Machine Learning I

Lecture 3: Linear Regression

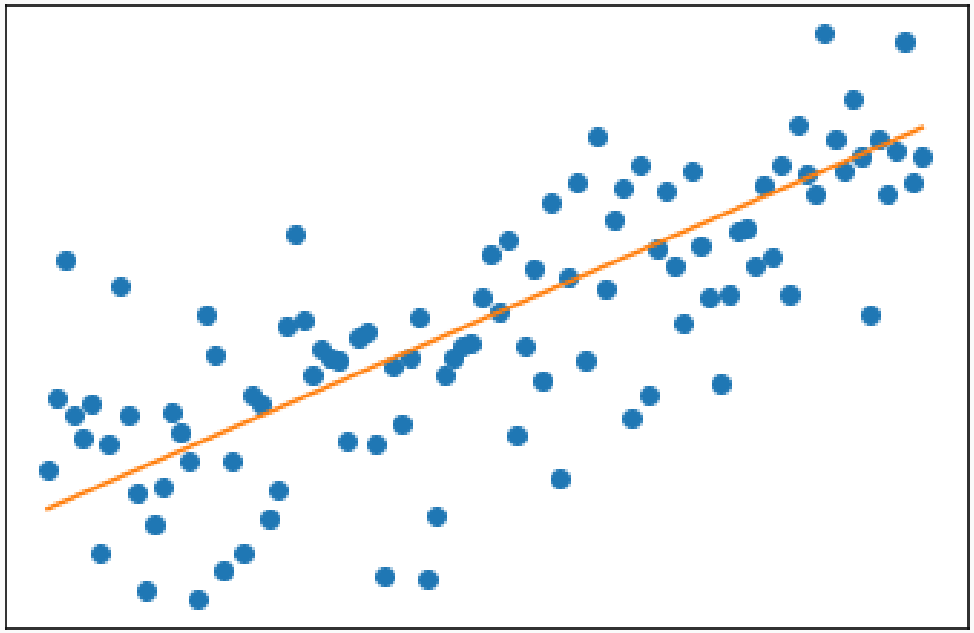
Nathaniel Bade

Northeastern University Department of Mathematics

Table of contents

Least Squares for Linear Models

Linear Models: Definition

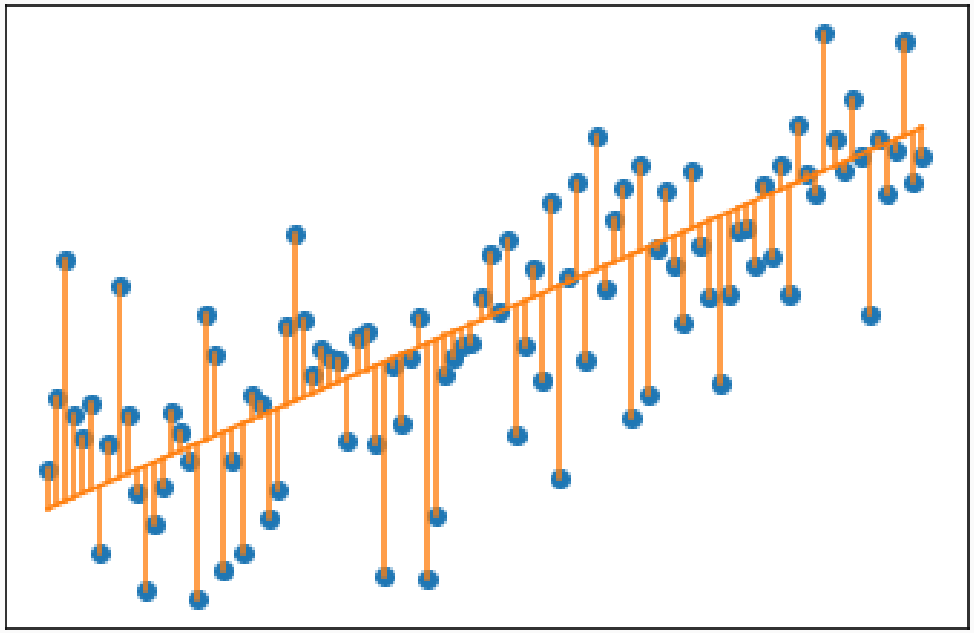


A **linear regression model** is a model of data fitting that assume that the regression function $E(Y|X)$ is linear in the inputs $X_1, \ldots, X_p$,

$$Y = \beta_0 + X_1\beta_1 + \ldots + X_p\beta_p + \epsilon$$

and uses the residual sum of squares $RSS(\beta)$ as a loss function. Here ϵ is a random variable with $E[\epsilon] = 0$.

Linear Models: Definition

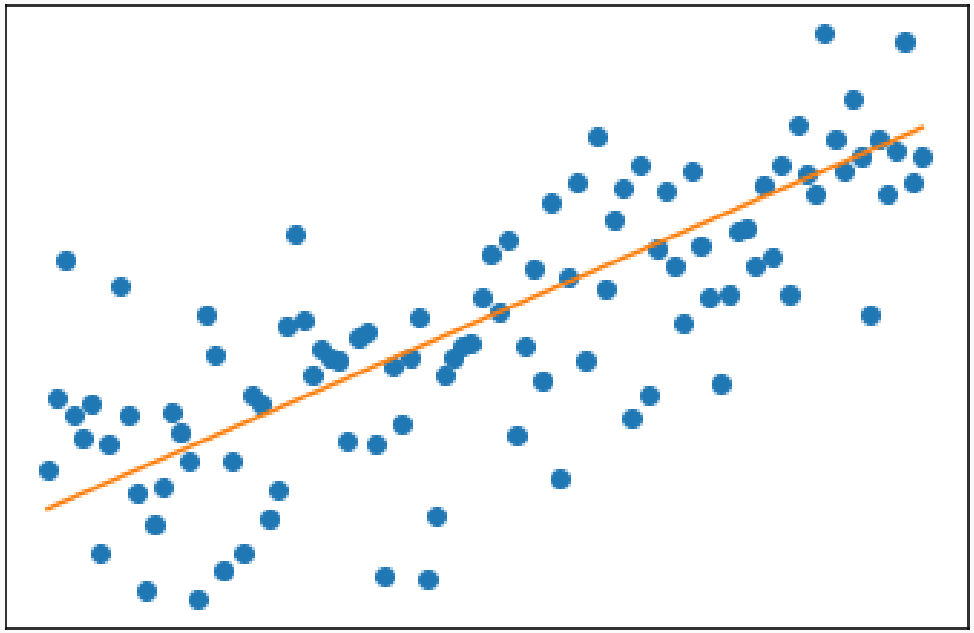


A **linear regression model** is a model of data fitting that assume that the regression function $E(Y|X)$ is linear in the inputs $X_1, \ldots, X_p$,

$$Y = \beta_0 + X_1\beta_1 + \ldots + X_p\beta_p + \epsilon$$

and uses the residual sum of squares $RSS(\beta)$ as a loss function. Here ϵ is a random variable with $E[\epsilon] = 0$.

Linear Models: Definition

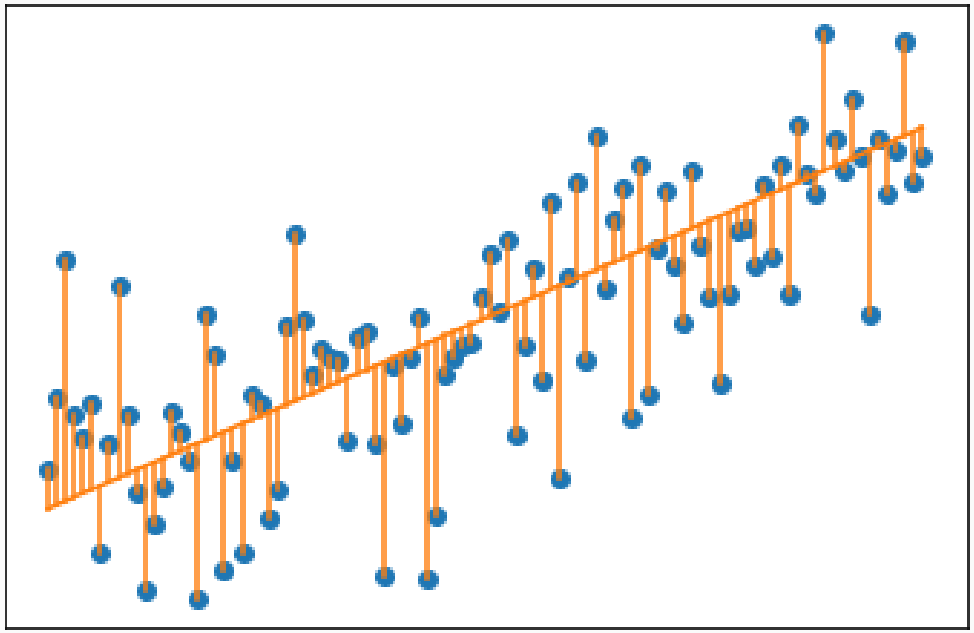


In vector form, we add a constant term $X = (1, X_1, \ldots, X_p)$ and let β be a $p + 1$ vector including β_0:

$$Y = \beta_0 + X_1\beta_1 + \ldots + X_p\beta_p\epsilon = X^T\beta + \epsilon\,,$$

where again we uses the residual sum of squares $RSS(\beta)$ as a loss function.

Linear Models: ERM Solution

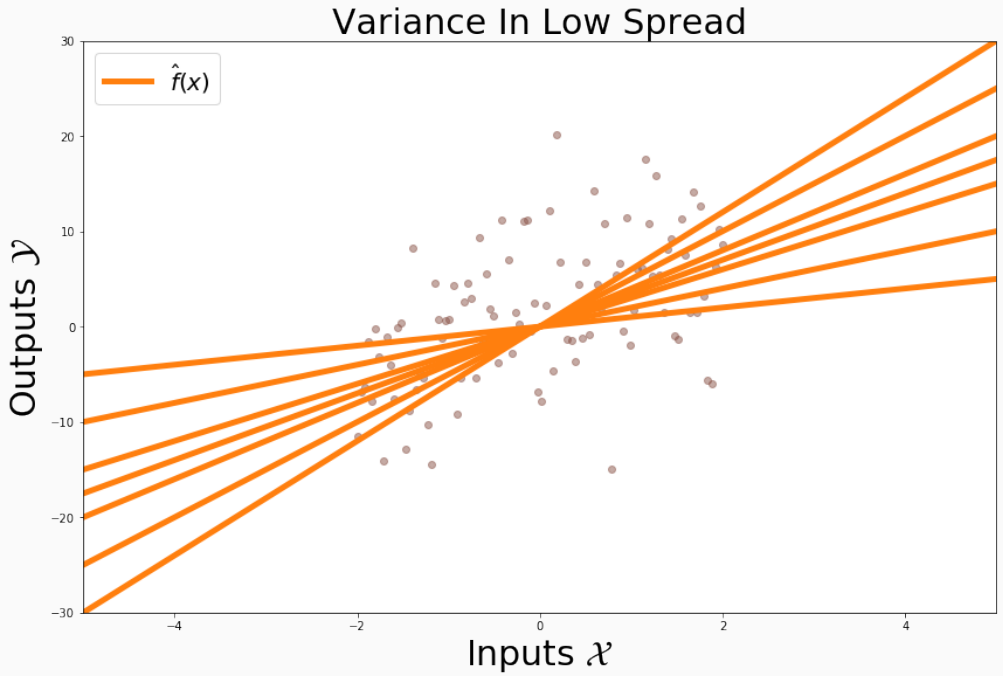


Given the $N \times (p+1)$ matrix $\mathbf{X}$ of N input data points x_i and a vector $\mathbf{y}$ of the N corresponding labels y_i, we saw that

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - x_i^T)^2 = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$$

is minimized by $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

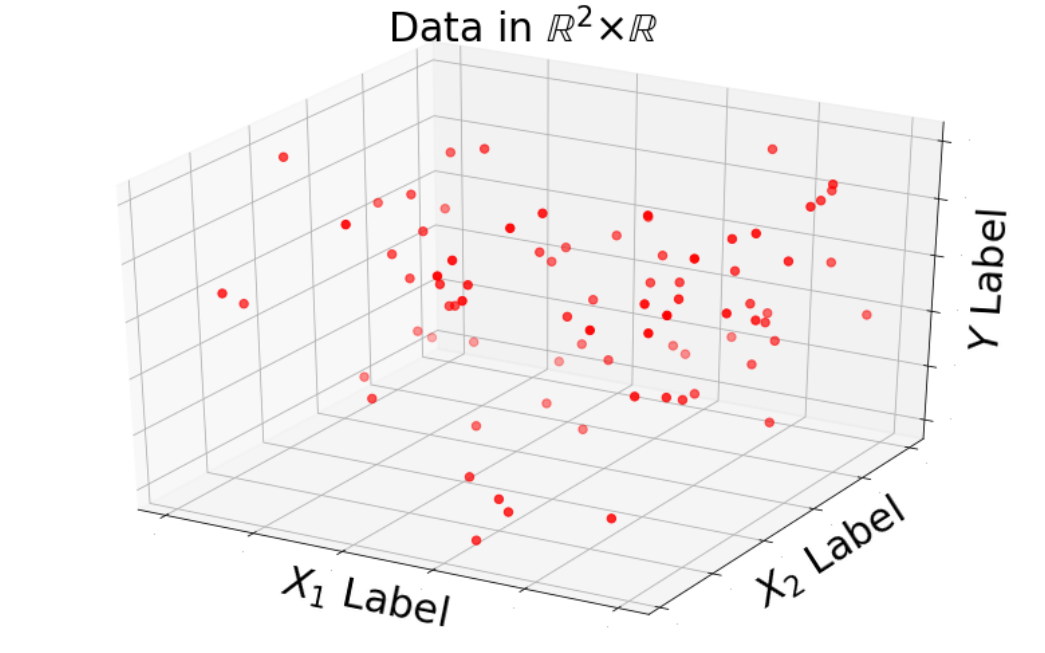
Linear Models: Goodness of Fit



We now turn our attention to quantifying the goodness of our fit. We would like to answer two questions:

- How much do we expect $\hat{f}$ to change if we draw different data from the same distribution?
- Which parameters correspond to statistically significant features?

Linear Models: Pointwise Variance



To the second question: when fitting multivariate (multi-feature) linear models, one of our main goals is to understand which parameters most directly control the labeling.

For a multilinear model, the coefficients β give some measure of correlation, but we still need to investigate what exactly these variables mean, what constitute expected and unexpected values.

Example Data

In the lecture we're going to use as an example a data set of undergraduate confidence in admission to graduate school. The data has 8 input features and 1 output feature

Data Sample:

Serial No.	GRE Score	TOEFL Score	University Rating
1	337	118	4
2	324	107	4
3	316	104	3

SOP	LOR	CGPA	Research	Chance of Admit
4.5	4.5	9.65	1	0.92
4.0	4.5	8.87	1	0.76
3.0	3.5	8.00	1	0.72

Example Data

The features of the data are

Serial No.

GRE Score: Out of 340.

TOEFL Score: Out of 120.

University Ranking: Out of 5.

SOP: Statement of purpose strength out of 5.

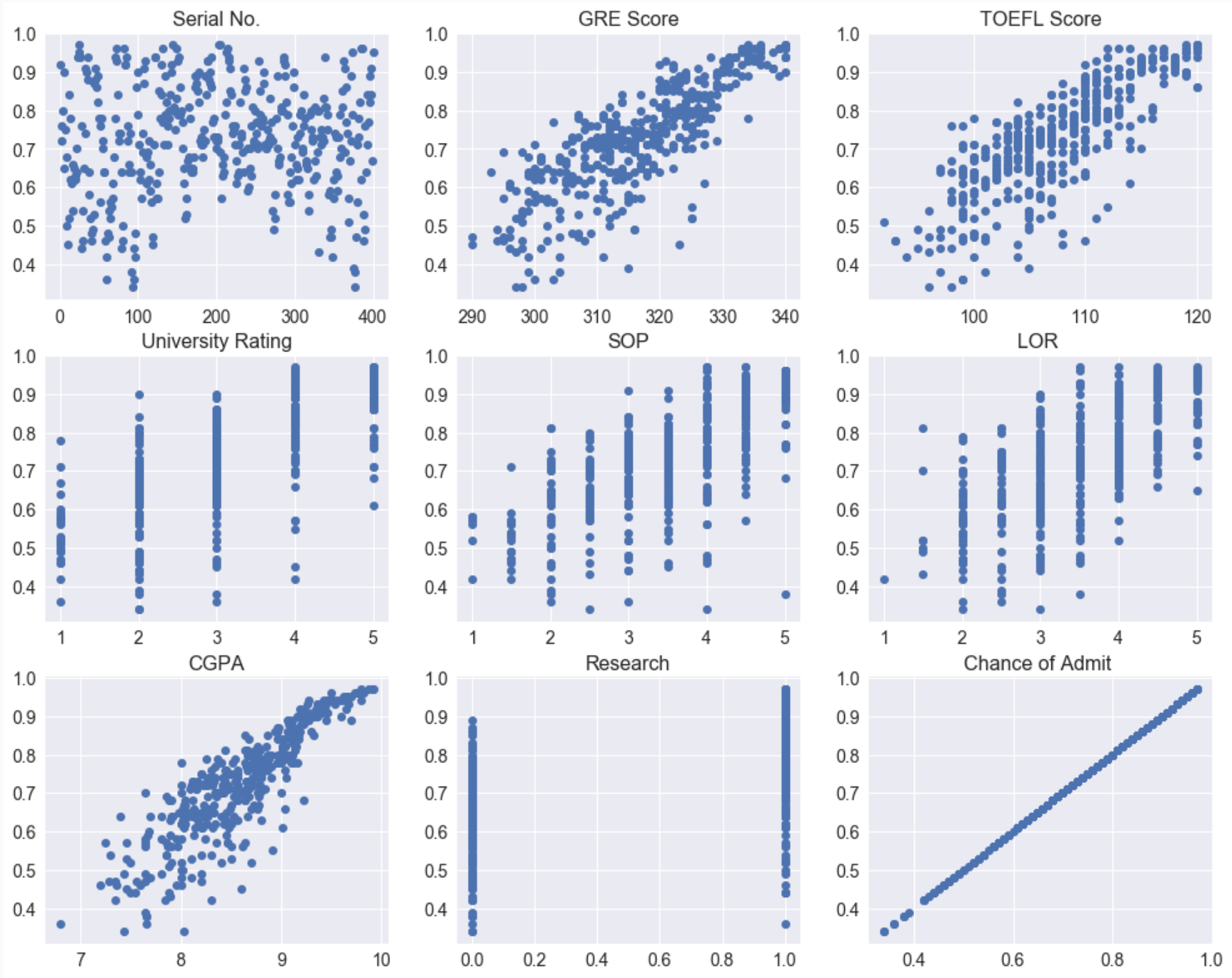
LOR: Letter of recommendation strength out of 5.

CGPA: Undergraduate GPA out of 10.

Research: Research experience 0 or 1.

Chance of Admit: Chance of admission.

Note: This data claims to be self reported, it is not authoritative.


Serial No.
GRE Score
TOEFL Score
University Rating
SOP
LOR
CGPA
Research
Chance of Admit

Linear Models: Parameter Variance

The fit parameter $\hat{\beta}_i$ in the linear model

$$\hat{Y} = X^T \hat{\beta}$$

describes the expected correlation between the i'th feature and the chance of admission. Before we discuss the fit of the model, we will discuss the quantitative meaning of the parameters and understand what the data informs us an expected for the parameters $\hat{\beta}_i$.

In particular, we want to understand quantitatively when a parameter value $\hat{\beta}_i$ is statistically significant. Statistically, this mean we want expressions for the p-values and confidence intervals for the $\hat{\beta}_i$ that can be computed from the data set.

Mathematically, this boils down to understanding variance of $\hat{\beta}_i$ as a function of the input data $\mathbf{X}$ and $\mathbf{y}$.

Variance of Parameters

Review: Confidence Intervals

First, lets quickly recall the confidence interval paradigm in statistics. Suppose we want to estimate the mean μ of a distribution from a data set x_i, $i = 1, \ldots, N$ drawn from the distribution. Our **point estimate** for the mean is

$$\hat{m} = \frac{1}{N} \sum_{i=1}^{N} x_i \, .$$

One way to quantify the accuracy of this estimate is to compute a **confidence interval**, or **interval estimate** for the statistic $\hat{m}$.

To compute a confidence interval we have to make some assumptions about the distribution of the statistic over the space of different draws from background distribution.

Review: Confidence Intervals

For computations of the mean, we often invoke the **Central Limit Theorem** (CLT) to assume that the distribution of test statistics is drawn from a Normal Distribution:

$$\mathcal{N}(\sigma^2, \mu) = \mathcal{N}(X|\sigma^2, \mu) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)^2} .$$

The CLT roughly states that for X_j, $j = 1, \ldots, p$ drawn independently from any distributions with expected value μ and variance σ^2, then

$$\hat{m}_p = \frac{X_1 + \ldots + X_p}{p} , \Rightarrow \frac{\hat{m}_p - \mu}{\sigma/\sqrt{p}} \xrightarrow{p} \mathcal{N}(1, 0) ,$$

that is that m_p is approximately normally distributed over samples from the dataset when p is large.

Review: Confidence Intervals

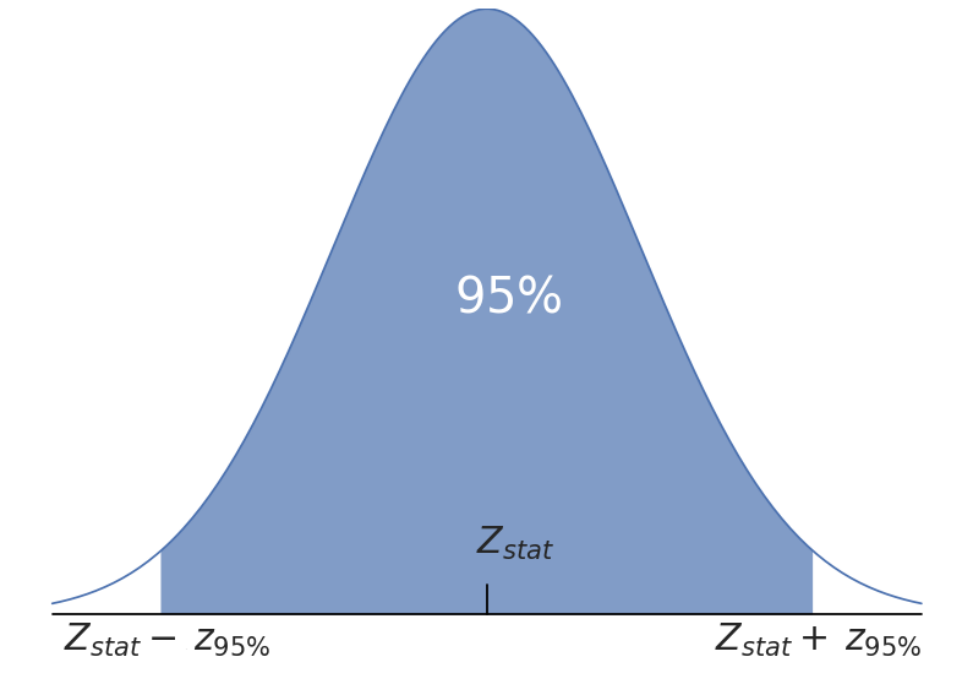
Given a specific test statistic $\hat{m}$, the $C\%$ confidence interval I_C is the interval in which $C\%$ of the computations of $\hat{m}$ from random draws of the data will lie. This interval should contain $\hat{m}$ with a high probability, no matter what the value of μ is.

Formally, for any statistic $\theta = f(x_1, \ldots, x_N)$ the $1 - \alpha$ confidence interval I is a set such that

$$P_\theta(\theta \in I) \geq 1 - \alpha \,.$$

That is I traps θ with probability $1 - \alpha$.

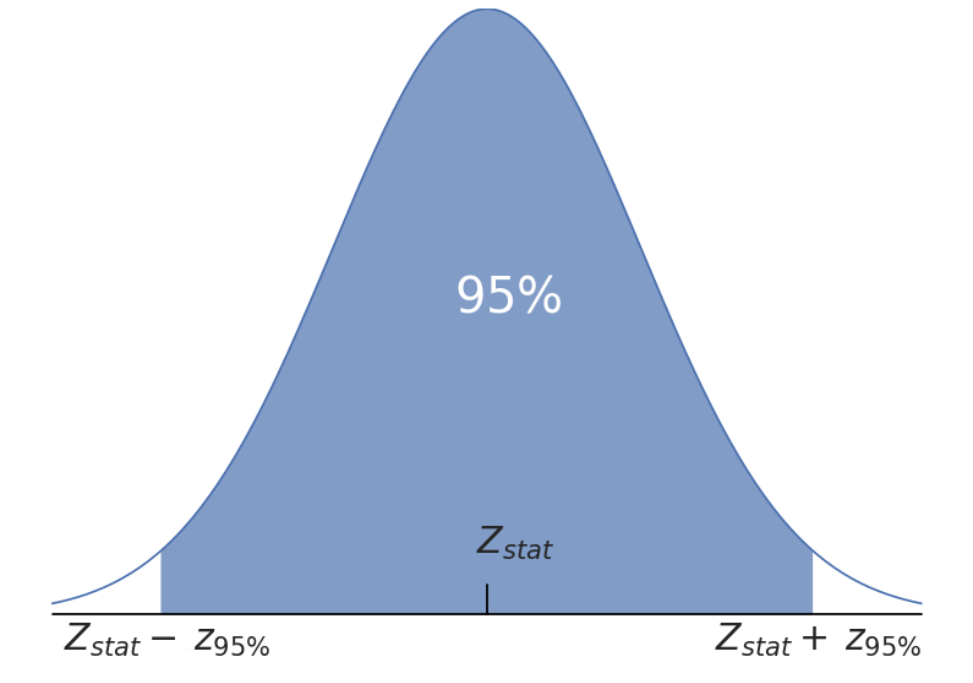
Review: Confidence Intervals



By the CLT, $Z = \frac{\hat{m}-\mu}{\sigma/\sqrt{p}}$ is distributed by the **standard normal distribution** $Z \sim \mathcal{N}(1, 0)$. It is standard to use Z to parameterize this distribution. Then, if 95% of the distribution Z is captured in the interval $[-z_{95}, z_{95}]$, than we expect that 95% of the sampled $\hat{m}$ will fall in the interval

$$I_{95} = \left[\hat{m} - z_{95}\frac{\sigma}{\sqrt{p}}, \hat{m} + z_{95}\frac{\sigma}{\sqrt{p}}\right] .$$

Review: Confidence Intervals



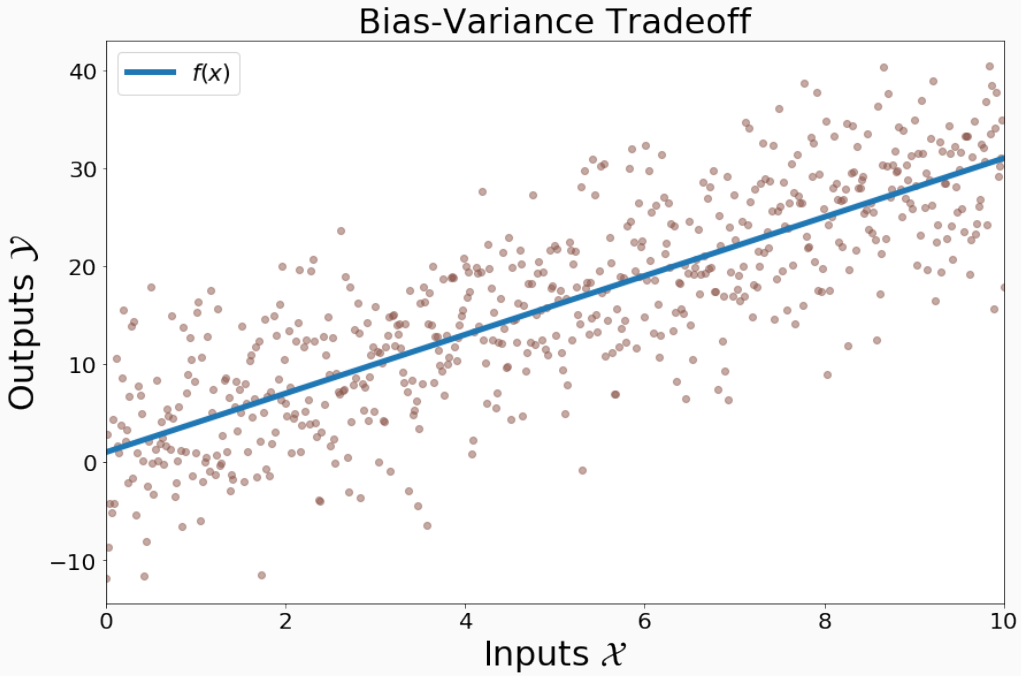
In summary, the process for constructing a confidence interval is

(1) Construct a test statistic S

(2) Determine the distribution $\mathcal{D}$ for S over different samples (possibly up to assumptions on data distribution).

(3) Find an interval I_C that captures C% of the draws from $\mathcal{D}$. Generally this will be centered S, but it need not be.

Linear Models: Definition

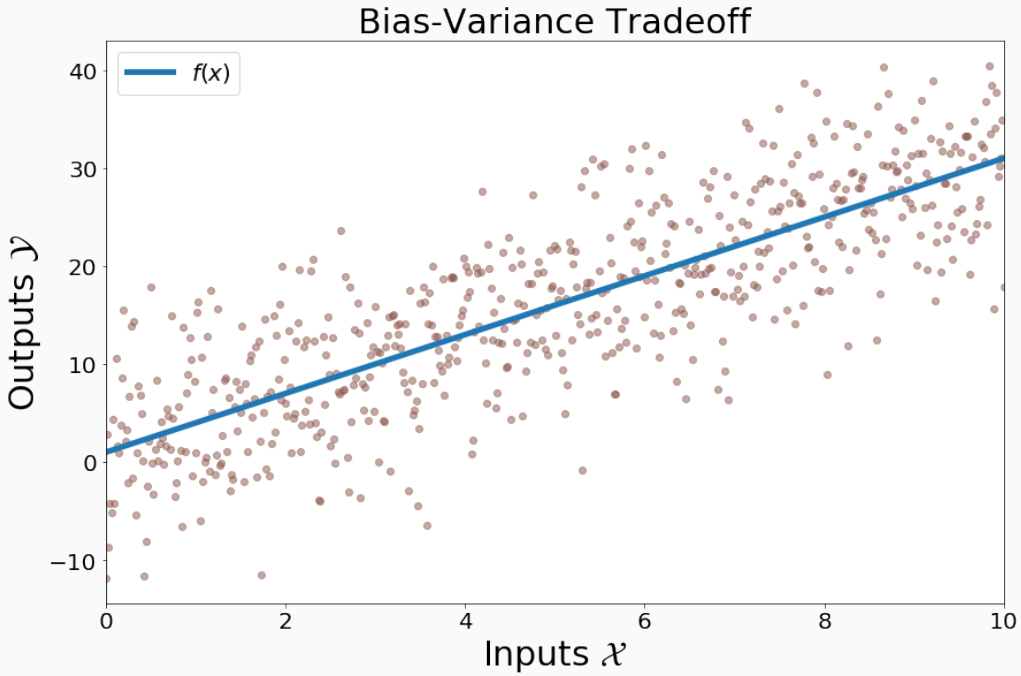


Given our linear model

$$Y = \beta_0 + X_1\beta_1 + \ldots + X_p\beta_p + \epsilon = X^T\beta + \epsilon\,,$$

fit by minimizing RSS, we want to compute confidence intervals for each of the parameters β_j.

Linear Classifier with Noise

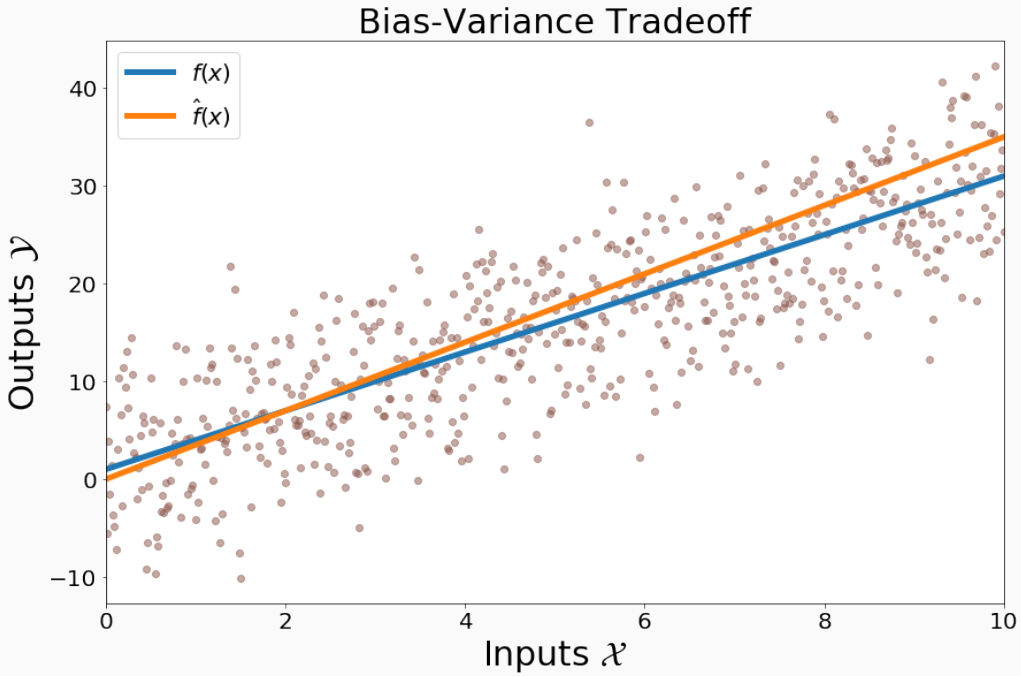


Suppose that we know that the relationship between X and Y is almost linear

$$Y = f_*(X) = X^T \beta_* + \epsilon\,,$$

where β_* denotes the actual weights and ϵ is a random variable drawn from a normal distribution with $E[\epsilon] = 0$ and variance σ^2.

Linear Classifier with Noise



The variance in the fit parameters $\hat{\beta}$ over draws of N data points $\mathbf{X}$ is

$$\mathbf{Var}(\hat{\beta}) = \mathrm{Var}_\epsilon\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\right).$$

We would like to derive an explicit expression for $\mathbf{Var}(\hat{\beta})$ in terms of $\mathbf{X}$, $\mathbf{y}$ and σ^2.

Linear Classifier with Noise

Recall that for a p-vector X of random variables, $\mathrm{Var}(X)$ is a $p \times p$ matrix with jk'th entry given by the covariance

$$\mathrm{Var}(X)_{jk} = \mathrm{Cov}(X_j, X_k) = E[(X_j - E[X_j])(X_k - E[X_k])\,.$$

For a set of datapoint $x_i \in \mathbb{R}^p$ collected into the matrix $\mathbf{X}$ let the sample mean of the j'th feature be $\hat{m}_j = E[\mathbf{x}_j]$. One can compute the sample covariance from datapoints as

$$\mathrm{Cov}(\mathbf{x}_j, \mathbf{x}_k) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{X}_{ij} - \hat{m}_j)(\mathbf{X}_{ik} - \hat{m}_k)\,.$$

All together, $\mathrm{Var}(X) = E[(X - E[X])(X - E[X])^T]$.

Variance of Linear Transform

For vector ϵ of i.i.d. random variables each with variance σ_ϵ^2, and a matrix A,

$$\text{Var}(A\epsilon) = \sigma_\epsilon^2 AA^T .$$

This follow from the definition of the (co)variance matrix $\text{Var}(\epsilon) = E[(\epsilon - E[\epsilon])(\epsilon - E[\epsilon])^T]$:

$$\begin{aligned}
\text{Var}(A\epsilon) &= E[(A\epsilon - E[A\epsilon])(A\epsilon - E[A\epsilon])^T] && \text{Definition of Var.} \\
&= A\,E[(\epsilon - E[\epsilon])(\epsilon - E[\epsilon])^T]\,A^T && A \text{ constant} \\
&= A\,\text{Var}(\epsilon)\,A^T && \text{Definition of Var.} \\
&= A(\sigma_\epsilon^2 I)A^T && \epsilon \text{ i.i.d.} \\
&= \sigma_\epsilon^2 AA^T
\end{aligned}$$

The second the last line come from the fact that i.i.d. variables are uncorrelated, and so all off diagonal entries are 0.

Linear Models: Parameter Variance

The formula

$$\mathrm{Var}(A\epsilon) = \sigma_\epsilon^2 A A^T \,. \tag{1}$$

allow us to make a direct derivation of the parameter variance. Assume that the input points are fixed and the variance comes only from the labels. Then

$$\begin{aligned}
\mathrm{Var}_\epsilon(\hat{\beta}) &= \mathrm{Var}_\epsilon\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\right) && \text{Definition of } \hat{\beta}, \\
&= \mathrm{Var}_\epsilon\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\beta_* + \epsilon)\right) && \text{Definition of } \mathbf{y}, \\
&= \mathrm{Var}_\epsilon\left(\beta_* + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\epsilon\right) && \text{Distribution}, \\
&= \mathrm{Var}_\epsilon\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\epsilon\right) && \mathrm{Var}(\text{const}) = 0, \\
&= \sigma_\epsilon^2 \left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\right)\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\right)^T && (1) \text{ Above}, \\
&= \sigma_\epsilon^2 (\mathbf{X}^T\mathbf{X})^{-1} && \text{Multiplication}.
\end{aligned}$$

Linear Models: Parameter Variance

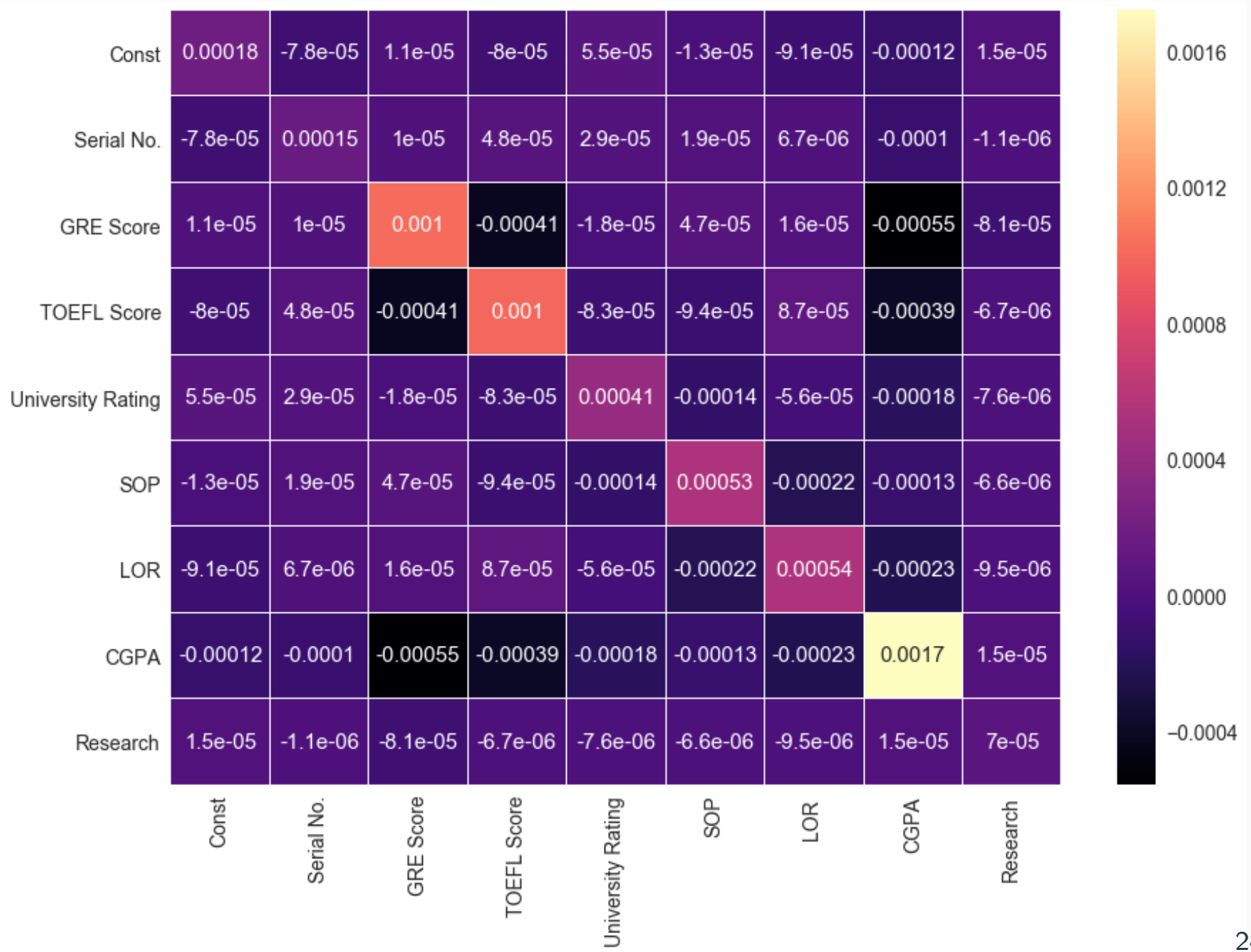
The variation of the parameters is then a $p+1 \times p+1$ matrix

$$\text{Var}(\hat{\beta}) = \sigma_\epsilon^2 \left(\mathbf{X}^T \mathbf{X}\right)^{-1},$$

with each term scaled by the variance of the noise parameter (recall that $\mathbf{X}$ contains a column of 1's).

Note, that unlike for the variance matrix of the i.i.d. variables ϵ in the computation for $\text{Var}(A\epsilon)$, we expect there to be correlation between the β_i's, and there will almost certainly be non-diagonal terms in the matrix.

For example, the variance in the parameters for the admission dataset is



Confidence Intervals for Coefficients

Estimating Total Variance

In practice, the variance σ_ϵ^2 of the random variable isn't given to us but must be estimated from the data. The variance is typically estimated by

$$\hat{\sigma}^2 = \frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{RSS(\hat{\beta})}{N-p-1}\,.$$

The factor of $N-p-1$ in the denominator makes $\hat{\sigma}$ into an **unbiased estimator**, in the sense that

$$E[\hat{\sigma}^2] = \sigma^2\,.$$

Linear Models: Parameter Variance

The intuition here is roughly that just as the sample variance for one parameter is corrected by

$$\bar{s} = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2\,,$$

the sample variance for our $p+1$ parameters will be corrected by a $p+1$ shift:

$$\hat{\sigma}^2 = \frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N-p-1}\sum_{i=1}^{N}(r_i - E[r_i])^2\,,$$

where $r_i = y_i - \hat{y}_i$ is leftover error in the fit at each datapoint, and $E[r_i]$ is expected to vanish for RSS.

Linear Models: Parameter Variance

Since

$$\mathrm{Var}_\epsilon(\hat{\beta}) = \sigma_\epsilon^2 \left(\mathbf{X}^T\mathbf{X}\right)^{-1},$$

(and linear regression is an unbiased predictor, $E_\mathcal{T}[\hat{\beta}] = \beta_*$), if we assume that ϵ is normally distribution, that is $\epsilon \sim N(0, \sigma_\epsilon^2)$, then the the linear parameters are normally distributed around the true solution β_* with covariance matrix $\sigma_\epsilon^2 \left(\mathbf{X}^T\mathbf{X}\right)^{-1}$.

$$\hat{\beta} \sim N(\beta_*, \sigma_\epsilon^2 \left(\mathbf{X}^T\mathbf{X}\right)^{-1}).$$

Linear Models: Parameter Variance

The standard deviation for each coefficient can be read off the diagonal entries of the covariance matrix $\sigma_\epsilon^2 (\mathbf{X}^T\mathbf{X})^{-1}$,

$$\hat{\sigma}_i = \sqrt{\text{Var}(\hat{\beta}_i)} = \sigma_\epsilon \sqrt{(\mathbf{X}^T\mathbf{X})^{-1}_{ii}},$$

where $(\mathbf{X}^T\mathbf{X})^{-1}_{ii}$ is the i'th diagonal entry of $(\mathbf{X}^T\mathbf{X})^{-1}$.

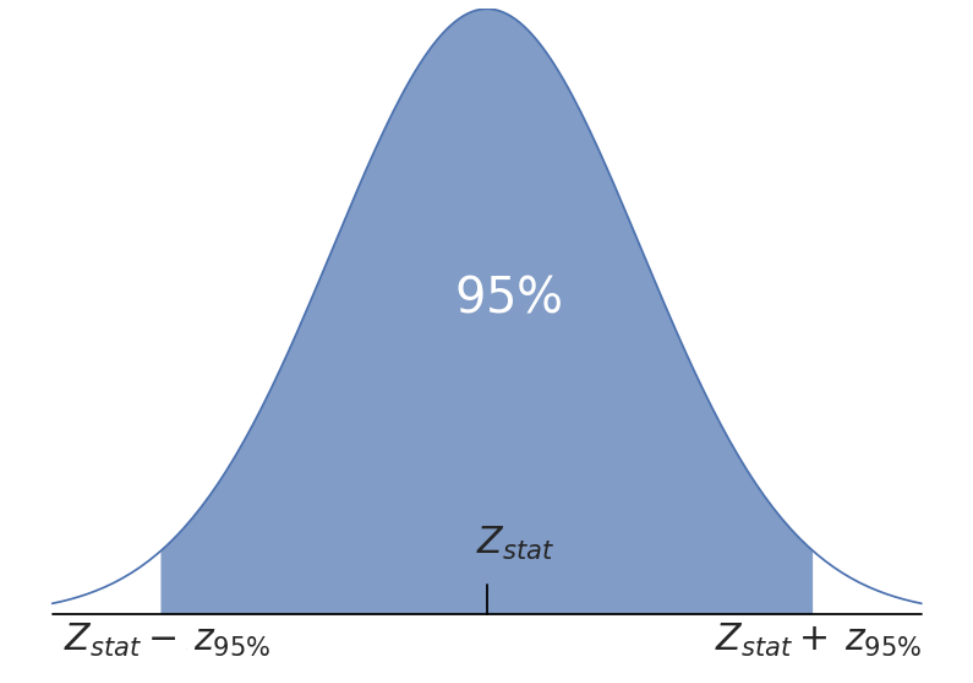
To test the hypothesis that $\beta_i = 0$, that is the i'th feature has no bearing on the outcome, we write the standardized z score

Z-Score of β_i

$$z_i \approx \frac{\hat{\beta}_i}{\hat{\sigma}_i} = \frac{\hat{\beta}_i}{\sigma_\epsilon \sqrt{(\mathbf{X}^T\mathbf{X})^{-1}_{ii}}}.$$

Practically speaking, if $|z_i| > 2$ and N is large, the chance that $(\beta_*)_i = 0$ is less than 5%.

Z-scores



Given a linear distribution with normal noise, the $z_i \sim \mathcal{N}(1, 0)$ are distributed according to a standard normal distribution. The 95% CI around each β is then

$$I_{95} = \left[\hat{\beta}_i - z_{95}\sigma\sqrt{(\mathbf{X}^T\mathbf{X})^{-1}_{ii}}, \hat{\beta}_i + z_{95}\sigma\sqrt{(\mathbf{X}^T\mathbf{X})^{-1}_{ii}}\right]$$

Fit, standard error and z-score on example data

T-score For Application Data

Example: The fit coefficients for the admissions data with their standard errors.

Variable	**coef**	**std err**	**z**
const	0.3257	0.014	24.024
Serial No.	0.0722	0.012	5.884
GRE Score	0.0979	0.032	3.077
TOEFL Score	0.0994	0.032	3.144
University Rating	0.0433	0.020	2.133
SOP	0.0078	0.023	0.336
LOR	0.0663	0.023	2.860
CGPA	0.3077	0.041	7.418
Research	0.0217	0.008	2.587

T-score For Application Data

We find the highest significance for the constant β_0, with the second highest being for GPA.

Variable	coef	std err	z
const	0.3257	0.014	24.024
Serial No.	0.0722	0.012	5.884
GRE Score	0.0979	0.032	3.077
TOEFL Score	0.0994	0.032	3.144
University Rating	0.0433	0.020	2.133
SOP	0.0078	0.023	0.336
LOR	0.0663	0.023	2.860
CGPA	0.3077	0.041	7.418
Research	0.0217	0.008	2.587

T-score For Application Data

Surprisingly, we also find a high significance for the coefficient of the Serial No. This is suspicious, and we see that the coefficient itself is small.

Variable	coef	std err	z
const	0.3257	0.014	24.024
Serial No.	0.0722	0.012	5.884
GRE Score	0.0979	0.032	3.077
TOEFL Score	0.0994	0.032	3.144
University Rating	0.0433	0.020	2.133
SOP	0.0078	0.023	0.336
LOR	0.0663	0.023	2.860
CGPA	0.3077	0.041	7.418
Research	0.0217	0.008	2.587

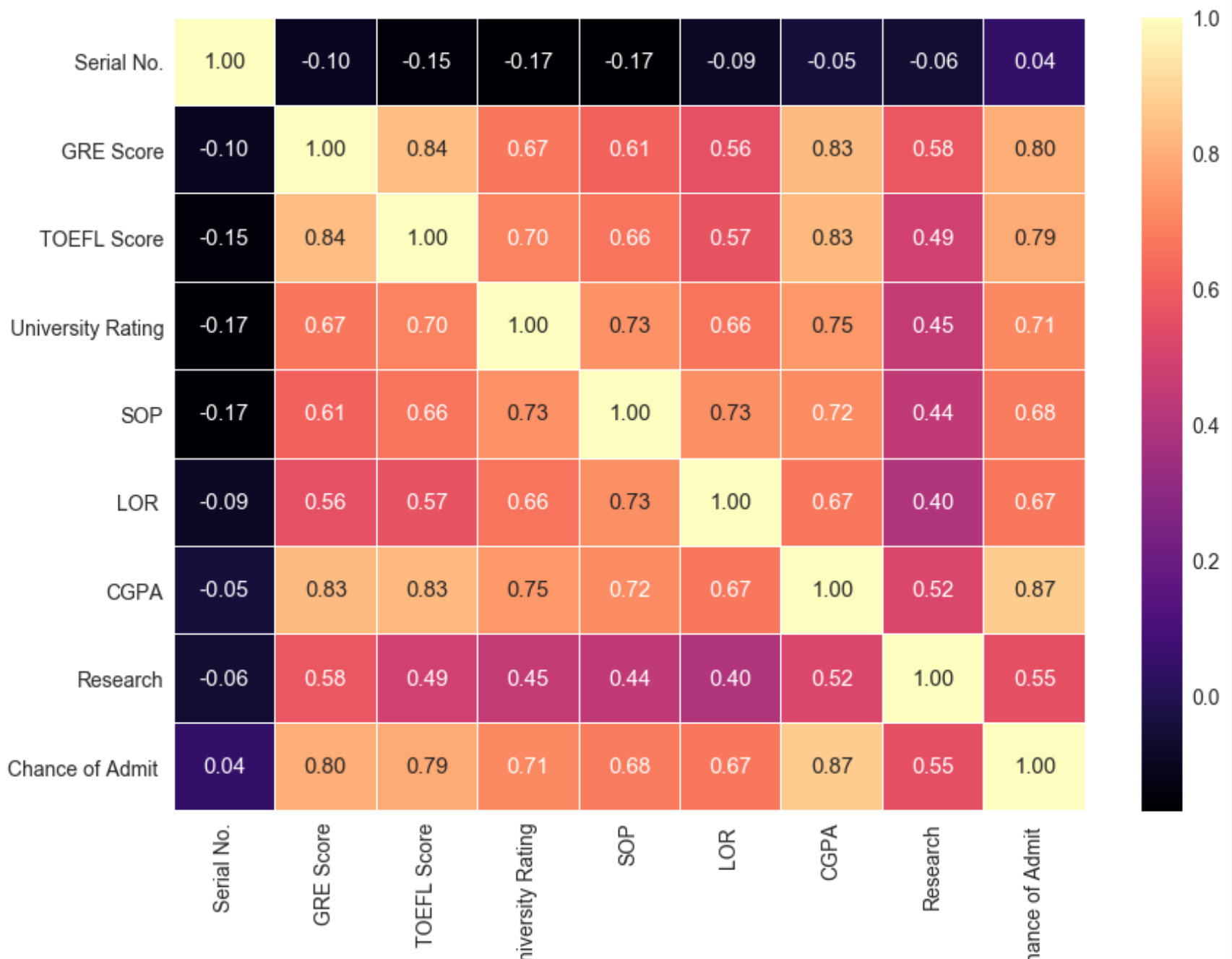
T-score For Application Data

Surprisingly, we also find a high significance for the coefficient of the Serial No. This is suspicious, and we see that the coefficient itself is small.

Variable	coef	std err	z
Serial No.	0.0722	0.012	5.884

This high significance mostly likely comes from the fact that the correlation is so low that the standard error is low. It means that the model may be over fitting, using a effectively random parameter to generate a tighter fit to the specific dataset. Such terms tend to lead to a higher variance in our models, and should be eliminated if possible. If we compute the correlation matrix we will see that Serial No. is effectively uncorrelated with the acceptance percentage.

Correlation



Correlation

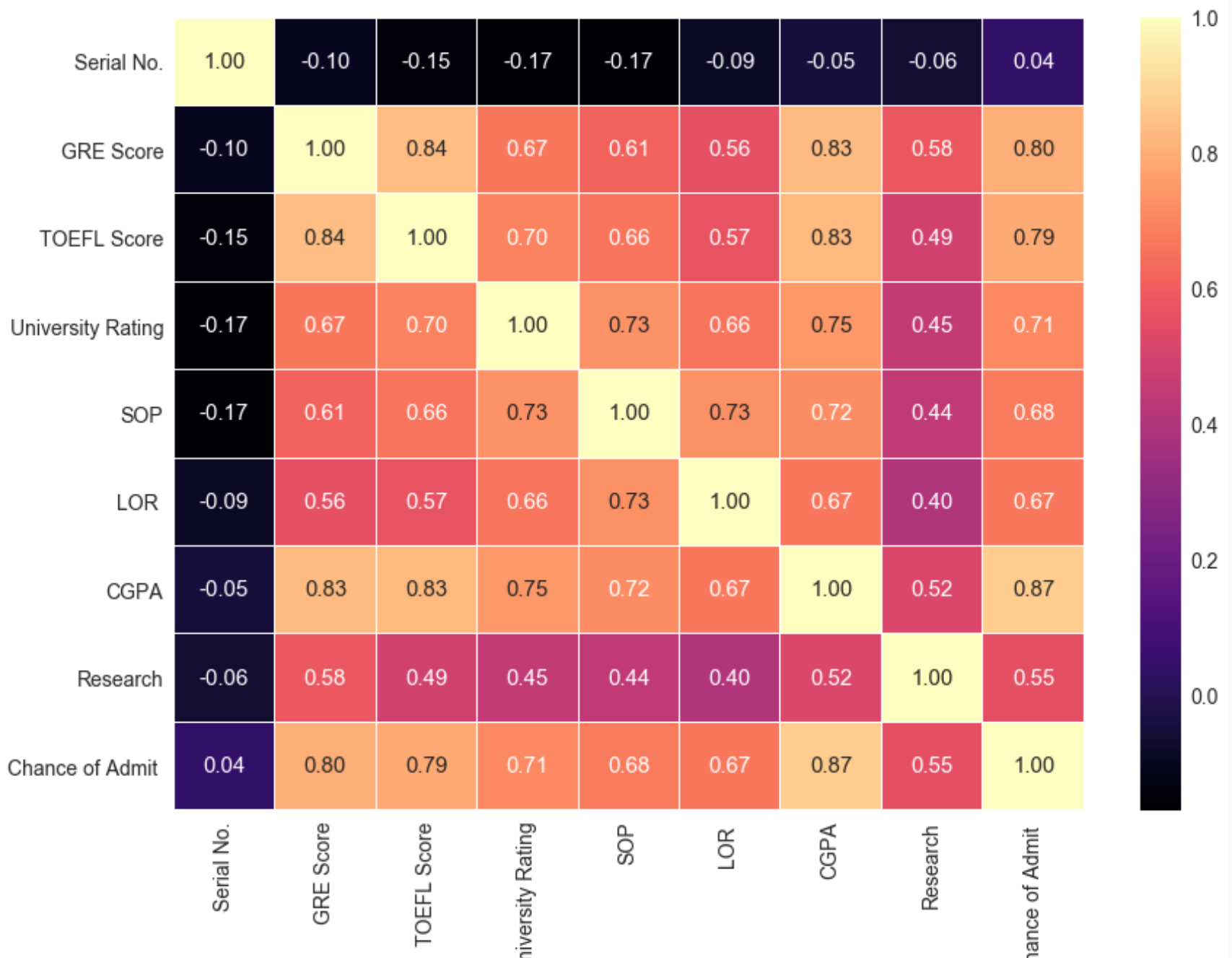
Lets breakdown what we're seeing on the previous slide:

Serial No. is basically uncorrelated with anything.

Admit is highly correlated with **CGPA**, **TOEFL Score** and **GRE Score**

Research has a lowish correlation with **Admit**, but also with everything else.

Correlation



T-score For Application Data

Going back to our fit, we find that **CGPA** has a highly significant coefficient, but **TOEFL** and **GRE** are much more similar to **research**. Why could this be?

Variable	**coef**	**std err**	**z**
const	0.3257	0.014	24.024
Serial No.	0.0722	0.012	5.884
GRE Score	0.0979	0.032	3.077
TOEFL Score	0.0994	0.032	3.144
University Rating	0.0433	0.020	2.133
SOP	0.0078	0.023	0.336
LOR	0.0663	0.023	2.860
CGPA	0.3077	0.041	7.418
Research	0.0217	0.008	2.587

T-score For Application Data

What about **SOP**? The correlation with **Admit** is .68 but the z value of the coefficient is utterly insignificant. Why could that be?

Variable	**coef**	**std err**	**z**
const	0.3257	0.014	24.024
Serial No.	0.0722	0.012	5.884
GRE Score	0.0979	0.032	3.077
TOEFL Score	0.0994	0.032	3.144
University Rating	0.0433	0.020	2.133
SOP	0.0078	0.023	0.336
LOR	0.0663	0.023	2.860
CGPA	0.3077	0.041	7.418
Research	0.0217	0.008	2.587

T-score For Application Data

The answer is probably internal correlation. The model gets the best fit by using **CGPA** to account for general trends and the other variables, highly correlated with it, to fine tune the fit.

Variable	**coef**	**std err**	**z**
const	0.3257	0.014	24.024
Serial No.	0.0722	0.012	5.884
GRE Score	0.0979	0.032	3.077
TOEFL Score	0.0994	0.032	3.144
University Rating	0.0433	0.020	2.133
SOP	0.0078	0.023	0.336
LOR	0.0663	0.023	2.860
CGPA	0.3077	0.041	7.418
Research	0.0217	0.008	2.587

This could be useful, or it lead to overfitting.

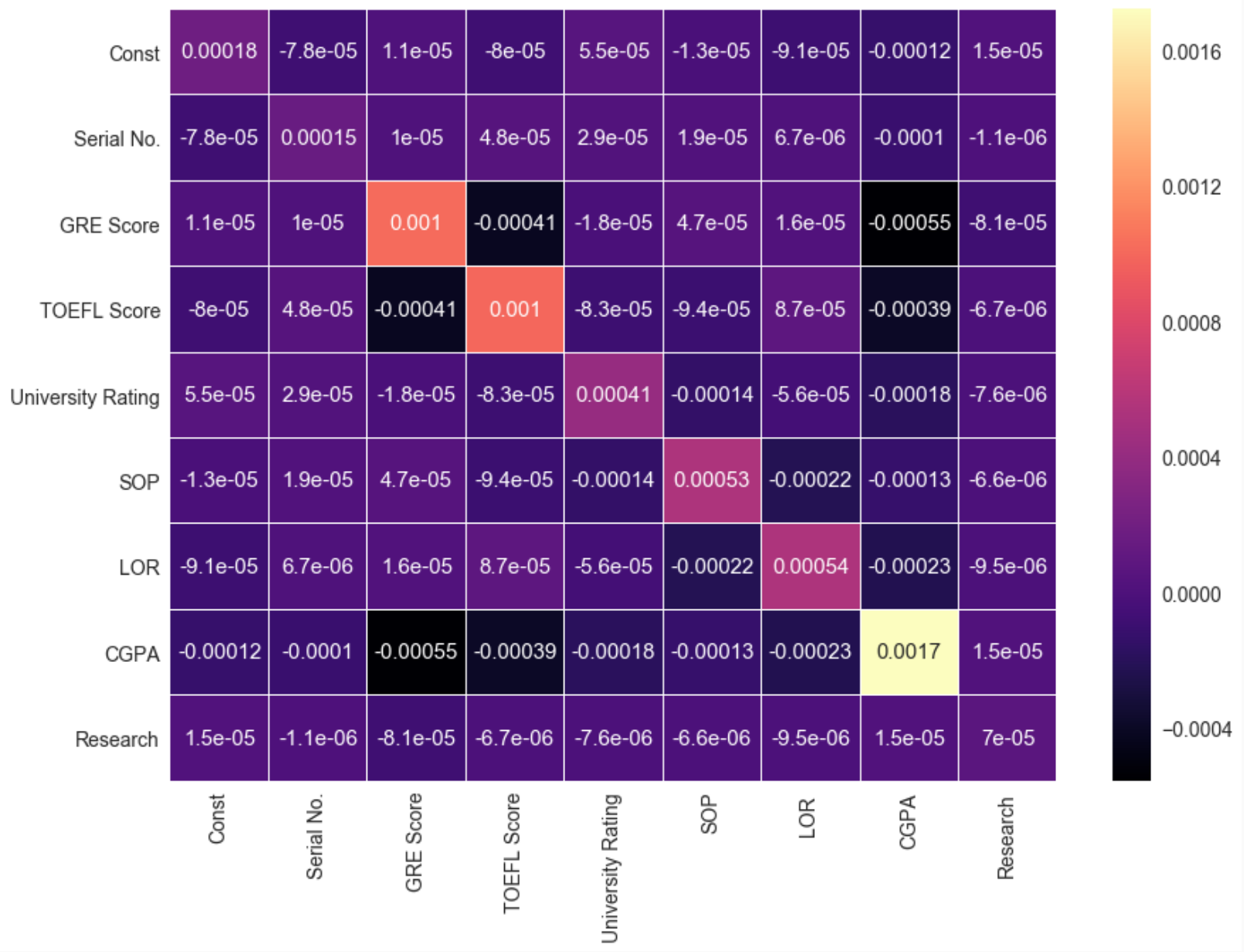
T-score For Application Data

For comparison, lets recall the variance matrix for the parameters

Variable	**coef**	**std err**	**z**
const	0.3257	0.014	24.024
Serial No.	0.0722	0.012	5.884
GRE Score	0.0979	0.032	3.077
TOEFL Score	0.0994	0.032	3.144
University Rating	0.0433	0.020	2.133
SOP	0.0078	0.023	0.336
LOR	0.0663	0.023	2.860
CGPA	0.3077	0.041	7.418
Research	0.0217	0.008	2.587

Variance for Admission

Varaiance Matrix $\text{Var}(\hat{\beta}) = \hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}$



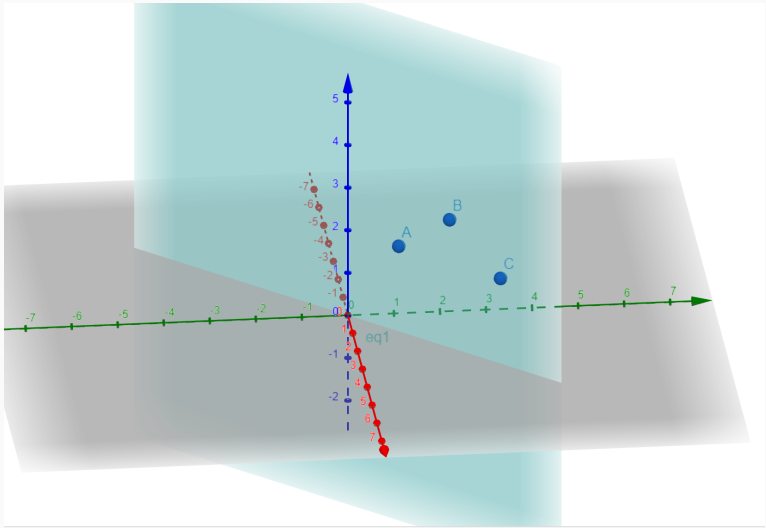
Variance for Admission

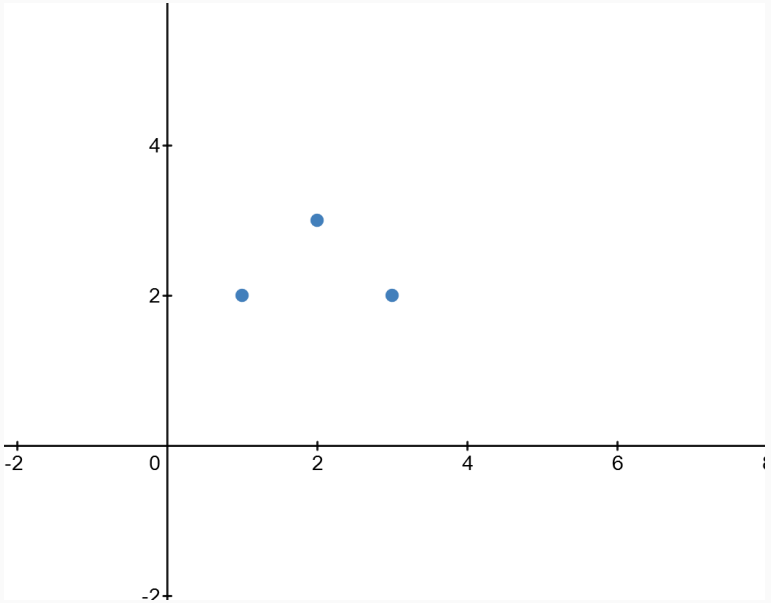
These values are probably low due to to the relatively large sample size and the small number of parameters. Note that we we see that **CGPA** has a negative covariance with **GRE** and **TOEFL**.

This is what we would expect, since all three are highly correlated we would expect a lessening of fit by **CGPA** could be compensated for by a tightening of the fit on **GRE** or **TOEFL**.

Feature Selection via Statistical Significance

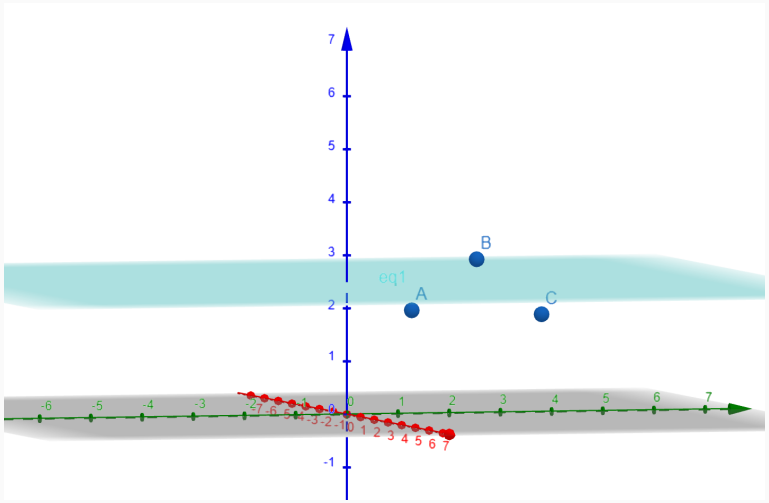
Overfitting

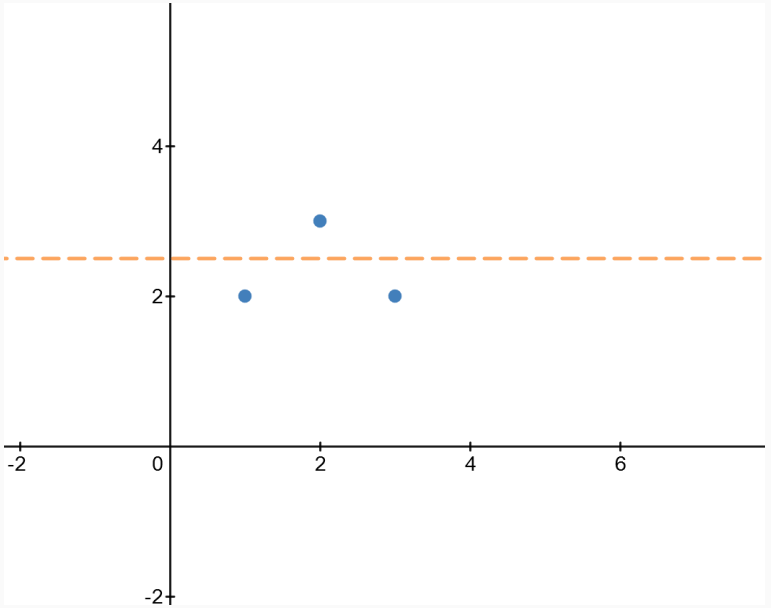




One of the goals of preliminary data analysis is to determine if variables are **redundant** or **random**, since both can lead to additional unnecessary computations, or overfitting. For example, assume that X_1 and X_2 are linearly correlated, that is $X_2 = mX_1$. Then all of the datapoints lie on the hyperplane $X_2 = mX_1$ and the problem is simply $p - 1$ dimensional.

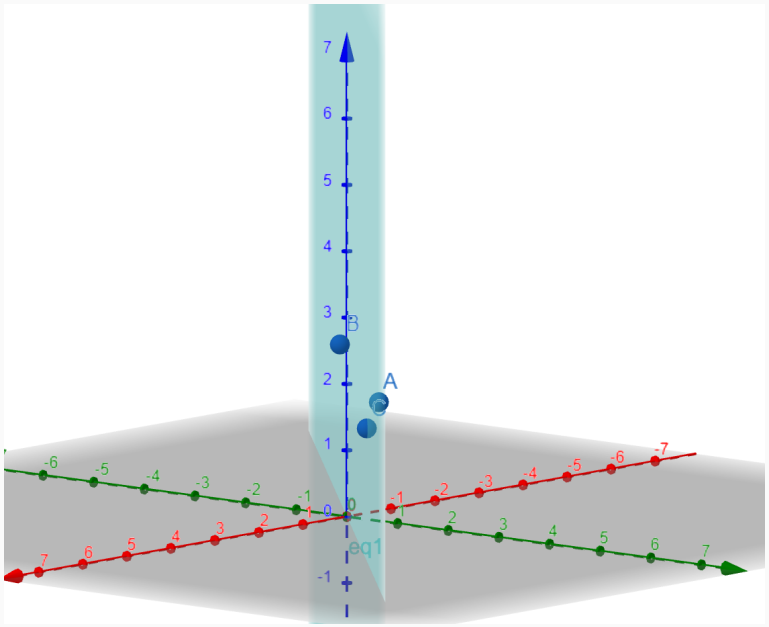
Overfitting

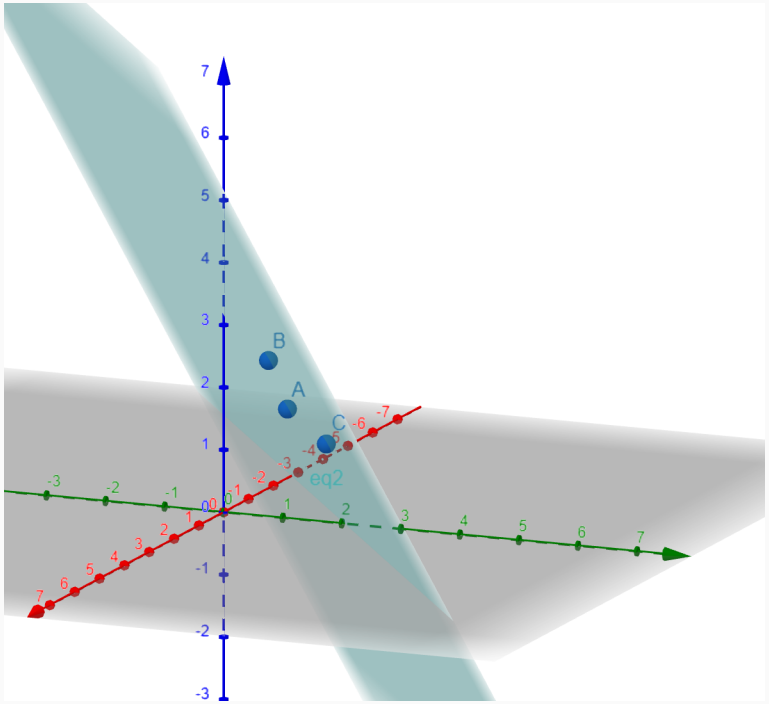




Notice that any rotation of this plane that includes the horizontal fit line will *also* be a best fit line for the data.

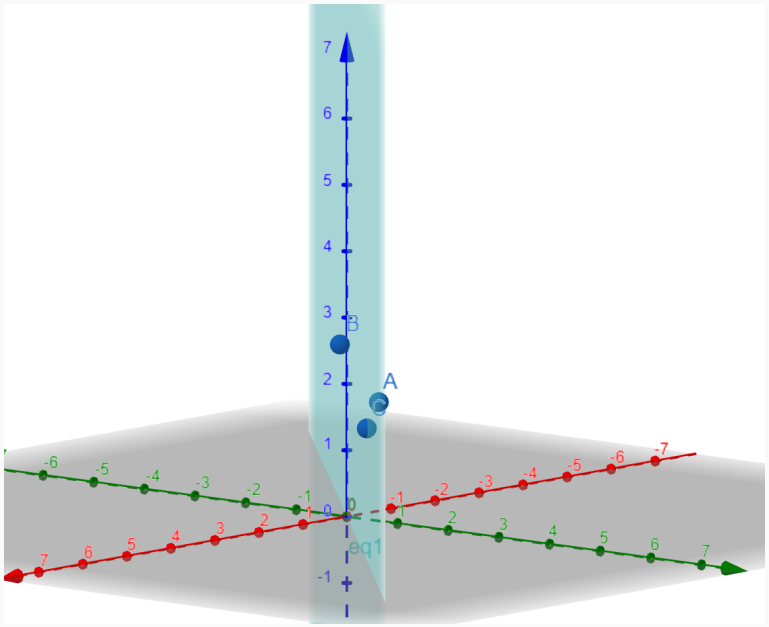
Overfitting

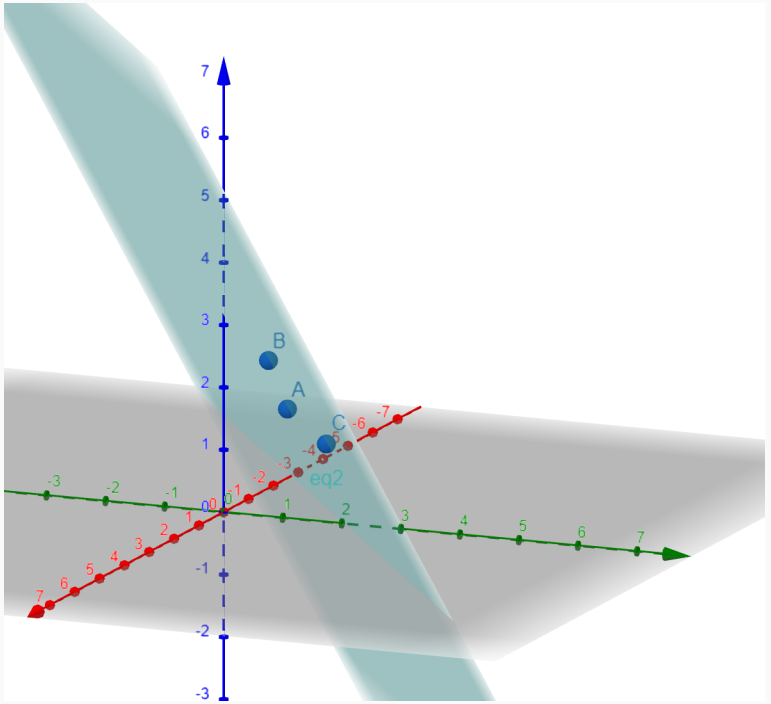




On the other hand, assume that $X_2 = mX_1 + \epsilon$. The data points no longer lie on a hyperplane, but clustered very close to one. The linear function will then effectively fit to X_1, and then use X_2 to try to capture the other points, possibly resulting in absurdly high β_2. If we genuinely think X_1 and X_2 capture different data, this is a feature; if we think say X_1 and X_2 differ by sampling error or encoding error we may have over fit.

Overfitting





If X_1 has no correlation with the labels, we need to worry about overfitting as well. A truly uncorrelated variable at best gives us no information and so is a computational hazard and at worst may allow a linear classifier to over fit, since by a linear transform of the data $(X_1, mX_1 + \epsilon) \to (X_1, \epsilon)$. In short, random features increase model variance.

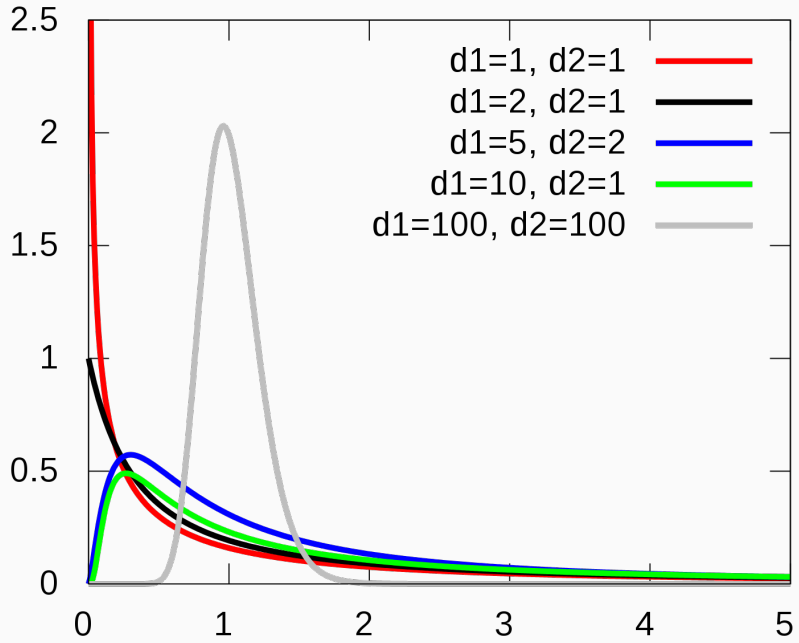
Overfitting

One of the main challenges of predictive statistics is figuring out how to identify and reduce model variance.

> A probability approach will use the F-statistic to compare significance of dropping groupings of variables from the model.
>
> A more computational approach will iterate over all possible subsets of features, pulling out the smallest number of features that give a consistent error.
>
> A more machine learning approach asks us to modify the loss function, attempting to punish the algorithm for incorrect coefficients.

We will explore all of these options in the lectures and labs, starting with a review of F-statistics.

F-statistic



The $F_{d1,d2}$ distribution is used the compare the ratio of sample variances (normalized by the true variances) for normally distributed random variables, where $d1$ and $d2$ are the respective number of degrees of freedom in each sample. Formally,

$$F \sim \frac{s_1^2}{\sigma_1^2} \Big/ \frac{s_2^2}{\sigma_2^2} \,.$$

F-statistic

Consider two models, models A and B, with model A fitting a subset of the parameters of model B. Assume model A has p_A parameters and model B has $p_B > p_A$ parameters.

The F-statistic measure the relative sum squared of residuals between the two models, which in turn allows us to use the F distribution to test the level of significance.

Computing RSS on all the test data, the F-statistic is defined as

$$F = \frac{(RSS_A - RSS_B)/(p_B - p_A)}{RSS_B/(N - p_B - 1)}.$$

The F statistic measures if the removed β_j are indeed 0.

F-statistic

Under the Gaussian assumption for ϵ and the null hypothesis that the smaller model (model A) is the true model, the F-statistic will have a

$$F \sim F_{p_B - p_A, N - p_B - 1}$$

distribution. The F distribution assess the statistical significance of setting the β_j dropped from model A to 0. If significance is found, we should assume the variables are not redundant.

Exercise: Show that that square of the z-score z_j is the F statistics of dropping the single variable X_i from the model.

T-score For Application Data

For example, we could consider dropping the insignificant **SOP** and the significant but uncorrelated **Serial No.** from the classifier.

Variable	**coef**	**std err**	**z**
const	0.3257	0.014	24.024
Serial No.	0.0722	0.012	5.884
GRE Score	0.0979	0.032	3.077
TOEFL Score	0.0994	0.032	3.144
University Rating	0.0433	0.020	2.133
SOP	0.0078	0.023	0.336
LOR	0.0663	0.023	2.860
CGPA	0.3077	0.041	7.418
Research	0.0217	0.008	2.587

F-statistic

Comparing this new model to the old model give a test statistics

$$F = \frac{(RSS_A - RSS_B)/(p_B - p_A)}{RSS_B/(N - p_B - 1)}\,.$$

is

$$F = \frac{(1.239 - 1.114)/(8 - 6)}{1.114/(320 - 9)} = 17.45$$

Checking against $F_{6,311}$, we find that this is highly statistically significant, as we predicted from the standard error. By contrast, the F statistic for just dropping **SOP** is

$$F = \frac{(1.114861 - 1.114456)/(8 - 7)}{1.114/(320 - 9)} = 0.1130 \approx (0.336)^2\,.$$

Checking against $F_{7,311}$, F is not statistically significant, so we can safely drop it.

F-statistic

We've come to an impasse: Correlation tells us to drop the feature **Serial No.** while the F statistic (and the z_j number) tell us it is significant. How do we understand this?

> There is actually a small but significant correlation between the serial number and the chance of admit, we should not drop the variable but should note this.
>
> The linear model is over fitting, we should drop the variable.
>
> The fit is bad enough

Subset Selection Methods

Subset Selection

There are a few reasons why we might not be satisfied with the least squares estimate.

Prediction accuracy: Least squares estimates often have low bias at the cost of high variance. One is often able to reduce variance by selectively setting some coefficients to 0.

Interpretive power: With a large number of related parameters, one often wants to find a small number of predictors with strong, definite effects. You don't have to do quantum mechanics to build a car, and often when looking for the "big picture" we are willing to sacrifice some granularity.

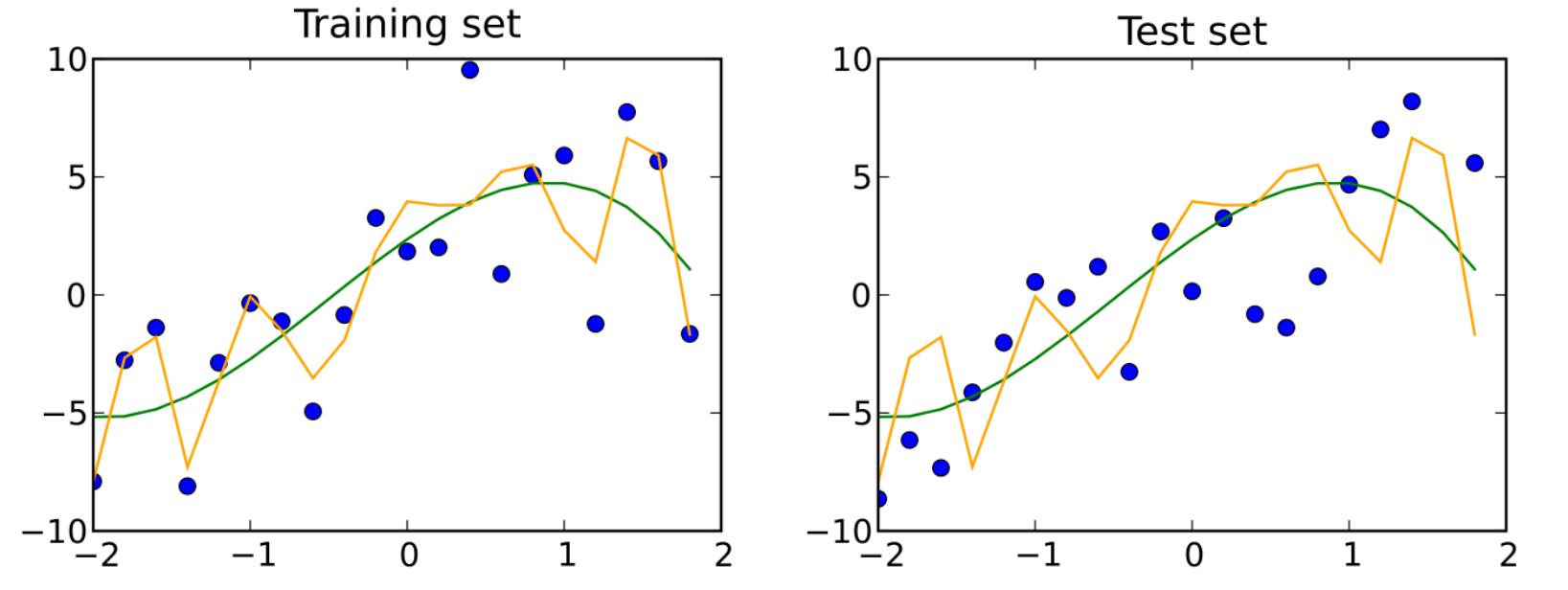
Best Subset Selection

In best subset selection, we perform a regression on every subset of variables of size k and pick the model with the smallest mean squared error. There are efficient algorithms (for example the *leaps and bounds* procedure) that make this possible for p up to 30 or 40, but the total number of subsets grows exponentially

$$Pow(\{1, \ldots, k\}) = 2^k \, .$$

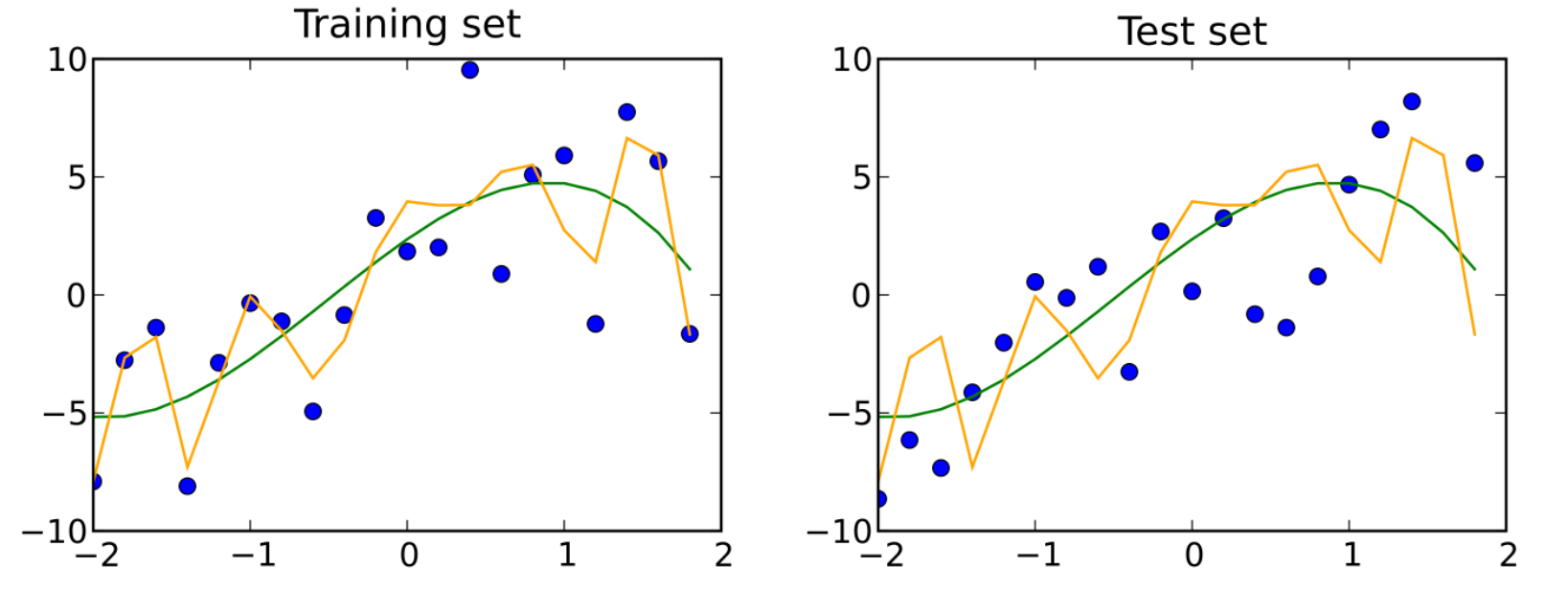
The question becomes how to choose k. There are many criteria, typically one uses the smallest model that minimizes the estimates of the expected prediction error.

Best Subset Selection



A common way to evaluate models is to randomly cut the available data into a training set and a test set. The exact percentages of data to use for training and test depend a bit on the problem, but an 80%-20% split between training and test data is common. Here, we see how low variance data fits better to the test set.

Best Subset Selection



In best subset selection, we train each of our models on a random 80% of the training data and then validate on the remaining 20%. If we have the computational power, we may do this for many random partitions of the data into training and test sets.

Best Subset Selection

For the Student Admissions data, we can perform best subset selection for subsets of all size k.

For $k = 1$, we are fitting a linear model to only one variable and picking out the best test regression.

Variable	**MSE train**	**MSE test**
CGPA	0.004629	0.005665
TOEFL Score	0.007271	0.008874
GRE Score	0.006701	0.009308
LOR	0.011041	0.011981
University Rating	0.009177	0.013495
SOP	0.010336	0.013964
Research	0.013240	0.017531
Serial No.	0.018787	0.026214

Best Subset Selection

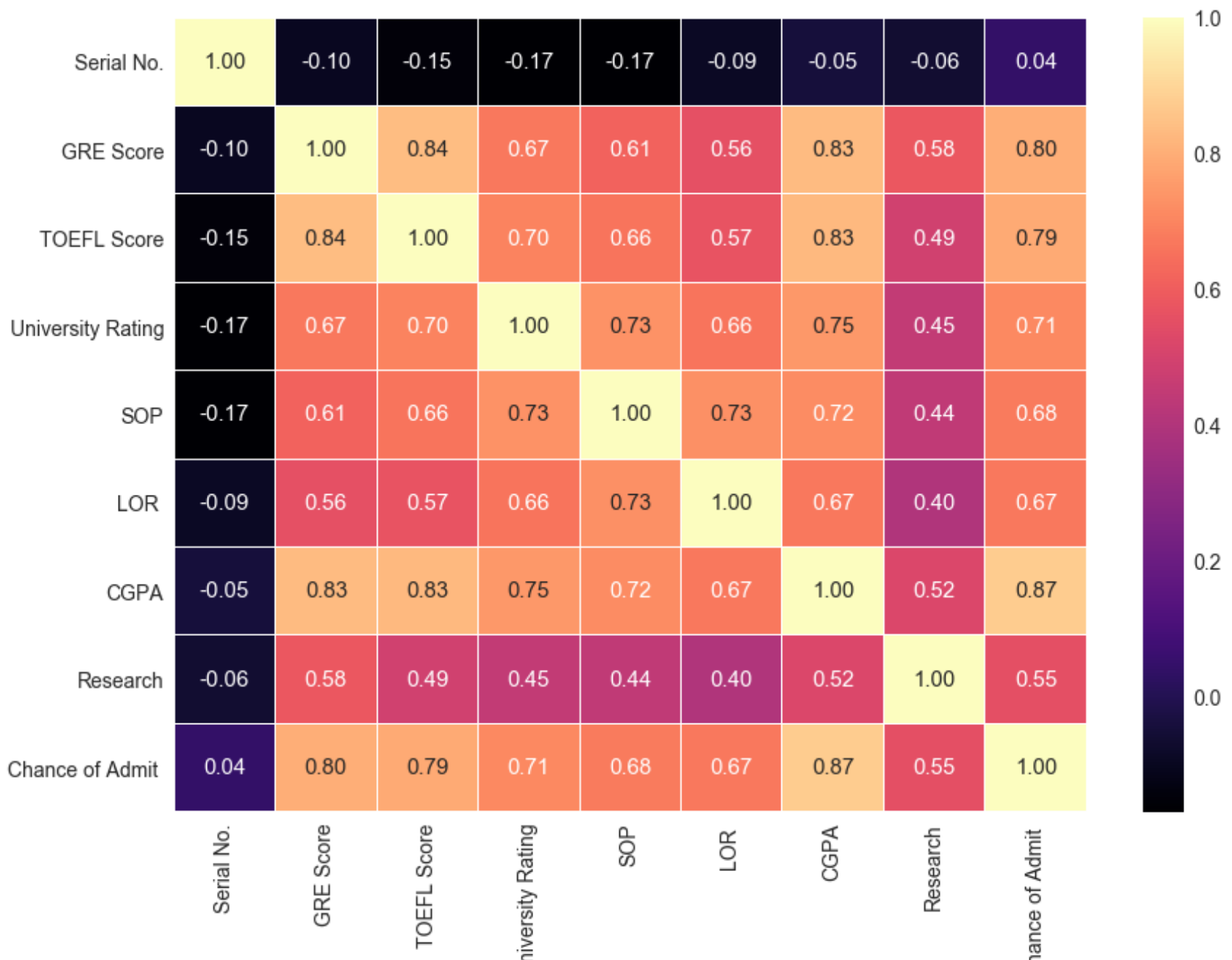
Another common human readable scoring mechanism is the r^2 statistic

$$r^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

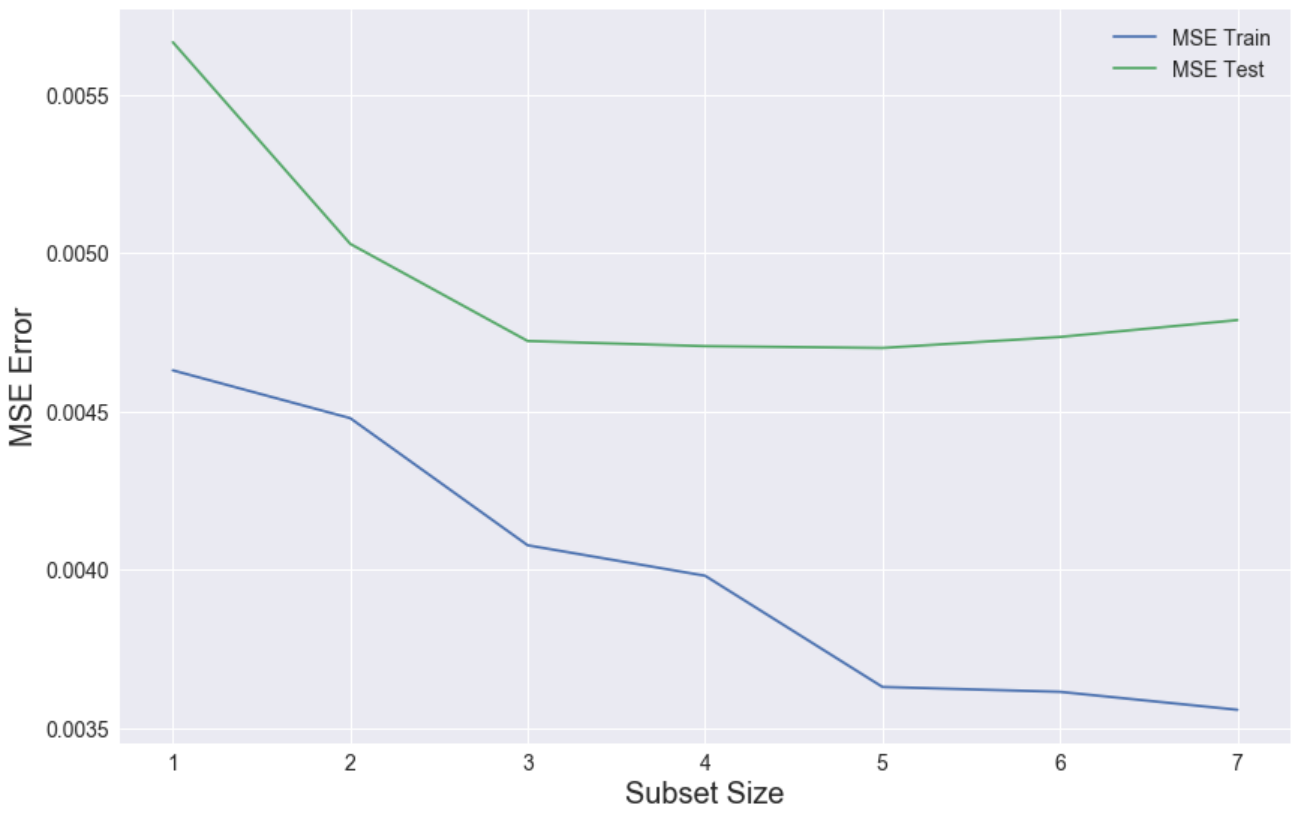
where $r^2 = 1$ is perfect correlation.

Variable	r^2 **Train**	r^2 **Test**
CGPA	0.780614	0.755038
GRE Score	0.639539	0.645389
TOEFL Score	0.656348	0.615228
University Rating	0.477414	0.514386
SOP	0.459257	0.453055
LOR	0.536031	0.415755
Research	0.321136	0.299415
Serial No.	-0.015145	0.005851

Correlation

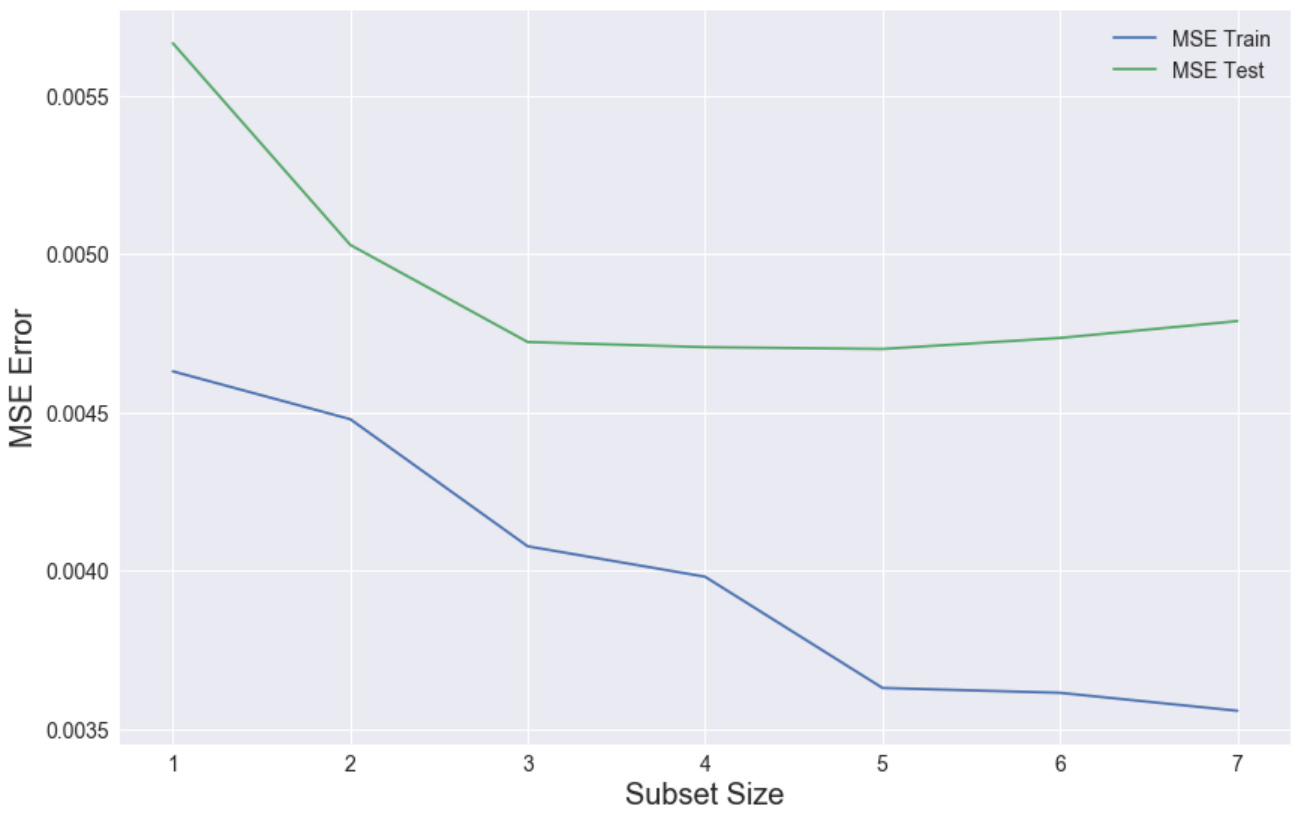


Best Subset Selection



For $k = 1$ the best subset regression matches up with the correlation matrix. As we increase k, the training error continuous to drop, but the test error levels out and even starts to rise a bit for $k > 5$.

Best Subset Selection



It turns out the best subsets use **GRE**, **LOR** and **CGPA**. This may be a little surprising given that coefficient of **LOR** scored rather middling significance in the full model. Why could this be?

Forward Subset Selection

For enough features, the 2^p possible subsets of variables becomes computationally expensive. **Forward subset selection** tries to forge a good path through the forest of parameter choices.

The algorithm starts by fitting the intercept, and then sequentially adds into the model the variable that most improves the fit. The algorithm produces a nested sequence of models and is classified as a **greedy algorithm** since it makes local optimization choices at each steps instead of global ones.

Forward subset selection has a number of regression steps polynomial in the number of features, and can be made even more efficient with a bit of linear algebra.

Backward Subset Selection

Backward subset regression starts with the full model and sequentially deletes the predictor with the smallest z-score. It can only be used when the data set is larger than the number of features, ie when $N > p$.

Backward subset selection is particularly useful when you expect to have a large number of features contributing to the final fit.

Forward and Backward Subset Selection

There are two main advantages to forward and backward subset selection:

Computational: Forward/backward subset selection are more computationally efficient then performing an exhaustive search of all subsets of variables.

Stability: If we allow our model to select the best subset of the features, we must pay the price of a larger variance. As always, we may be able to afford this, but for small training sets we may be more interested in stability.

We will be implementing these methods over the next (few) labs.

References

This lecture was adapted primarily from Elements of Statistical Learning, Chapter 3.

For a good explanation of the proof that $\hat{\sigma}$ is an unbiased estimator, see

`https://stats.stackexchange.com/questions/20227/why-is-rss-distributed-chi-square-times-n-p`

Geogebra and Desmose were used for pictures.