UNIVERSITY OF SAN FRANCISCO

Master of Science in Analytics

Linear Regression
KNN Regression

Machine Learning 1

Critical questions

- Strength of predictor / response relationship
  - Does feature predict response?
  - How strongly?
  - Which features (or feature sets) best predict response?
  - How accurate can we be in predicting response?
- Relationships
  - Is the relationship between feature and response linear?
  - Is there synergy between features?

Simple Linear Regression

- General form of function for Linear Regression:

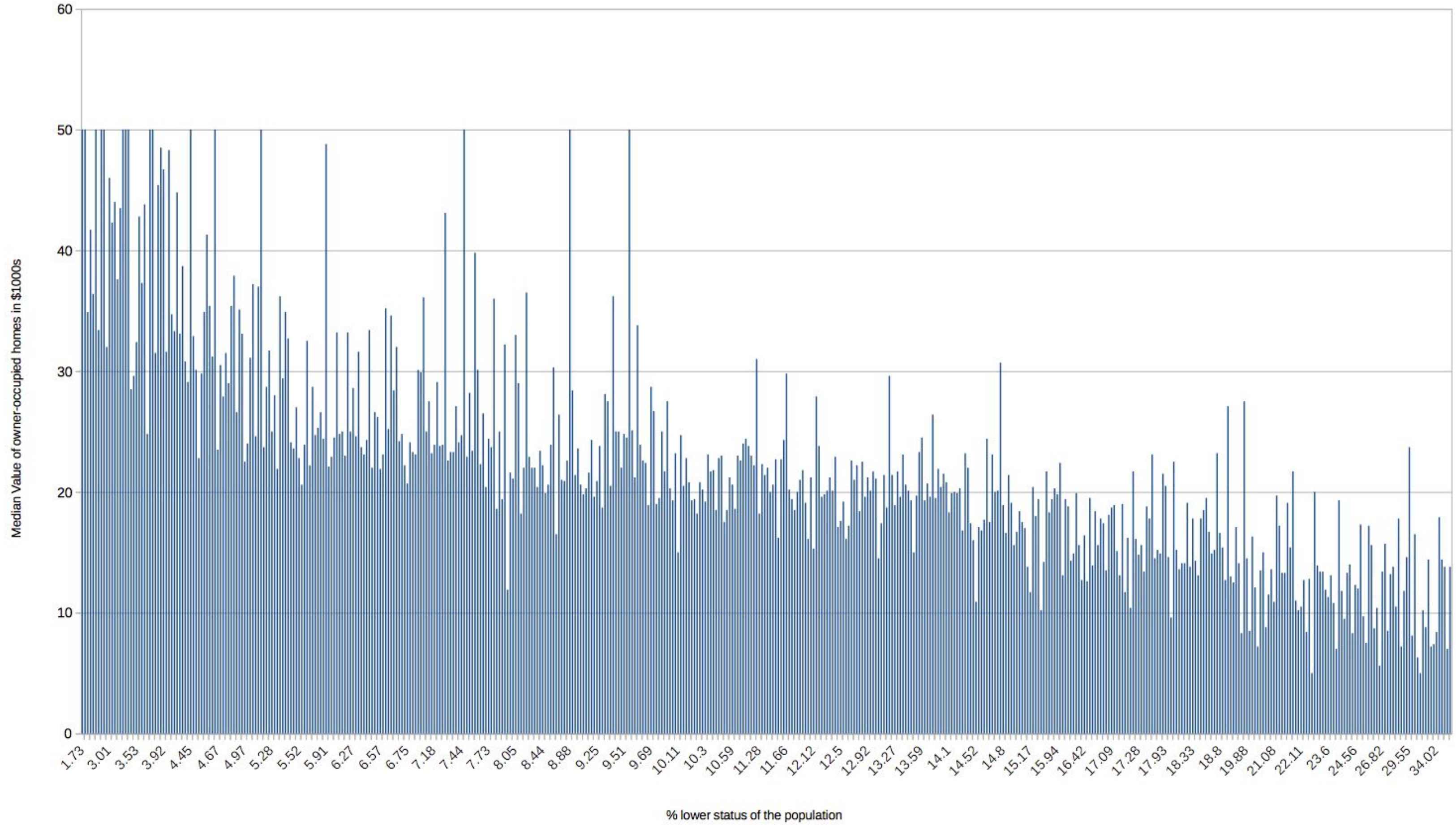
$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

- Linear Regression, simplified to 1 feature:

$$Y \approx \beta_0 + \beta_1 X_1$$

(Two columns of) Boston data

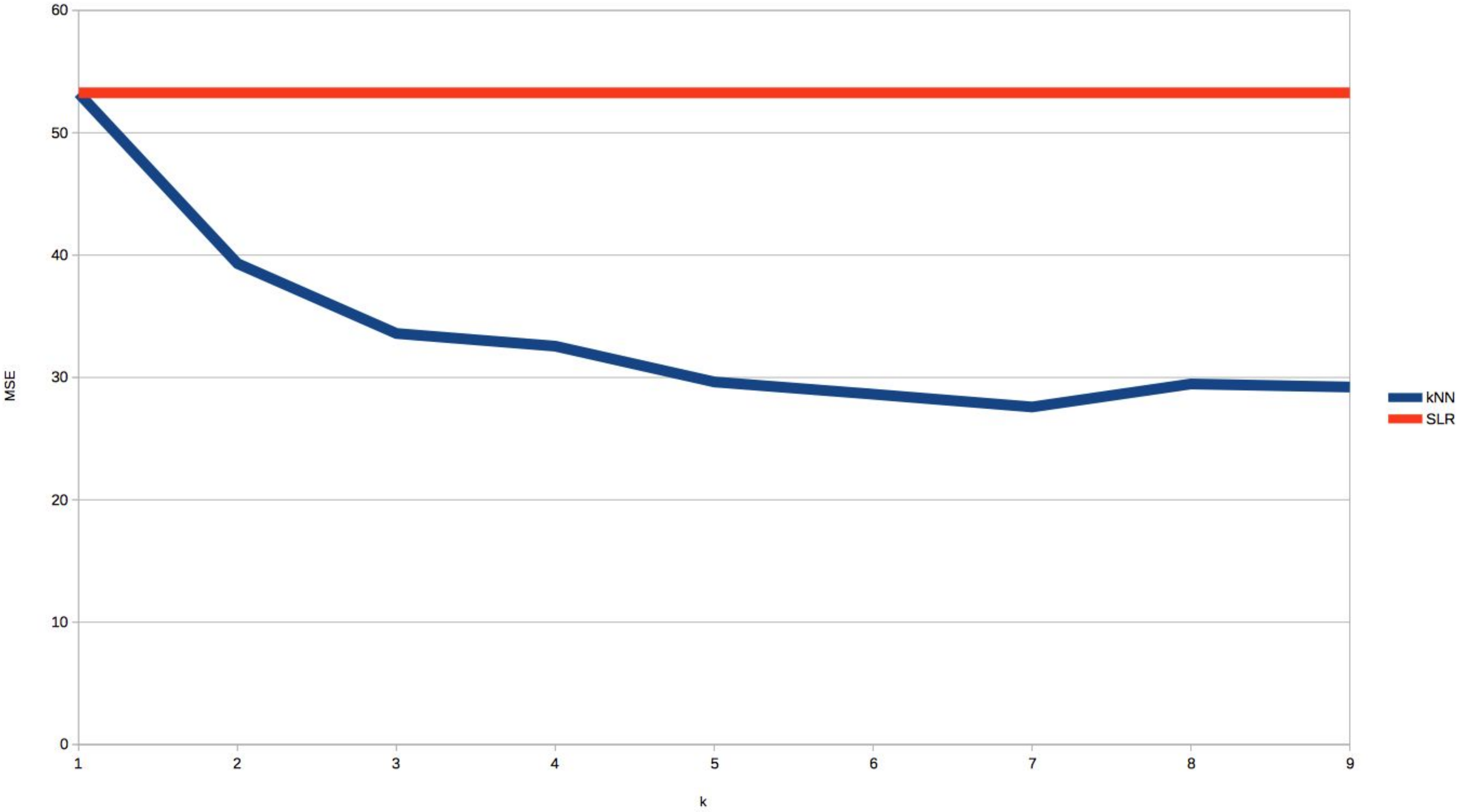
Full data on [Github](), at [UCI](), etc.



K Nearest Neighbours (KNN)

- Algorithm:
  - Determine “k”
  - Use some distance (Euclidean?) measure on features to determine k nearest training exemplars
  - Calculate response based on according to some function (mean?) of k nearest exemplars
- Observations:
  - Localised method, typically classified as a non-parametric method (i.e. no β)
  - Low bias; potential to keep variance low

SLR vs. KNN (Boston train/test)



SLR coefficients

- Method 1:
  - $r = \Sigma xy / (\Sigma x^2 \Sigma y^2)^{½}$
  - $\beta_1 = r\, \sigma_Y / \sigma_X$
  - $\beta_0 = \mu_X - \beta_1 \mu_Y$
- Method 2:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

Implementation in scikit-learn

1) Modify a hacked-up example
2) Import data
   a) If necessary, split data into train, test sets
3) Coerce data into:
   a) test_x, train_x = List-of-lists / numpy matrix: all features
   b) test_y, test_y = List / numpy vector: all targets

```
# 3a) SLR

from sklearn import linear_model

# SLR takes no parameters
algo = linear_model.LinearRegression()
algo.fit (train_x, train_y)
hypotheses = algo.predict (test_x)
```

```
# 3b) KNN

from sklearn.neighbors import KNeighborsRegressor

k = 5  # Can be changed to any integer > 0
algo = KNeighborsRegressor (n_neighbors=k)
algo.fit (train_x, train_y)
hypotheses = algo.predict (test_x)
```

4) Perform analysis on hypotheses

SLR coefficient accuracy

- Recall: there are two types of errors
  - Reducible errors: errors in coefficients
  - Irreducible errors: the ϵ term
- Estimate errors on μ, β_0 & β_1:
  - σ = standard deviation of each y_i of Y
  - Use standard error (SE), a/k/a Var:
    - μ:

$$\mathrm{Var}(\hat{\mu}) = \mathrm{SE}(\hat{\mu})^2 = \frac{\sigma^2}{n}$$

    - β_0 & β_1:

$$\mathrm{SE}(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right], \quad \mathrm{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Fit (or lack of fit) measures

- Background calculations
  - RSS = $(y_1 - \beta_0 - \beta_1 x_1)^2 + (y_2 - \beta_0 - \beta_1 x_2)^2 + \ldots + (y_n - \beta_0 - \beta_1 x_n)^2$
  - TSS = $\sum(y_i - \bar{y})^2$
- Measure 1: Residual Standard Error (RSE)

$$\mathrm{RSE} = \sqrt{\mathrm{RSS}/(n-2)}$$

  - Problem: RSE is expressed in terms of Y
  - Difficult to compare two models' RSE
- Measure 2: R^2 statistic

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

  - Measure of linear relationship between X and Y
  - Identical to the squared correlation

Confidence intervals

- Definition
  - Intuitive: x% chance that an interval will contain the true value of y
  - Full definition: A 95% confidence interval is a range of values such that with 95% probability, the range will contain the true unknown value of the parameter
- 95% confidence intervals for SLR:
  - β_0:

    $$\hat{\beta}_0 \pm 2 \cdot \mathrm{SE}(\hat{\beta}_0)$$

  - β_1:

    $$\hat{\beta}_1 \pm 2 \cdot \mathrm{SE}(\hat{\beta}_1)$$

Feature-response relationship

- SE can be used to test whether feature (x_i) really predicts response (y_i)
  - Designed as hypothesis test
  - H_0 (null hypothesis): no relation ~ $\beta_1 = 0$
  - H_a (alternative hypothesis): some relation ~ $\beta_1 \neq 0$
- The t-statistic or p-value are useful
  - Intuitively: looking for small standard error:
  - t-statistic:

$$t = \frac{\hat{\beta}_1 - 0}{\mathrm{SE}(\hat{\beta}_1)}$$

  - Low p-value = (low) probability that t-statistic will generate chance relationship

Multiple linear regression

- Reminder, formula is of the form:

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

- Basic calculations same as SLR
  - Individual feature's SE, t-statistic and p-value still apply as with SLR
  - Not trivial to derive coefficients; use package (scikit-learn) to estimate
- Feature set vs. response relationship
  - H_0 (null hypothesis): no relation ~ $\beta_1 = \beta_2 = \ldots = \beta_p = 0$
  - H_a (alternative hypothesis): some feature ~ $\beta_j \neq 0$
  - Can test hypothesis using F-statistic:

$$F = \frac{(\mathrm{TSS} - \mathrm{RSS})/p}{\mathrm{RSS}/(n-p-1)}$$

MLR fit (or lack of fit)

- *R^2* statistic
  - Problem: always increases with additional features
  - Look for magnitude of increase
- General RSE:

$$\mathrm{RSE} = \sqrt{\frac{1}{n-p-1}\mathrm{RSS}}$$

MLR feature selection

- Feature selection is not trivial:
  - An individual features may be indistinguishable from others
  - Individual features may be useful only in combination
  - Therefore, feature selection is ~ SAT = $O(2^p)$
- Trivial Approximations
  - Forward selection:

```
assume H_0
while (arbitrary stopping criteria not met):
    consider p SLR models not already considered
    add to set: model with lowest RSS
```

  - Backward selection
  - Mixed selection

What if...

- Variables are weird?
  - Boolean features (eg. female vs. male?): encode as {T,F} {0,1} or {1,-1}
  - Qualitative features (hair colour): encode as 1-hot (multiple booleans) — eg. {auburn: T/F}, {black: T/F}, {brown: T/F}, {grey: T/F}, ...
- Feature is not (directly) additive?
  - Example: feature to response only when other feature is present?
  - Use p-value to determine *main effect*
- Feature is not linear?
  - Discover through plot (eg. residual plot) of data
  - Try something other than SLR/MLR
- Error terms have non-constant variance?
  - Use weighted least squares

Good data, bad data, ugly data...

- Outliers
  - Data far outside of predicted value
  - Often change RSE, R^2
  - Consider removing if more than 3 standard deviations from mean
- High Leverage Points
  - Data far outside feature space
  - May dramatically change (regression) model
  - Test with leverage statistic:

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2}$$

Collinearity

- Definition & problem
  - Two features which are highly correlated (or overlapping)
  - Hard to determine what effect each has on the response
  - Technically: difficult to estimate coefficients for features
- Detection
  - Plot features
  - Use a correlation matrix
  - Use variance inflation factor (VIF)

$$\mathrm{VIF}(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

- Fix (pick one):
  - Drop all but one of the problematic features
  - Combine collinear features into a single feature