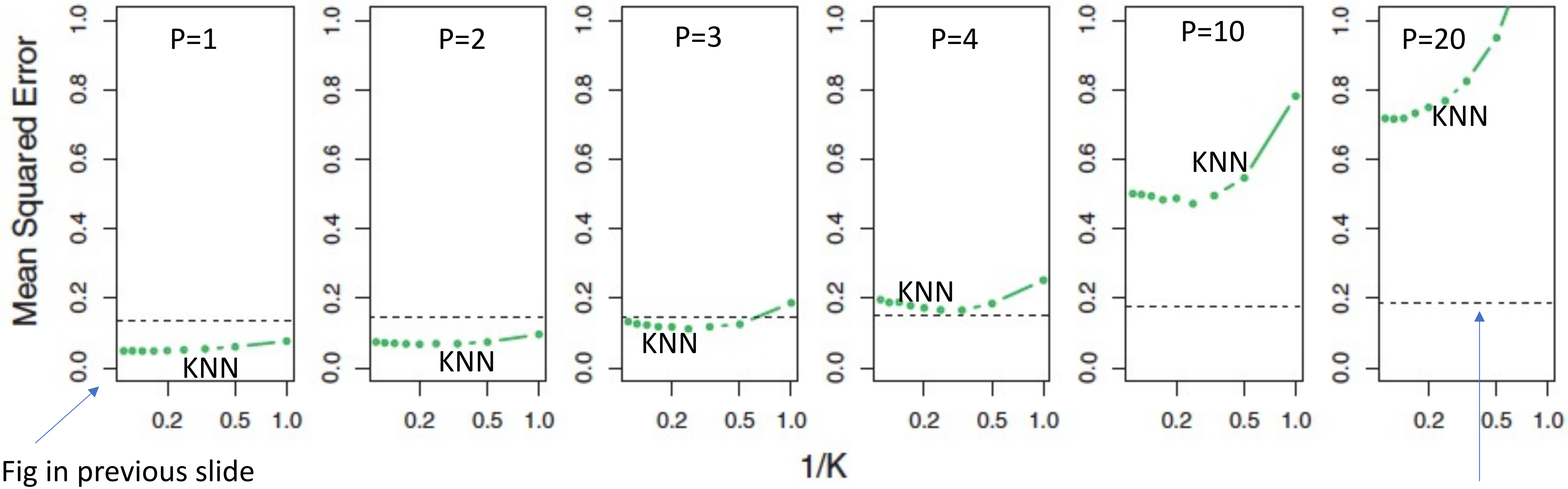


Fig in previous slide

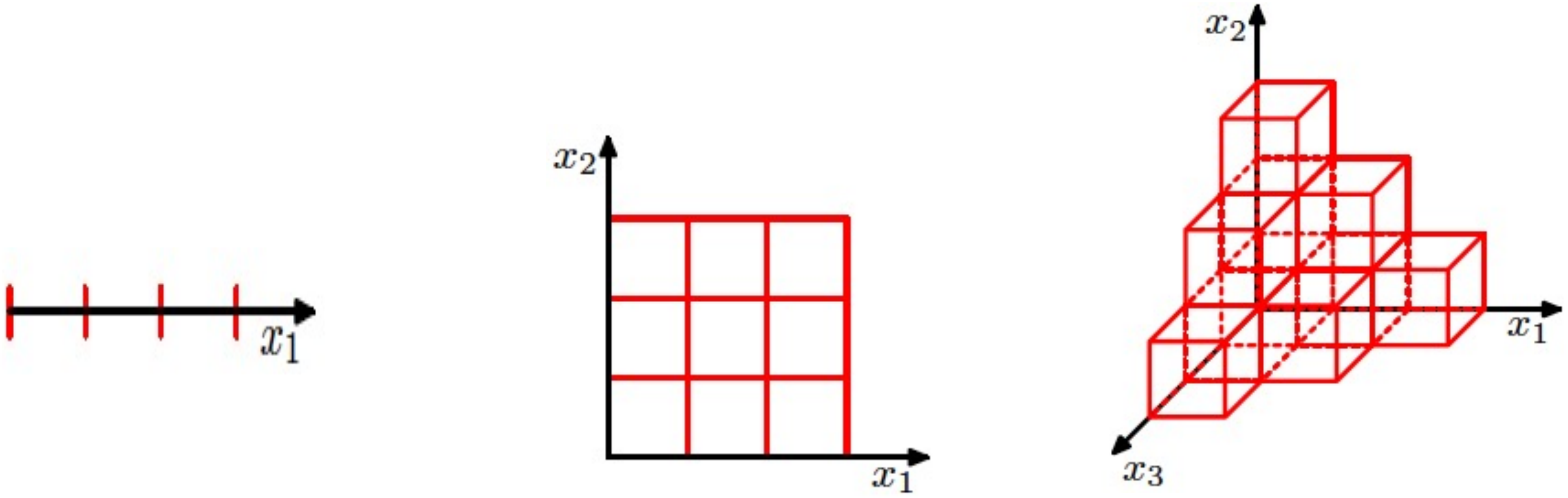
Horizontal black dashed line in all figures is linear regression (assume linear function with all predictors)

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 examples → leads to poor accuracy
  - In previous example:
    - One feature, 100 observations provide 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



Reference: Bishop, Pattern Recognition

Key takeaways

- Including qualitative features
- Relaxing the assumptions of linear models
- Comparing with KNN
- Curse of dimensionality