Machine Learning

Robson Glasscock, PhD, CPA

Fall 2020

Big Picture

Statistical models are used to make predictions, forecasts, or understand associations, correlations, and relationships between things.

It is often possible to express something of interest as an output that changes based on a series of inputs.

What positions taken on a tax return are most likely to trigger an IRS audit?

Can characteristics of journal entries be used to identify errors or fraud?

Are there any transactions that appear to be anomalies that an auditor should focus on?

Big Picture (cont.)

Other examples of expressing something of interest as an output that changes based on a series of inputs include:

> How much of COGS are fixed costs vs. variable costs?
>
> Did an accident caused by a counterparty result in economic damages to a business?
>
> What are ESOPs worth based on estimates of future cash flows?

Using data to understand relationships is a powerful tool to aid the decision making process.

Big Picture (cont.)

Today we will use housing price data as an intuitive example to:

Explore the relationships between various inputs and house prices

Some of the inputs **we would expect** to be valid predictors of house prices

Some of the inputs **we would <u>not</u> expect** to be valid predictors of house prices

And then you will see how a popular machine learning algorithm makes housing price predictions.

The steps we will go through are a flexible and can be applied to a variety of business decisions and situations.

Types of Machine Learning

Supervised Learning: Build a model that predicts outcomes, classifications, or values of something based on inputs.

Ex: Is an email spam or not based upon keywords or phrases?

Unsupervised Learning: Identify various groups within the data or reduce the data to a simpler mapping of characteristics.

Ex: Can questions on a 50-item questionnaire be reduced or grouped into common components?

Predicting Outcomes, Classifications, or Values

What show should Netflix recommend to someone based on what they have just finished watching?

Does a clinical trial indicate that a new drug is effective?

Which products is an Amazon Prime member likely to buy?

Are there company-specific characteristics that are predictive of future stock market returns?

Predicting Outcomes, Classifications, or Values (cont.)

Consider the price of a home in the U.S.

What "variables" or "features" do you think would be associated with the selling price?

Let's use modified house price data based on Kiel and McClain (1995) as an example.

The Kiel and McClain (1995) house price date runs from 1974 to 1992.

Housing Price Data

The dataset contains the following variables:

Variable We Are Interested in Predicting

Price: Selling price of the home

Variables That Are Probably Good Predictors

Area: Square footage of the home

Age: Age of the home

Agesq: Age of the home, squared

Variables That Are Probably Not Good Predictors

Grades: Final grades from my undergraduate Auditing courses

Rand: A randomly generated variable ~N(80.46, 11.64)

Housing Price Data (cont.)

Our first step is to take a look at some of the observations in the dataset.

Housing Price Data (cont.)

Price	Area	Age	Agesq	Grades	Rand
59,000	2,128	7	49	81	79
94,000	2,290	0	0	81	89
95,920	2,464	0	0	80	90
95,000	2,240	0	0	79	78
95,900	2,464	0	0	79	87
91,000	2,240	1	1	78	83
96,900	2,464	0	0	78	77
47,000	2,576	104	10,816	78	71
68,500	1,377	18	324	77	83
36,000	2,071	188	35,344	77	74

Housing Price Data (cont.)

We can see that the prices look relatively low, but they are not out of line with national summary statistics from this period.

At least some houses in the sample were either built in the early 1800's or these are errors.

Looking at summary statistics may be able to tell us more than just looking at a few observations.

Some Useful Statistics

Statistic	Excel Function	Measure of	What It Tells You	Price
Mean	=AVERAGE()	Central Tendency	This is the expected value of the distribution. It is calculated as an equally-weighted sum of all the values of price in the dataset.	$96,100.66
Median	=MEDIAN()	Central Tendency	Similar to the Mean but less influenced by outliers. The value of price where half of the observations in the data are larger than and half are smaller than.	$85,900.00
Variance	=VAR.S()	Dispersion	The sum of the difference between each house's price and the Mean, squared. This is harder to interpret than the Standard Deviation because the units are dollars, squared.	$1,868,290,835.11
Standard Deviation	=STDEV.S()	Dispersion	The square root of the Variance. A measure of dispersion from the Mean in non-squared dollars which is usually easier for people to interpret than the Variance.	$43,223.73
Covariance	=COVARIANCE.S()	Association	Measure of the linear relationship between two variables (e.g., price and area). Tells you if, on average, when one variable is above its Mean if the other variable is also above its Mean.	$19,385,134.69
Correlation	=CORREL()	Association	Measure of the linear relationship between two variables (e.g. price and area) bounded between -1 and +1. Calculated by the Covariance of the two variables divided by the product of each variable's Standard Deviation.	0.65

Summary Statistics

Variable	Obs	Mean	Std. Dev.	Min	Max
Price	321	96,100.66	43,223.73	26,000.00	300,000.00
Area	321	2,106.73	694.96	735.00	5,136.00
Age	321	18.01	32.57	0.00	189.00
Agesq	321	1,381.57	4,801.79	0.00	35,721.00
Grades	321	80.46	11.64	8.00	95.00
Rand	321	79.91	11.07	46.78	106.81

Legend:
Obs: Observation number
Mean: The average
Std. Dev.: The standard deviation
Min: The minimum value
Max: The maximum value

Summary Statistics (cont.)

This dataset does not contain any missing values for any of the variables.

The average house is 18 years old, the standard deviation is actually larger than the average, and some of the houses are brand new.

The randomly generated variable appears to follow the same distribution as grades.

Housing Price Data (cont.)

Now that we have more of a feel for the data we have, how do you think the prices of homes are related to each variable?

Our goal is to make the most accurate predictions we can for the price of any given house.

The predicted values can be plotted against the actual values as a line through the data.

We want this line to be the best fitting line possible, which will minimize the prediction errors.

Housing Price Data (cont.)

Visually inspecting the relationships between house prices and the various features one could use for predictions is a useful starting point to build a pricing model.

The next series of slides contains:

1) graphs of house price vs. the variables in the dataset

2) the inclusion of a red line with the optimal predictions based on each variable in the dataset.

Price and Area
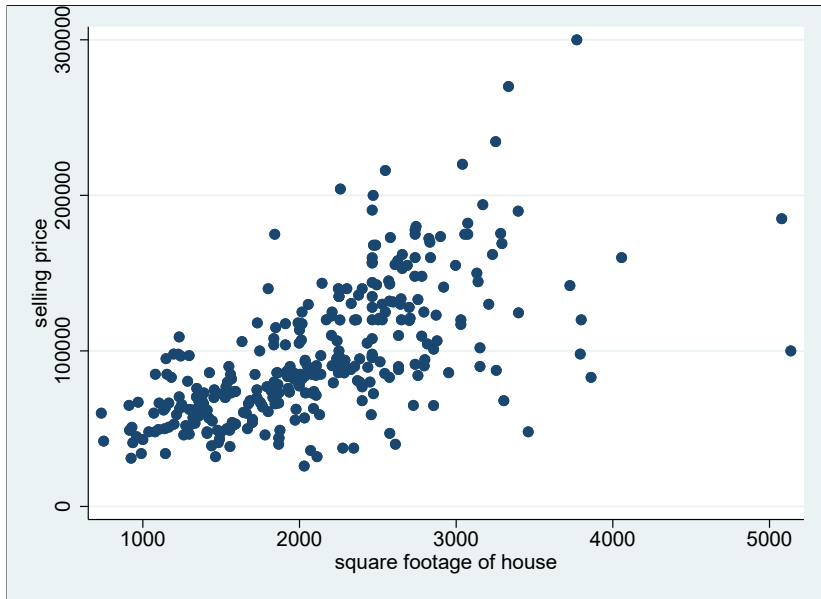
Price and Area Fitted

Price and Area Fitted (cont.)

We can see that price appears to generally increase linearly with size.

As size increases, the range of housing prices also increases. In statistical terms, this is called "heteroskedasticity" and has implications for hypothesis testing.

Price increasing with size is probably what most of us expected.

Let's take another look at the graph.

Possible Outliers

Outlier Considerations

There are two houses over 5,000 square feet.

Most of the other houses are between 1,000 and 3,000 square feet with a handful of houses around 4,000 square feet.

Note that our summary statistics revealed that the average area is 2,106 square feet.

The houses larger than 5,000 square feet may be outliers, and we should consider how much including these houses in the sample impact our predictions.

If we have a business reason to only be interested in pricing smaller or average size houses then these observations may be worth excluding.

Price and Age
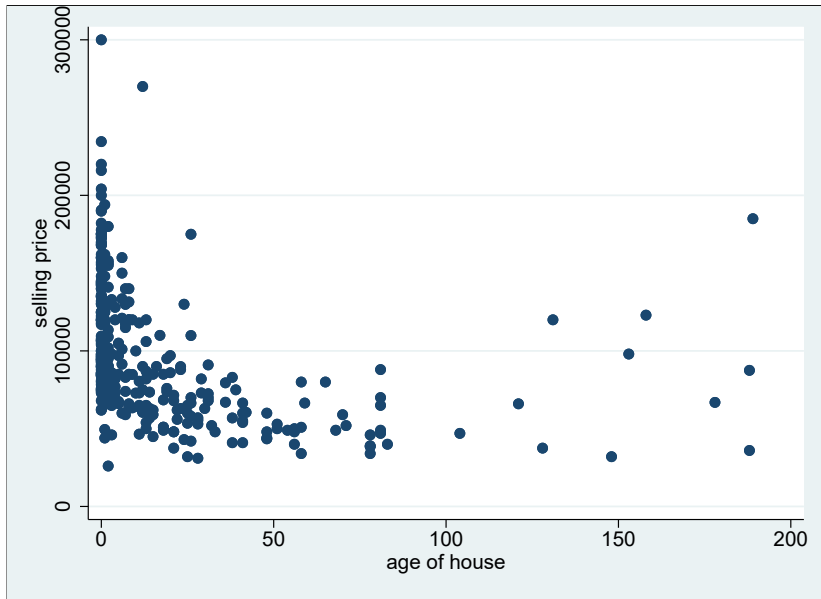
# Price and Age Fitted
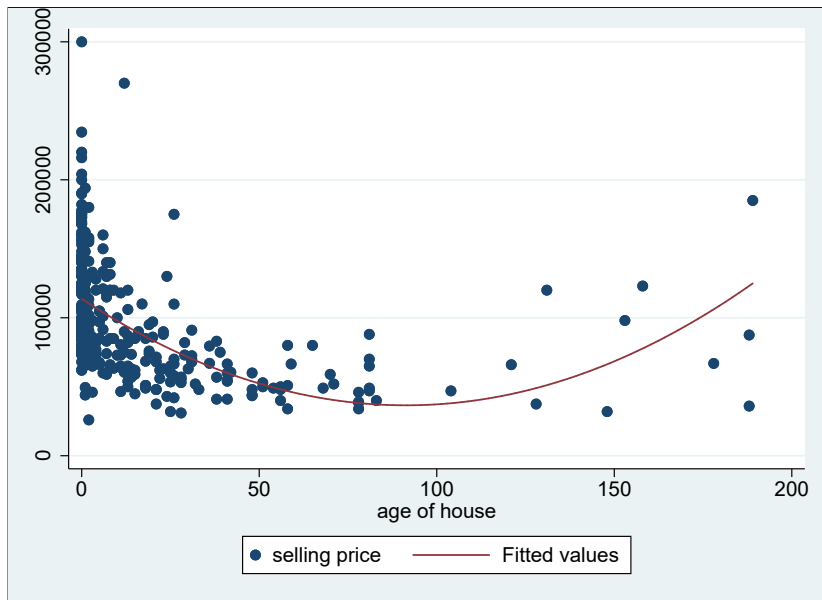
Price and Age Fitted (cont.)

Price appears to be higher for newer homes, but with notable variance.

Price then decreases as homes age before increasing again for the oldest homes in the sample.

The first half of the graph is probably what most of us expected, and the second half may be explained by the oldest homes being 19th century mansions.

The relationship appears to be nonlinear, and we have reason to test for heteroskedasticity later.

Price and Grades
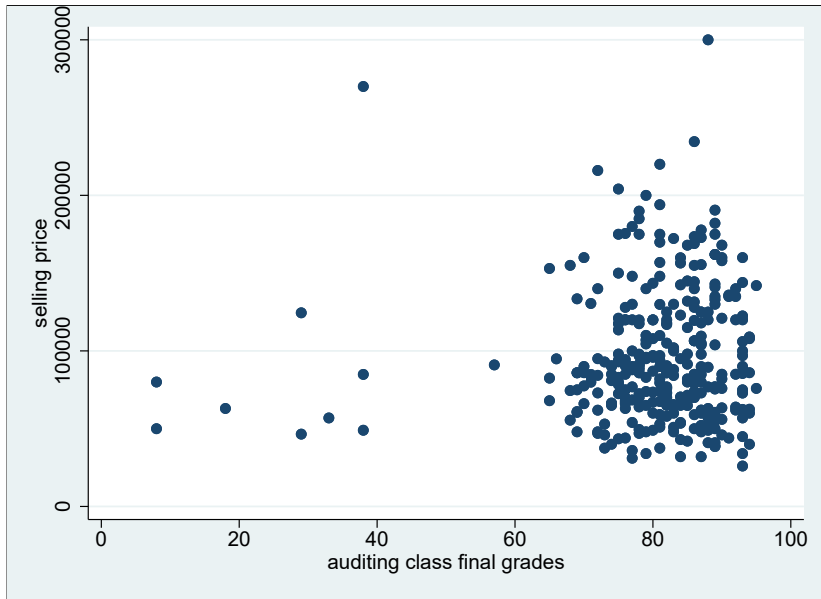
Price and Grades Fitted
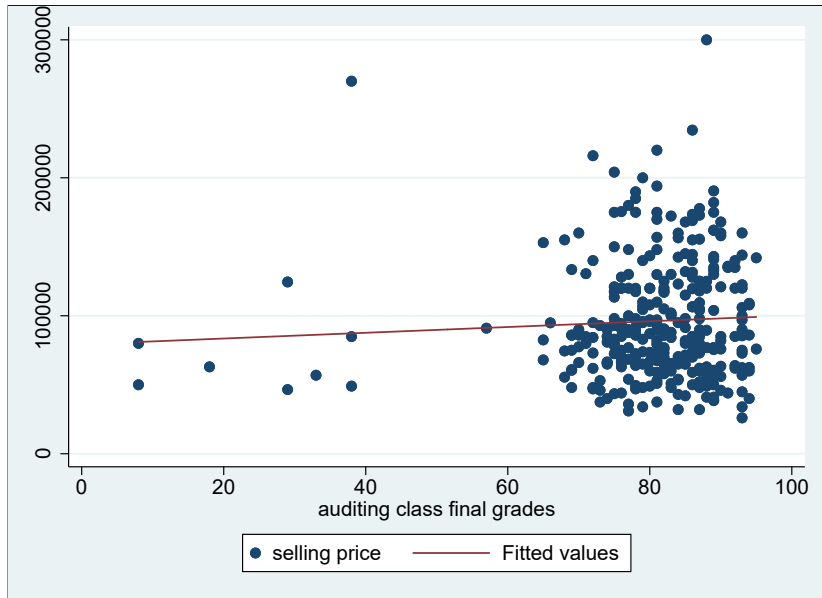
Price and Grades Fitted

The right side of the graph looks like a cloud with no clear pattern of changes in price as grades increase.

Prices are observed between approximately $60,000 and $200,000 for grades that are approximately between 75 and 90.

Plotting two variables together and seeing a cloud is typically indicative of there being no statistical relationship between the variables.
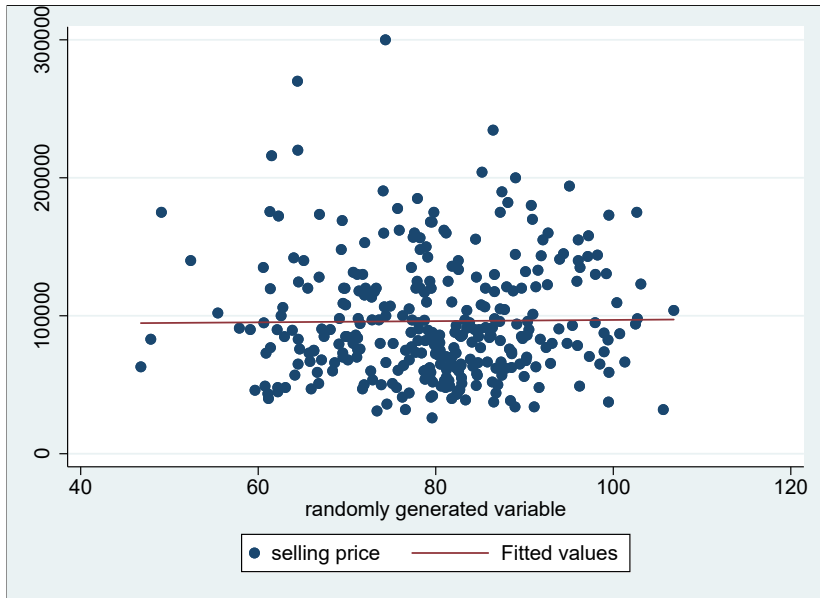
Price and Grades Fitted (cont.)

A line with a slightly positive slope may still be plotted against the data, but this doesn't mean there is a significant relationship between these variables.

In fact, it is unreasonable to believe that undergraduate final grades between 2014 and 2018 impacted Massachusetts housing prices prior to 1993.

Price and Rand
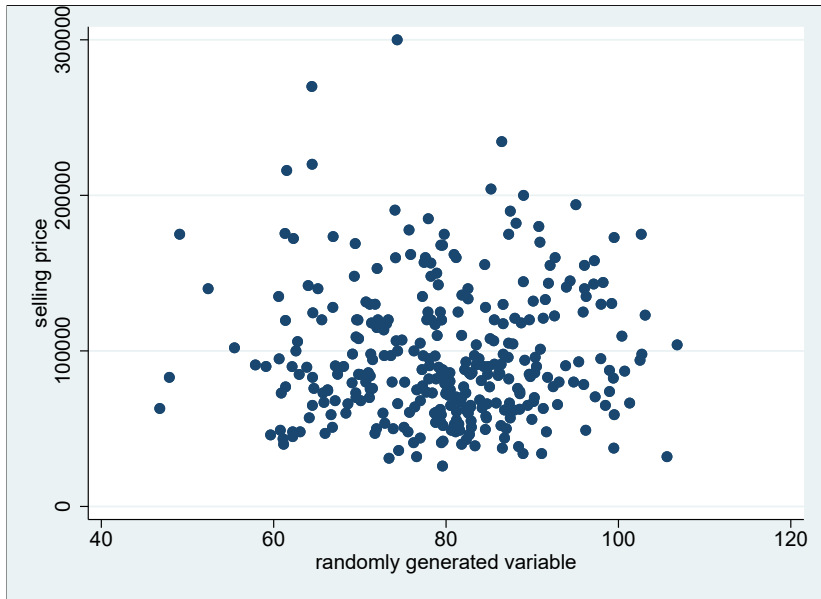
Price and Rand Fitted

Price and Rand Fitted

The cloud pattern is even more apparent when plotting the randomly generated data against house prices.

A line with a slightly positive slope is still able to be plotted against the data, but this doesn't mean there is a significant relationship between these variables.

Visually, the size of the house and the age of the house appear to have some association with the price of the house.

Grades from students who were not yet born when the housing data was collected and a randomly generated variable do not appear to be related to the housing prices.

Key Findings

Area has a positive, linear relationship with house price. This is a valid input to a pricing model.

Age has a nonlinear relationship with housing prices. The relationship is negative until around 100 years of age when it becomes positive. This is a valid input to a pricing model.
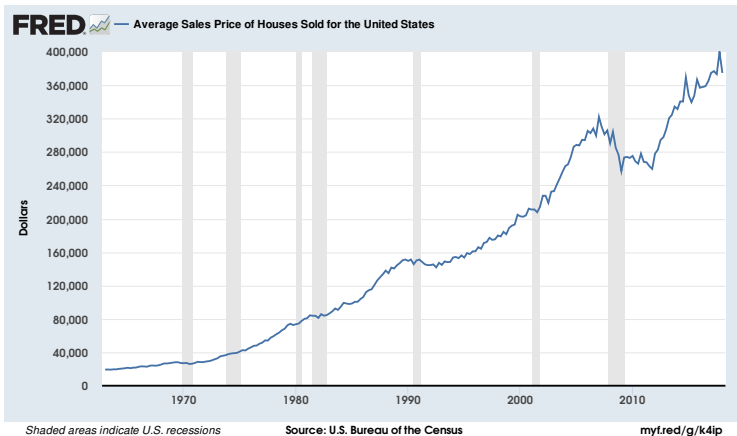
The "cloud" patterns for both Rand and Grades indicate that these are not valid inputs to a pricing model.

Let's take a closer look at the measures of precision around the fitted lines you saw on the scatterplots.

Estimates and Their Precision

					95% Confidence Interval	
Variable	Coefficient	Standard Error	t-Statistic	$P > t$	Lower Bound	Upper Bound
Area	40.14	2.66	15.09	0.00	34.90	45.37
Grades	208.37	207.51	1.00	0.32	-199.89	616.63
Rand	43.08	218.67	0.20	0.84	-387.14	473.29

National House Prices

FRED — Average Sales Price of Houses Sold for the United States

Dollars

Shaded areas indicate U.S. recessions   **Source: U.S. Bureau of the Census**   **myf.red/g/k4ip**

Predicting House Price

Say we wanted to predict the price of a house in our data. What should our prediction be?

Per FRED, the national average residential selling price between 1974 and 1992 was $95,645.

We could base our predictions on the national average, but our predictions would be more accurate if we only used the average for Massachusetts homes.

Predicting House Price (cont.)

But even using the Massachusetts average house price for our predictions would be fairly inaccurate.

This approach would predict the same price for every house, with the estimate derived from all houses sold in the entire state, rather than focusing on the traits of any one particular house.

An alternative to using a state-wide average would be to stratify houses by neighborhoods and use the average within each neighborhood to predict prices.

However, this still ignores the individual characteristics of each house and is likely to perform poorly.

Intuition

If we were able to obtain estimates for how much a one-unit change in the various house characteristics (e.g., square feet or age) impacted prices, then we could use these estimates to build more accurate predictions.

The estimates for how much each house characteristic changes the house price, multiplied by the value of the characteristic for the particular house, gives the contribution for that particular characteristic to overall price.

The series of *price estimates × characteristics* is then summed up, and this sum is the predicted price for the house.

The above is exactly what Ordinary Least Squares (OLS) regression enables us to do.

Ordinary Least Squares (OLS) Regression

Mathematically, the regression equation for cross-sectional data is

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + ... + \hat{\beta}_j x_{ij} + u_i$$

y is the "dependent variable"

x_j are the "independent variables"

$\hat{\beta}_j$ are estimates of the marginal effect of each x_j on y

u_i is the part of the dependent variable that our model does not predict accurately.

Ordinary Least Squares (OLS) Regression (cont.)

Below is the intuition of the equation:

$$y_i = \hat{\beta_0} + \hat{\beta_1} x_{i1} + \hat{\beta_2} x_{i2} + ... + \hat{\beta_j} x_{ij} + u_i$$

y_i is the house price

x_j are the features of the house we are using to predict the house price (e.g., number of bedrooms, age of the house, size of the house, etc.)

$\hat{\beta_j}$ are estimates of how much each feature changes the price of the house (e.g., \$76, \$32,721, etc.)

u_i is difference between the actual house price and what the predicted selling price was.

Ordinary Least Squares (OLS) Regression (cont.)

This model estimates average changes in y per one-unit changes in each x_j.

And this allows us to make predictions for each y_i based on the associated x_j values.

Question 1: What are the values of β_j that result in the best predictions?

Question 2: And how do we go about finding these values?

Question 1

The optimal values of β_j are the ones that "minimize the sum of squared residuals."

Step 1: Make predictions from the model.

Step 2: Calculate the difference between the actual values and each prediction.

Step 3: Square each difference.

Step 4: Sum up the total of the squared differences.

The optimal values of β_j minimize the sum in Step 4.

Question 1 (cont.)

For illustrative purposes, the rest of this example will limit predicting house prices using only one characteristic of the house.

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + u_i$$

This model has one independent variable and a constant term.

The independent variable we will use to predict house price is the square footage of the house, which is named "area" in our dataset.

And we are still looking for the "optimal" values of $\hat{\beta}_0$ and $\hat{\beta}_1$ which will result in the most accurate predictions.

Question 2

In a model with one independent variable and a constant, the following OLS equations result in minimization of the sum of squared residuals.

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

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

where the sum of squared residuals formula is

$$\sum_{i=1}^n \hat{u}_i^2 = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

A machine learning approach can also be taken to solve for $\hat{\beta}_0$ and $\hat{\beta}_1$, but let's first look at the output from the OLS regression.

OLS Regression of Price on Area

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.645339272
R Square	0.416462776
Adjusted R Square	0.414633506
Standard Error	33,070.15
Observations	321

ANOVA

	df	SS	MS	F	Significance F
Regression	1	248,983,535,162.23	248,983,535,162.23	227.67	0.00
Residual	319	348,869,500,735.76	1,093,634,798.54		
Total	320	597,853,035,897.99			

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	11,541.532	5,900.313	1.956	0.051	-66.910	23,149.975
X Variable 1	40.138	2.660	15.089	0.000	34.904	45.371

OLS Regression of Price on Area (cont.)

$\hat{\beta}_0$ is named "Intercept" in the previous output, and the value is $11,541.53.

$\hat{\beta}_1$ is named "X Variable 1" in the previous output, and the value is $40.14.

The interpretation is that our house price predictions start at $11,541.53 and each square foot adds around forty dollars to the overall house price.

Predicting house prices with this model begins with adding $11,541.53 to $40.14 × *area*.

For example, the prediction for a house with 1,500 square feet is:

$$\$11,541.53 + (\$40.14 \times 1,500) = \$71,751.53.$$

OLS Regression of Price on Area (cont.)

Consider all of the possible values that could have been picked for $\hat{\beta}_0$ and $\hat{\beta}_1$

Now imagine that we went through all of these combinations of guesses for each parameter, made our predictions, and calculated the sum of squares residual function.

The smaller values of this function are related to more accurate predictions.

The function values traced our from the parameter combinations will be convex and shaped like a parabola or a bowl.

And the OLS estimates of \$11,541.53 and \$40.14 resulted in the smallest values of the function, which means they give the most accurate predictions, and are at the bottom of the bowl.

OLS in Excel

OLS in Excel (cont.)

OLS in Excel (cont.)

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.645339272
R Square	0.416462776
Adjusted R Square	0.414633506
Standard Error	33,070.15
Observations	321

ANOVA

	df	SS	MS	F	Significance F
Regression	1	248,983,535,162.23	248,983,535,162.23	227.67	0.00
Residual	319	348,869,500,735.76	1,093,634,798.54		
Total	320	597,853,035,897.99			

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	11,541.532	5,900.313	1.956	0.051	-66.910	23,149.975
X Variable 1	40.138	2.660	15.089	0.000	34.904	45.371

Finding the Minimum

Consider the following quadratic equation
$y = x^2 + 4x - 2$

The first derivative of this equation is

$$\frac{\partial}{\partial x} = x^2 + 4x - 2$$
$$\frac{\partial}{\partial x} = 2x + 4$$

Next, we set the first derivative equal to zero and solve for the minimum

$$2x + 4 = 0$$
$$2x = -4$$
$$x = -2$$

Let's look at a graph of this function to see how the math lines up with what we can see.

Finding the Minimum (cont.)
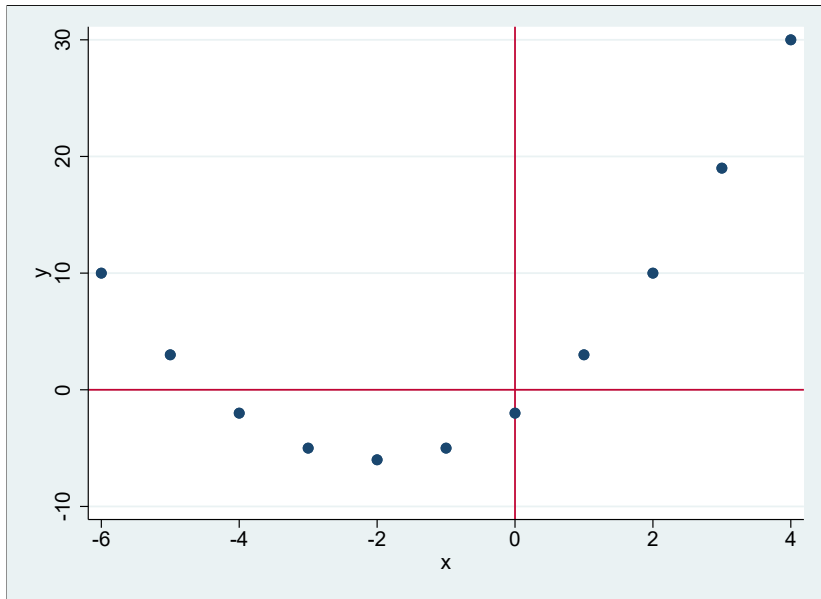
Finding the Minimum (cont.)

We can see that setting the first derivative of the function equal to zero and solving for *x* yielded the value of *x* where the parabola was at its minimum.

Not every function can be solved quite so easily, though.

Think of the machine learning algorithm starting somewhere on the curve of the parabola and updating the values of the parameters, which in turn updates the predictions, until it has reached the parabola's minimum.

The algorithm takes a series of steps to "walk" down to the minimum of the function before settling on its final predictions.

Machine Learning

Broadly, machine learning algorithms:

1: Define a "cost function" which is an equation with the difference between the predictions and actual values.

2: Use the first and second derivatives of the function with respect to the parameters to solve for the values of the parameters that result in the smallest value of the function by:

3: Making initial guesses for the parameters of the function.

4: Then adjusting the subsequent guesses of the parameters to descend to the minimum of the function.

Machine Learning (cont.)

The sum of squares residual function is similar to the parabola we graphed for the quadratic equation because both are convex.

However, taking the first derivative is more complex mathematically, and an iterative method is used for the solutions.

The derivatives of the sum of squares residual function with respect to $\hat{\beta}_0$ and $\hat{\beta}_1$ are

$$\frac{\partial}{\partial \hat{\beta}_0} = -2 \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)(1)$$

$$\frac{\partial}{\partial \hat{\beta}_1} = -2 \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)(x_i)$$

And the iterative method to find the solution is called "Gradient Descent."

Gradient Descent

Gradient descent iterates via the following formula:

$$\hat{\beta}_j = \hat{\beta}_j + \frac{\alpha}{2n} \frac{\partial}{\partial \hat{\beta}_j} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

where an initial guess is used for each $\hat{\beta}_j$

α is the "learning parameter" which is a scalar that controls the size of the updates to the parameter vector after each iteration.

The values of the $\hat{\beta}_j$ are updated after each iteration as the algorithm moves down to the minimum of the sum of squares residual function.

Gradient Descent (cont.)

Using linear algebra, the entire $\hat{\beta}_j$ parameter vector is updated simultaneously.

A vector of ones is added to the X matrix for estimation of $\hat{\beta}_0$ along with the other parameters.

Some texts flip the order of the predicted value and the observed value.

If this formula is used, the update is subtracted from $\hat{\beta}_j$ after each iteration rather than being added.

You will actually see the gradient descent steps "walking" to the minimum in a few slides.

Data Preparation

Machine learning algorithms can have difficulty taking the right "steps" to descend if the independent variables are on different scales.

For example, number of bedrooms in a house could range from one to 10, but we saw square feet (area) that ranged from around 1,000 to over 5,000.

One way to deal with this problem is to *standardize* variables by subtracting the mean of the variable from all observations and then dividing by the standard deviation.

This converts variables into *z scores* where

$$z = (X_i - \mu)/\sigma$$

and *z* now has a mean of zero and a variance of 1.

Data Preparation in Excel

Data Preparation in Excel (cont.)

▲	A	B	C
1		**Data**	**z Score**
2		10	=STANDARDIZE(B2,B8, B9)
3		7	=STANDARDIZE(B3,B8, B9)
4		9	=STANDARDIZE(B4,B8, B9)
5		11	=STANDARDIZE(B5,B8, B9)
6			
7			
8	**Mean**	=AVERAGE(B2:B5)	
9	**Standard Deviation**	=STDEV.S(B2:B5)	
10			
11			
12			
13			

Subset for Illustrative Purposes

When we originally estimated the relationship between house price and area, we used all 321 observations in the dataset.

However, to illustrate gradient descent manually in Excel we are going to only use 25 observations going forward.

This will make the example easier to follow.

Area is no longer statistically significant at conventional levels using the 25 observations, but don't be alarmed by this. That doesn't matter for the illustration of gradient descent.

Linking the Formula to Excel

The gradient descent formula is:
$$\hat{\beta}_j = \hat{\beta}_j + \frac{\alpha}{2n} \frac{\partial}{\partial \hat{\beta}_j} \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

Our initial guesses for $\hat{\beta}_j$ are:

$\hat{\beta}_0 = 0$

$\hat{\beta}_1 = 0$

The learning rate parameter is $\frac{\alpha}{2n} = \frac{.02}{2 \times 25} = .0004$

The derivatives $\frac{\partial}{\partial \hat{\beta}_j} \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$ are

(Y-XB)X0, where X0 is equal to 1

(Y-XB)X1, where X1 is equal to scaled

Where (Y-XB) is the difference between the actual price for the house and our predicted house price, and then this difference is being multiplied by either 1 or the value of scaled.

Gradient Descent in Excel Definitions

Price: House price.

Area: House square footage.

Scaled: Z score of area.

B0: Value of $\hat{\beta}_0$ to predict house prices.

B1: Value of $\hat{\beta}_1$ to predict house prices.

XB: $\hat{\beta}_0 + \hat{\beta}_1 \times$ scaled

(Y-XB)X0: First derivative of the cost function with respect to $\hat{\beta}_0$

(Y-XB)X1: First derivative of the cost function with respect to $\hat{\beta}_1$

U_sq: (Actual house price - Predicted house price)^2

Gradient Descent in Excel- First Guess

Price	Area	Scaled	B0	B1	XB	(Y-XB)X0	(Y-XB)X1	U_Sq
60,000	1,660	0.106	0	0	0	60,000	6,366	3,600,000,000
40,000	2,612	2.191	0	0	0	40,000	87,649	1,600,000,000
34,000	1,144	-1.024	0	0	0	34,000	-34,819	1,156,000,000
63,900	1,136	-1.042	0	0	0	63,900	-66,558	4,083,210,000
44,000	1,868	0.562	0	0	0	44,000	24,714	1,936,000,000
46,000	1,780	0.369	0	0	0	46,000	16,971	2,116,000,000
56,000	1,700	0.194	0	0	0	56,000	10,848	3,136,000,000
38,500	1,556	-0.122	0	0	0	38,500	-4,685	1,482,250,000
60,500	1,642	0.067	0	0	0	60,500	4,034	3,660,250,000
55,000	1,443	-0.369	0	0	0	55,000	-20,305	3,025,000,000
39,000	1,439	-0.378	0	0	0	39,000	-14,740	1,521,000,000
41,000	1,482	-0.284	0	0	0	41,000	-11,635	1,681,000,000
50,900	1,290	-0.704	0	0	0	50,900	-35,849	2,590,810,000
52,000	1,274	-0.739	0	0	0	52,000	-38,446	2,704,000,000
49,000	1,476	-0.297	0	0	0	49,000	-14,549	2,401,000,000
80,000	1,838	0.496	0	0	0	80,000	39,677	6,400,000,000
50,000	1,536	-0.165	0	0	0	50,000	-8,275	2,500,000,000
59,000	2,458	1.854	0	0	0	59,000	109,382	3,481,000,000
42,000	750	-1.887	0	0	0	42,000	-79,256	1,764,000,000
71,500	2,106	1.083	0	0	0	71,500	77,431	5,112,250,000
43,000	1,000	-1.339	0	0	0	43,000	-57,597	1,849,000,000
48,000	1,410	-0.441	0	0	0	48,000	-21,191	2,304,000,000
37,500	2,346	1.609	0	0	0	37,500	60,323	1,406,250,000
59,000	1,215	-0.869	0	0	0	59,000	-51,246	3,481,000,000
59,000	2,128	1.131	0	0	0	59,000	66,737	3,481,000,000
						1,278,800	44,980	68,471,020,000
						512	18	13,694,204

Linking the Formula to Excel (cont.)

Since the initial guess we used for both parameters was 0, we predicted a price of 0 for all houses. This means that the difference between the actual house price and our predicted house price is equal to the house price at the first step.

The difference then multiplied by either 1 or the value of scaled for the derivatives, the sums are calculated, and then the learning rate parameter is multiplied to each sum.

The above then gives us the values that we change $\hat{\beta}_0$ and $\hat{\beta}_1$ by for the next iteration.

In our example, we will use $\hat{\beta}_0 = 512$ and $\hat{\beta}_1 = 18$ for the second iteration.

Gradient Descent in Excel- Second Guess

Price	Area	Scaled	B0	B1	XB	(Y-XB)X0	(Y-XB)X1	U_Sq
60,000	1,660	0.106	512	18	513	59,487	6,311	3,538,652,142
40,000	2,612	2.191	512	18	551	39,449	86,442	1,556,227,961
34,000	1,144	-1.024	512	18	493	33,507	-34,314	1,122,712,705
63,900	1,136	-1.042	512	18	493	63,407	-66,045	4,020,475,622
44,000	1,868	0.562	512	18	522	43,478	24,421	1,890,369,037
46,000	1,780	0.369	512	18	518	45,482	16,780	2,068,597,971
56,000	1,700	0.194	512	18	515	55,485	10,748	3,078,584,649
38,500	1,556	-0.122	512	18	509	37,991	-4,623	1,443,290,967
60,500	1,642	0.067	512	18	513	59,987	3,999	3,598,473,814
55,000	1,443	-0.369	512	18	505	54,495	-20,119	2,969,718,378
39,000	1,439	-0.378	512	18	505	38,495	-14,549	1,481,886,593
41,000	1,482	-0.284	512	18	506	40,494	-11,491	1,639,730,477
50,900	1,290	-0.704	512	18	499	50,401	-35,498	2,540,276,107
52,000	1,274	-0.739	512	18	498	51,502	-38,078	2,652,433,587
49,000	1,476	-0.297	512	18	506	48,494	-14,398	2,351,650,779
80,000	1,838	0.496	512	18	520	79,480	39,419	6,316,999,915
50,000	1,536	-0.165	512	18	509	49,491	-8,191	2,449,404,378
59,000	2,458	1.854	512	18	545	58,455	108,371	3,417,001,521
42,000	750	-1.887	512	18	478	41,522	-78,355	1,724,112,346
71,500	2,106	1.083	512	18	531	70,969	76,856	5,036,598,308
43,000	1,000	-1.339	512	18	487	42,513	-56,945	1,807,319,458
48,000	1,410	-0.441	512	18	504	47,496	-20,968	2,255,910,194
37,500	2,346	1.609	512	18	540	36,960	59,454	1,366,007,421
59,000	1,215	-0.869	512	18	496	58,504	-50,815	3,422,730,583
59,000	2,128	1.131	512	18	532	58,468	66,136	3,418,522,043
						1,266,012	44,548	67,167,686,957
						506	18	13,433,537

Linking the Formula to Excel (cont.)

After using values of $\hat{\beta}_0 = 512$ and $\hat{\beta}_1 = 18$ in the second iteration, the next "step" the algorithm will take will add 506 to $\hat{\beta}_0$ and 18 to $\hat{\beta}_1$.

You can also see the cost function is decreasing at each step.

This is because the newer estimates of $\hat{\beta}_0$ and $\hat{\beta}_1$ result in more accurate predictions.

Gradient Descent in Excel- Third Guess

Price	Area	Scaled	B0	B1	XB	(Y-XB)X0	(Y-XB)X1	U_Sq
60,000	1,660	0.106	1,018	36	1,022	58,978	6,257	3,478,437,009
40,000	2,612	2.191	1,018	36	1,096	38,904	85,247	1,513,490,419
34,000	1,144	-1.024	1,018	36	981	33,019	-33,814	1,090,237,784
63,900	1,136	-1.042	1,018	36	981	62,919	-65,537	3,958,847,934
44,000	1,868	0.562	1,018	36	1,038	42,962	24,130	1,845,730,088
46,000	1,780	0.369	1,018	36	1,031	44,969	16,590	2,022,198,672
56,000	1,700	0.194	1,018	36	1,025	54,975	10,649	3,022,265,828
38,500	1,556	-0.122	1,018	36	1,014	37,486	-4,562	1,405,232,669
60,500	1,642	0.067	1,018	36	1,020	59,480	3,966	3,537,833,236
55,000	1,443	-0.369	1,018	36	1,005	53,995	-19,935	2,915,492,039
39,000	1,439	-0.378	1,018	36	1,004	37,996	-14,360	1,443,666,395
41,000	1,482	-0.284	1,018	36	1,008	39,992	-11,349	1,599,379,055
50,900	1,290	-0.704	1,018	36	993	49,907	-35,150	2,490,738,318
52,000	1,274	-0.739	1,018	36	991	51,009	-37,713	2,601,872,400
49,000	1,476	-0.297	1,018	36	1,007	47,993	-14,250	2,303,300,026
80,000	1,838	0.496	1,018	36	1,036	78,964	39,163	6,235,362,892
50,000	1,536	-0.165	1,018	36	1,012	48,988	-8,107	2,399,824,326
59,000	2,458	1.854	1,018	36	1,084	57,916	107,371	3,354,226,385
42,000	750	-1.887	1,018	36	950	41,050	-77,462	1,685,074,007
71,500	2,106	1.083	1,018	36	1,057	70,443	76,287	4,962,257,537
43,000	1,000	-1.339	1,018	36	970	42,030	-56,298	1,766,524,584
48,000	1,410	-0.441	1,018	36	1,002	46,998	-20,748	2,208,801,181
37,500	2,346	1.609	1,018	36	1,076	36,424	58,593	1,326,741,891
59,000	1,215	-0.869	1,018	36	987	58,013	-50,389	3,365,529,051
59,000	2,128	1.131	1,018	36	1,058	57,942	65,540	3,357,225,239
						1,253,352	44,121	65,890,288,965
						501	18	13,178,058

Linking the Formula to Excel (cont.)

The series of steps continues until the algorithm "converges" to the optimal values of $\hat{\beta}_0$ and $\hat{\beta}_1$.

Convergence occurs because the derivatives will be equal to 0 at the minimum of the cost function.

This means that the guesses the algorithm makes settle and will no longer materially change as more iterations are run.

Let's look at the first three and last three iterations of the algorithm to see it settling on the optimal values.

Iterations

You can see below that there are large changes in the parameter estimates initially followed by immaterial changes near the point of convergence.

The values of $\hat{\beta}_0$ and $\hat{\beta}_1$ that result in the best predictions are 51,149 and 1,874.

These values are very different from the values of 0 that the algorithm stated with.

iteration	beta_0	beta_1	cost
1	0.00	0.00	13,694,204.00
2	511.52	17.99	13,433,537.39
3	1,017.92	35.81	13,178,057.80
998	51,149.70	1,874.03	594,708.32
999	51,149.73	1,874.03	594,708.32
1000	51,149.75	1,874.03	594,708.32

Linking the Formula to Excel (cont.)

Let's look at a graph of the cost function for various combinations of values for $\hat{\beta}_0$ and $\hat{\beta}_1$

Cost Function Plane
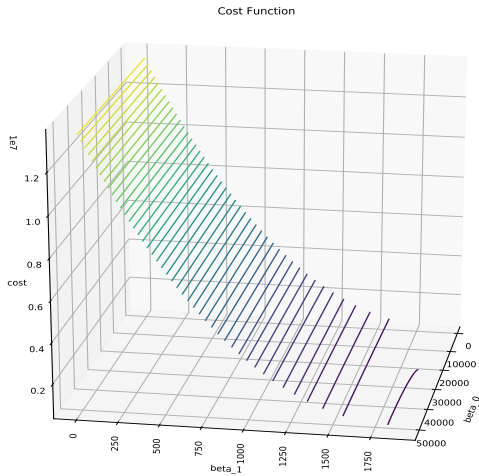
Cost Function Plane (cont.)

The initial guesses we used of 0 for $\hat{\beta}_0$ and $\hat{\beta}_1$ put us in the upper right hand portion of the cost function.

Each iteration of gradient descent results in use descending the cost function plane to arrive at the minimum.

The series of steps the algorithm took is shown on the next slide.

Gradient Descent Path

Descent Path

Gradient Descent Results vs. OLS

Lastly, let's see how close the gradient descent estimates for $\hat{\beta}_0 = 51,149$ and $\hat{\beta}_1 = 1,874$ are to the OLS results.

The estimate of $\hat{\beta}_0$ differs by \$3 and the estimate of $\hat{\beta}_1$ is nearly identical.

```
      Source |       SS           df       MS      Number of obs   =        25
-------------+----------------------------------   F(1, 23)        =      0.65
       Model |  84300940.9         1  84300940.9   Prob > F        =     0.4277
    Residual |  2.9735e+09        23   129284411   R-squared       =     0.0276
-------------+----------------------------------   Adj R-squared   =    -0.0147
       Total |  3.0578e+09        24   127410100   Root MSE        =      11370

-------------+----------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      scaled |   1874.177   2320.959     0.81   0.428    -2927.092    6675.446
       _cons |      51152   2274.066    22.49   0.000     46447.74    55856.26
------------------------------------------------------------------------------
```

Conclusion

Supervised machine learning models may be used to make predictions or understand associations, correlations, and relationships.

Analysts need to understand their data by looking at summary statistics, graphs, histograms, etc.

Building an accurate model involves talking to professionals in the area, considering theories, and conducting exploratory data analyses.

The appropriate technique or algorithm is then chosen.

Conclusion (cont.)

Machine learning algorithms express their accuracy in the form of a mathematical function.

Once the function has been specified, derivatives from calculus are used to find the values of the input parameters that result in the "optimal" value of the function.

For the cost function we used today, smaller values of the cost function were better. This is because the cost function was larger for less accurate predictions.

And today you have seen a start-to-finish, detailed example of how a machine learning algorithm makes predictions.

Thank you!

For Further Information

Robson Glasscock, PhD, CPA

Robson@xbanalyticsllc.com

Cell (303) 204-6024

Attributions

https://jakevdp.github.io/PythonDataScienceHandbook/04.12-three-dimensional-plotting.html

https://pythonprogramming.net/3d-graphing-pandas-matplotlib/

https://stackoverflow.com/questions/36589521/how-to-surface-plot-3d-plot-from-dataframe

https://www.python-course.eu/matplotlib_contour_plot.php

http://www.adeveloperdiary.com/data-science

https://www.coursera.org/learn/machine-learning