Model Selection and Evaluation

Delta Analytics builds technical capacity around the world.

This course content is being actively developed by Delta Analytics, a 501(c)3 Bay Area nonprofit that aims to empower communities to leverage their data for good.

Please reach out with any questions or feedback to inquiry@deltanalytics.org.

Find out more about our mission here.

Module 4: Model Selection and Evaluation

In the last module, we went through the nuts and bolts of linear regressions. In this module, we will step back and explain how to **select**, **evaluate**, and **validate** a model.

*This is a very important module: as a researcher, you **must** consider which model to use and why, and whether or not its results are significant.*

Let's get started!

Model evaluation:
Let’s recap the performance measures in module 3.

Performance Measure

Measuring performance is very different depending on whether our output is continuous or categorical. For example, in classification tasks accuracy is commonly used.

Classification task: Does this patient have malaria?

Temperature of patient	Malaria test result	Predicted test result
X	Y	Y*
39.5°C	Yes	Yes
37.8°C	No	Yes
37.2°C	No	No
37.2°C	No	No

Accuracy is a common performance metric for category prediction tasks. It is simply the number of correct predictions over the total number of predictions.

Accuracy = number of cases the correct class is predicted (Y*=Y)/number of total cases

Performance Measure

If you are predicting a numerical value, there are a few different performance measures we can use. Our choice depends upon our model architecture.

Regression task: How does number of malaria cases change for every additional mosquito net bought.

mosquito nets	malaria cases	predicted malaria cases
X	Y	Y*
2007: 1000	80	58
2008: 2200	40	42
2009: 6600	42	38
2010: 12600	35	30

Mean squared error (MSE) is the average squared difference between the true Y and the predicted Y*.

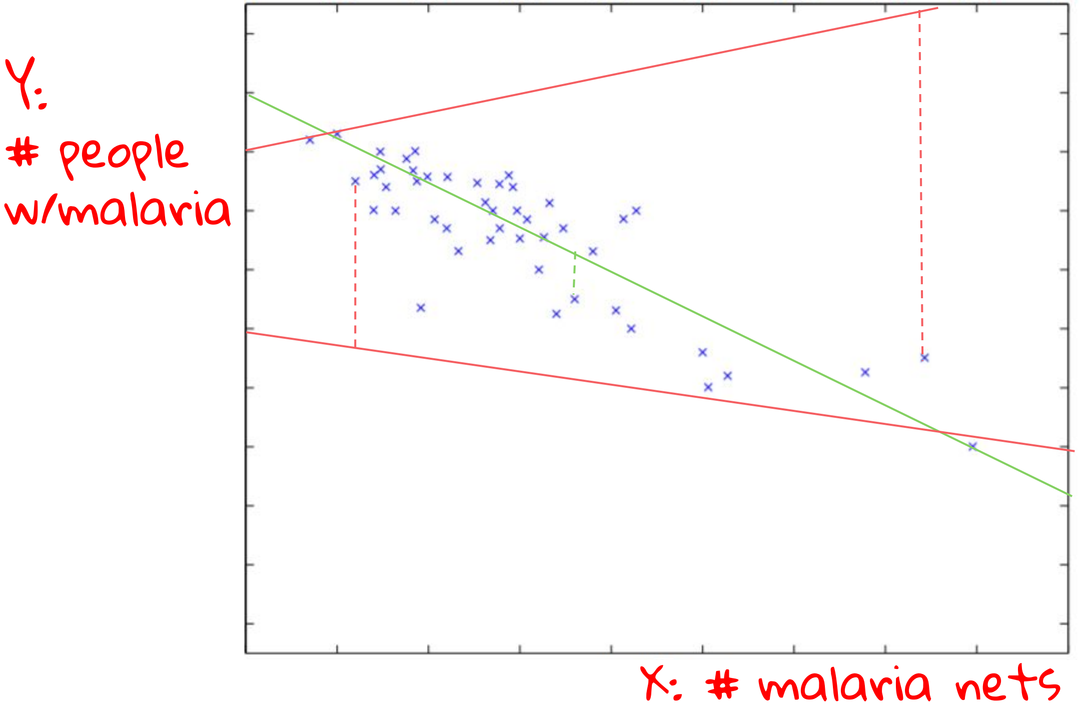
- MSE is always>0 (irreducible error)
- Smaller (closer to 0) is better.

Performance Measure

MSE is a way to quantify distance between Y* and Y.

Visually, we can tell our MSE is much higher for the red lines than it is for the green line (a good fit!)

In the last module, we formalized and quantified.



Performance Measure

We will evaluate our model in a few different ways.

Alongside evaluating purely based upon how well our predicted Y* approximates the true Y, we will evaluate our model along other dimensions:

1) How well does the model perform on **unseen data**?
2) Are there holistic objectives that model choice based upon accuracy might hurt?
3) Is getting one class wrong more costly than another? (specific to classification problems)

Performance Measure

1) Unseen data: Recall that measuring how well the model performs on unseen data is extremely important!

We already know that how your model performs on unseen data is the **most important metric** we will evaluate our models by.

If we have high predictive power on our training data but low predictive power in the real world, what use is our model?

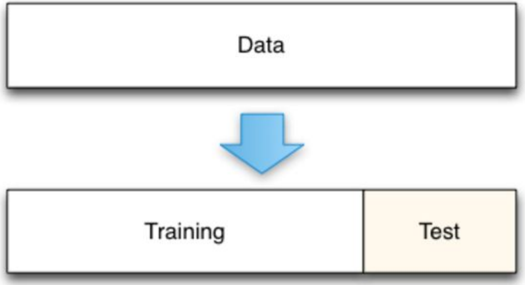
Our model is naturally highly optimized for the data it was trained on. However, the ultimate impact of our model depends upon its ability to ***generalize ("extrapolate")*** to unseen data.

In this module, we will go over what this means using real-life examples of model evaluation on Kiva data.

Performance Measure

1) Unseen data: Recall that measuring how well the model performs on unseen data is extremely important!

Remember our test/training split from Module 3?



Our "test" data is our unseen data. Here we introduce the concept of **generalization error,** our model's capability to generalize to unseen data.

Generalization error = Test True Y - TestY*

Performance Measure

2) Are there holistic objectives that model choice based upon accuracy might hurt?

We have values as a society that are not captured by optimizing for accuracy.

Choosing a model to deploy based purely on predictive power may pose ethical problems. Some examples:

- A risk score algorithm that predicts whether someone will commit a future crime, used by US courts, **discriminated based upon race**. It was clear the model relied heavily on race as a feature. [Link]
- Carnegie Mellon found that women were less likely than men to be shown ads by Google for high paying jobs. [Link]
- Orbitz, a travel site, was showing Mac users higher priced travel services than PC users. [Link]

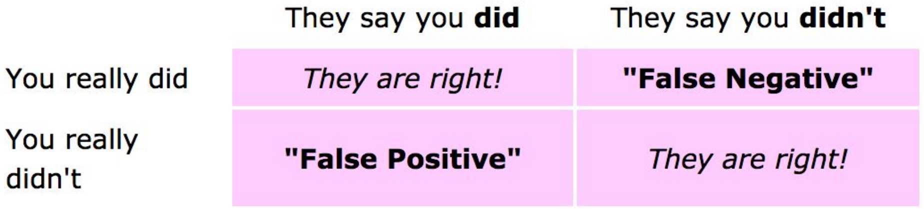
Performance Measure

3) Is getting one class wrong more costly than another? Particularly for fields like medicine, a false negative diagnosis is more costly than a false positive.

E.g. Classification task: Does Dr. Delta's patient have malaria?

False positive:

Patient is told they have malaria when they do not.



	They say you **did**	They say you **didn't**
You really did	*They are right!*	**"False Negative"**
You really didn't	**"False Positive"**	*They are right!*

False negative:

Patient is told they do not have malaria when they do.

In any classification model we work on, we will evaluate performance on FP, FN in addition to accuracy.

> Which is more costly?

Performance Measure

3) Is getting one class wrong more costly than another? Particularly for fields like medicine, a false negative diagnosis is more costly than a false positive.

E.g. Classification task: Does Dr. Delta's patient have malaria?

False positive:

Patient is told they have malaria when they do not.

False negative:

Patient is told they do not have malaria when they do.

Thinking about false positives and negatives is ***extremely*** important. In our example, if Dr. Delta's model incorrectly classifies a patient, it could have disastrous consequences.

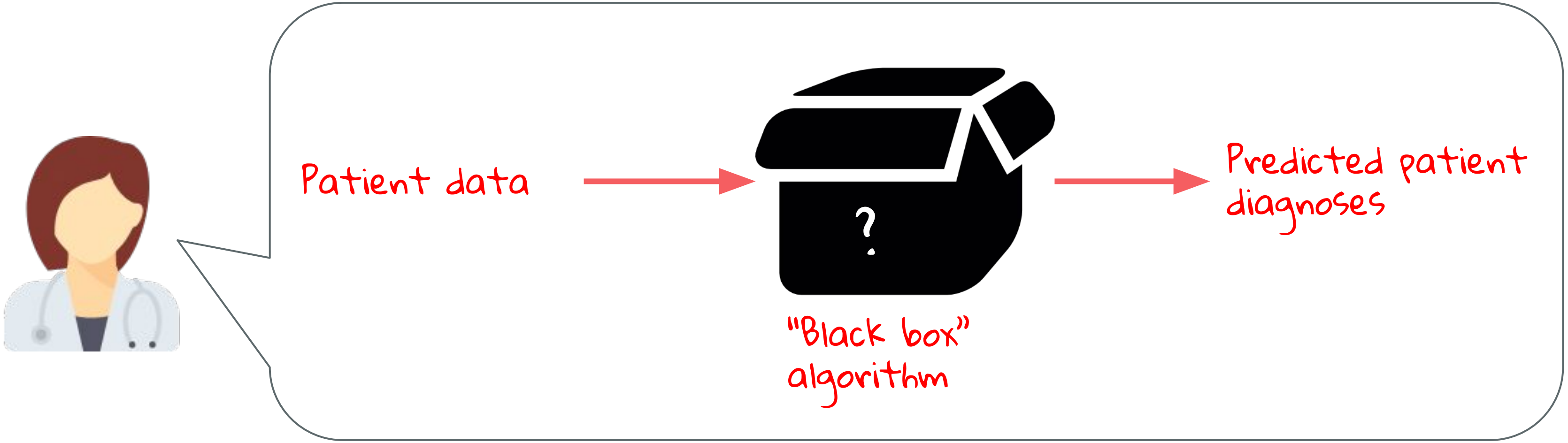
What should Dr. Delta do in this scenario?

Because the stakes are so high, Dr. Delta should not only use caution in selecting an algorithm, but also in **interpreting and presenting her results**.

The algorithm may say that a patient is 60% likely to have malaria, but Dr. Delta cannot use this result to conclude that the patient definitely has malaria. Rather, Dr. Delta should present the result as a starting point for additional research.

This is a key tenet of machine learning ethics: ***transparency***.

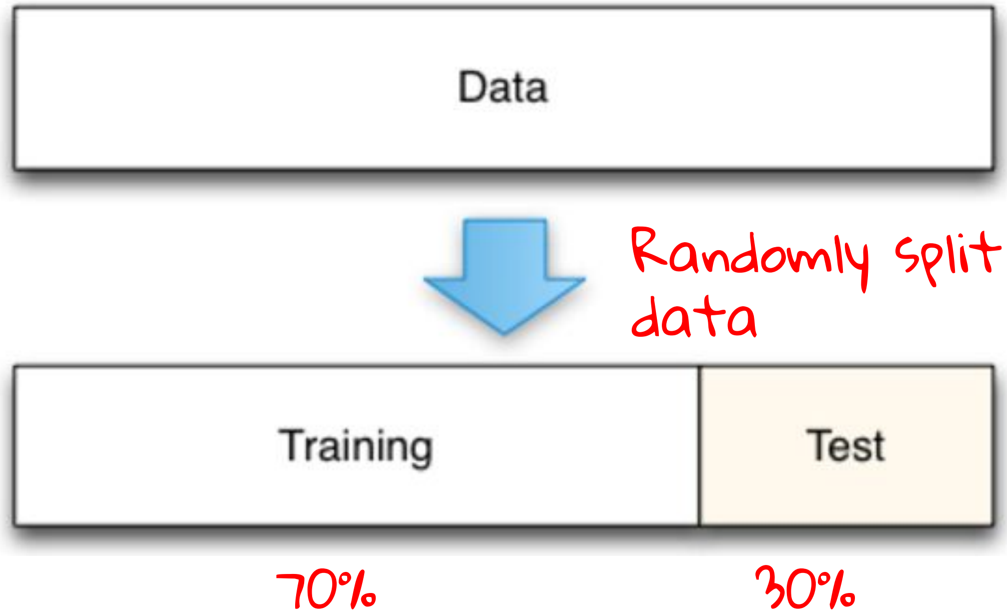
OLS Regression: Model development and evaluation



As a responsible machine learning practitioner, Dr. Delta should **explain and justify** the algorithm she used, as well as the **limits** of the conclusions she was able to draw.

Without transparency, there can't be good machine learning!

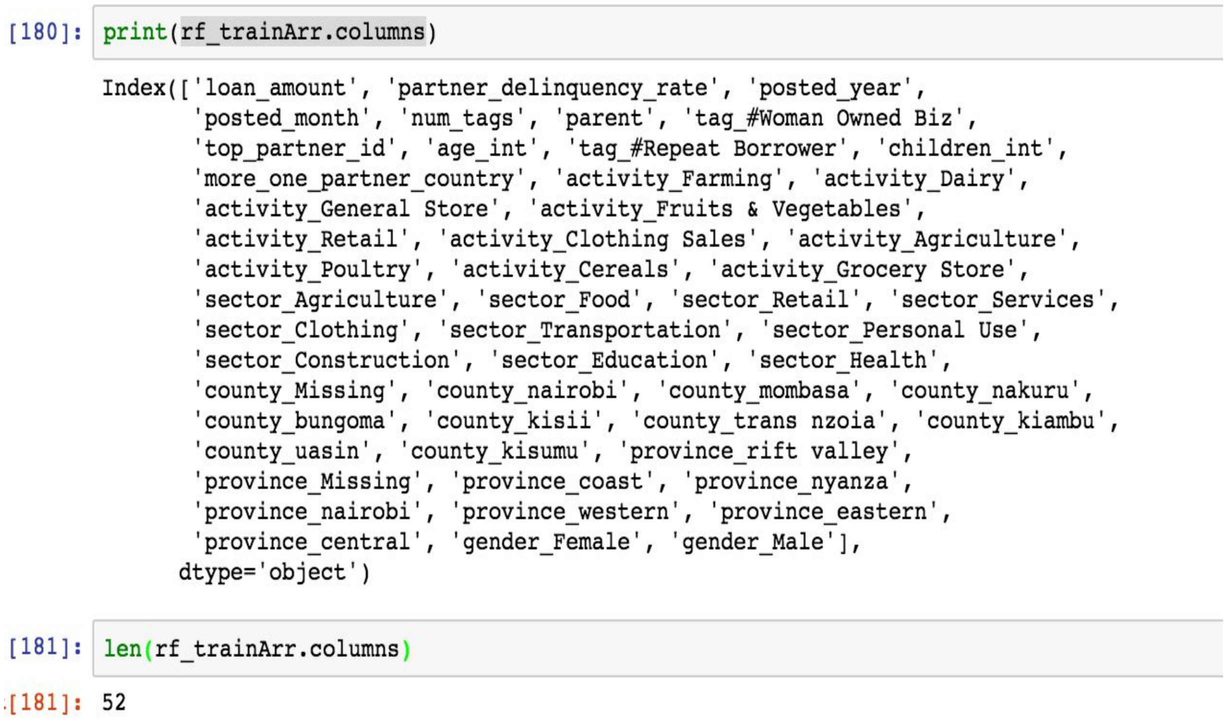
Our very first step for all models is creating a training and a test dataset.



Recall from the last module that we split our data into train/test sets so we can **create the model** using training data, and **test how well it generalizes to unseen data** by using the test data.

A common training/test split ratio is 70%/30%.

Using the training dataset, we make sure our OLS assumptions hold.

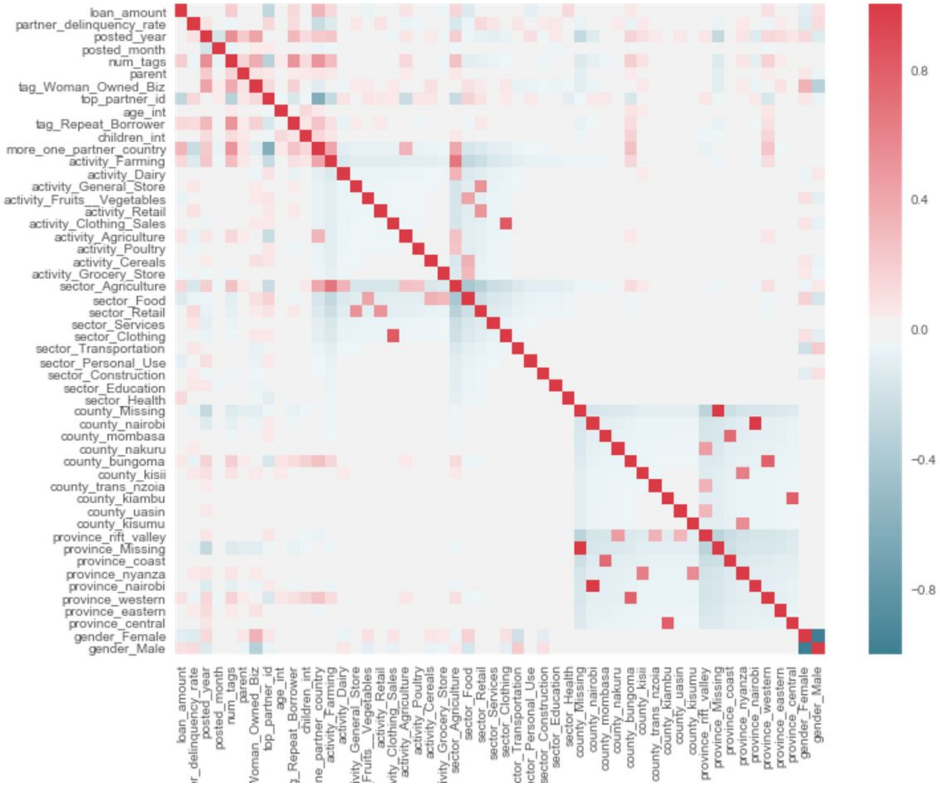


```
[180]: print(rf_trainArr.columns)

       Index(['loan_amount', 'partner_delinquency_rate', 'posted_year',
              'posted_month', 'num_tags', 'parent', 'tag_#Woman Owned Biz',
              'top_partner_id', 'age_int', 'tag_#Repeat Borrower', 'children_int',
              'more_one_partner_country', 'activity_Farming', 'activity_Dairy',
              'activity_General Store', 'activity_Fruits & Vegetables',
              'activity_Retail', 'activity_Clothing Sales', 'activity_Agriculture',
              'activity_Poultry', 'activity_Cereals', 'activity_Grocery Store',
              'sector_Agriculture', 'sector_Food', 'sector_Retail', 'sector_Services',
              'sector_Clothing', 'sector_Transportation', 'sector_Personal Use',
              'sector_Construction', 'sector_Education', 'sector_Health',
              'county_Missing', 'county_nairobi', 'county_mombasa', 'county_nakuru',
              'county_bungoma', 'county_kisii', 'county_trans nzoia', 'county_kiambu',
              'county_uasin', 'county_kisumu', 'province_rift valley',
              'province_Missing', 'province_coast', 'province_nyanza',
              'province_nairobi', 'province_western', 'province_eastern',
              'province_central', 'gender_Female', 'gender_Male'],
             dtype='object')

[181]: len(rf_trainArr.columns)

[181]: 52
```

To use the OLS algorithm, our assumptions must hold. Recall our discussion of assumptions from the previous module.

We start with 52 high potential features we want to try including in our model.

Checking assumptions: Correlation

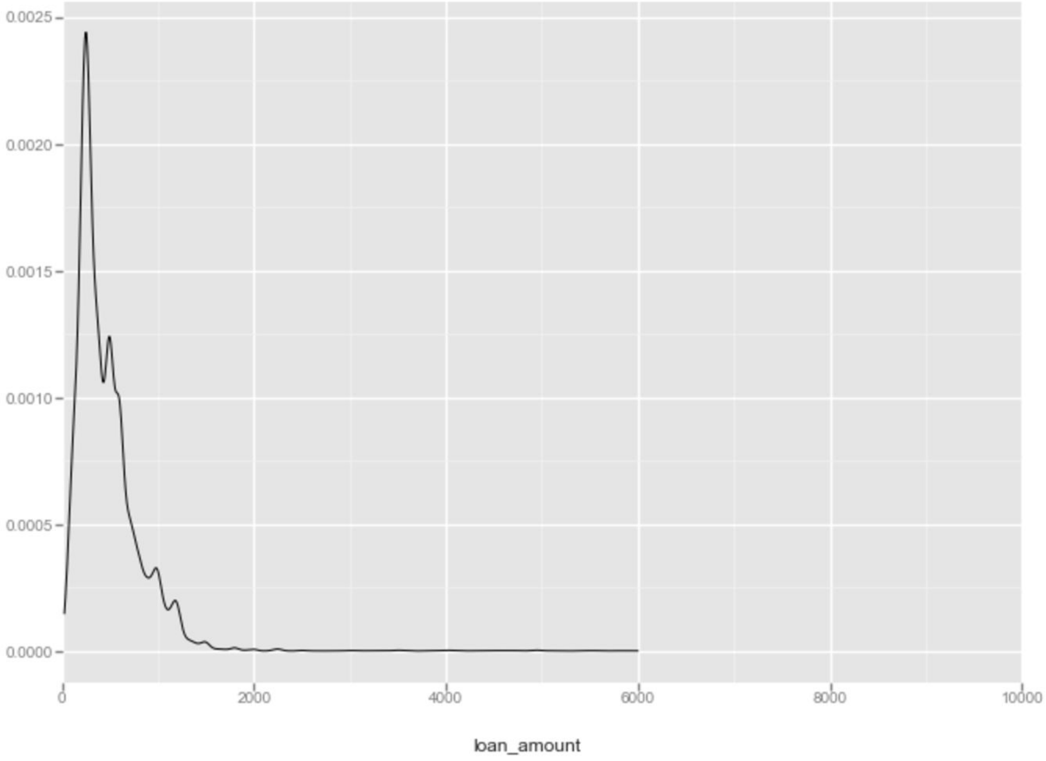


We want to exclude features that are heavily positive or negative correlated with each other.

The graphic is hard to see, but examples include:

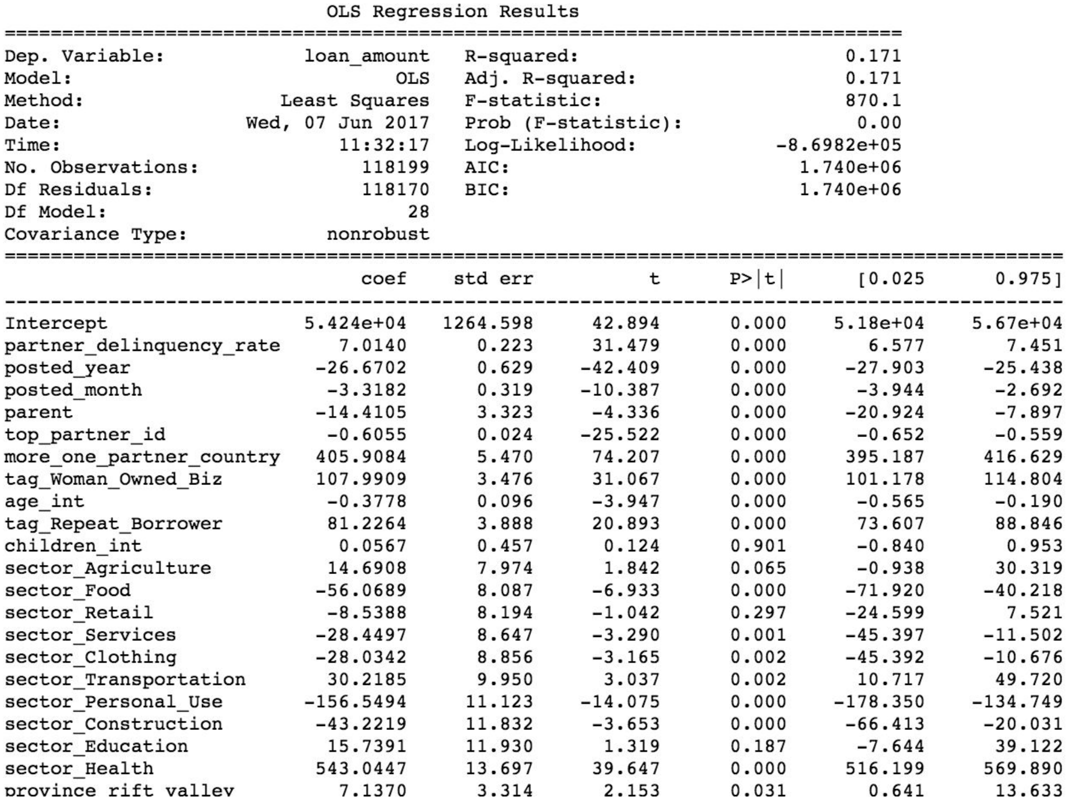
Activity & sector
Province & country

Checking assumptions: Distribution



Here we see that the feature loan_amount appears skewed to the left in this histogram, but it is not excessively skewed. *See discussion of skew in the Module 3 coding notebooks.*

We train our multivariate model using Python.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:            loan_amount   R-squared:                       0.171
Model:                            OLS   Adj. R-squared:                  0.171
Method:                 Least Squares   F-statistic:                     870.1
Date:                Wed, 07 Jun 2017   Prob (F-statistic):               0.00
Time:                        11:32:17   Log-Likelihood:            -8.6982e+05
No. Observations:              118199   AIC:                         1.740e+06
Df Residuals:                  118170   BIC:                         1.740e+06
Df Model:                          28
Covariance Type:            nonrobust
===========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------
Intercept               5.424e+04   1264.598     42.894      0.000    5.18e+04    5.67e+04
partner_delinquency_rate   7.0140      0.223     31.479      0.000       6.577       7.451
posted_year              -26.6702      0.629    -42.409      0.000     -27.903     -25.438
posted_month              -3.3182      0.319    -10.387      0.000      -3.944      -2.692
parent                   -14.4105      3.323     -4.336      0.000     -20.924      -7.897
top_partner_id            -0.6055      0.024    -25.522      0.000      -0.652      -0.559
more_one_partner_country 405.9084      5.470     74.207      0.000     395.187     416.629
tag_Woman_Owned_Biz      107.9909      3.476     31.067      0.000     101.178     114.804
age_int                   -0.3778      0.096     -3.947      0.000      -0.565      -0.190
tag_Repeat_Borrower       81.2264      3.888     20.893      0.000      73.607      88.846
children_int               0.0567      0.457      0.124      0.901      -0.840       0.953
sector_Agriculture        14.6908      7.974      1.842      0.065      -0.938      30.319
sector_Food              -56.0689      8.087     -6.933      0.000     -71.920     -40.218
sector_Retail             -8.5388      8.194     -1.042      0.297     -24.599       7.521
sector_Services          -28.4497      8.647     -3.290      0.001     -45.397     -11.502
sector_Clothing          -28.0342      8.856     -3.165      0.002     -45.392     -10.676
sector_Transportation     30.2185      9.950      3.037      0.002      10.717      49.720
sector_Personal_Use     -156.5494     11.123    -14.075      0.000    -178.350    -134.749
sector_Construction      -43.2219     11.832     -3.653      0.000     -66.413     -20.031
sector_Education          15.7391     11.930      1.319      0.187      -7.644      39.122
sector_Health            543.0447     13.697     39.647      0.000     516.199     569.890
province_rift_valley       7.1370      3.314      2.153      0.031       0.641      13.633
```

R2= 0.171

p-Score=0.000

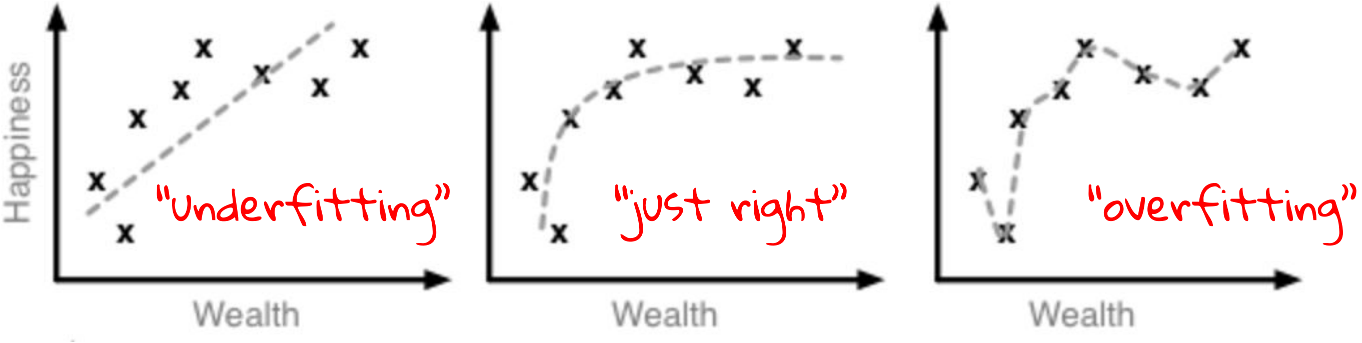
Now we have a model derived from our training data, let’s see how well the model generalizes to the real world (make sure it is not over or underfit).

To understand the concepts of underfitting and overfitting, let’s take a look at a more intuitive example.

Performance

Ability to generalize to unseen data

Recall our discussion of overfitting vs. underfitting in the last module.



Let’s take another look at underfitting and overfitting by considering a musical analogy ...

Performance

Ability to generalize to unseen data

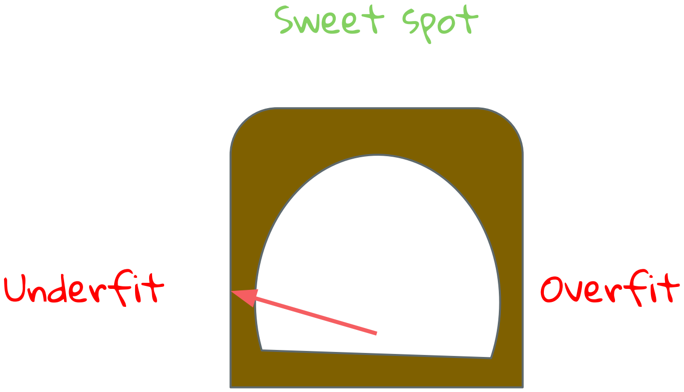
Your choice of audio recording equipment for a music concert is analogous to to underfitting/overfitting!

You love rock music and want to record your favorite band in concert.

However, you need to be careful about how you record.

Performance

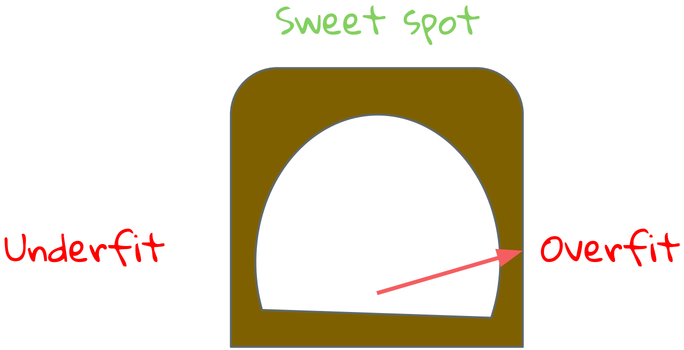
Underfitting occurs when error on our training set is still very high. Prediction performance on both seen and unseen data is low.



If you use a cellphone to record, you are probably underfitting. **You will not be able to capture the qualities of how the artist sounds live.**

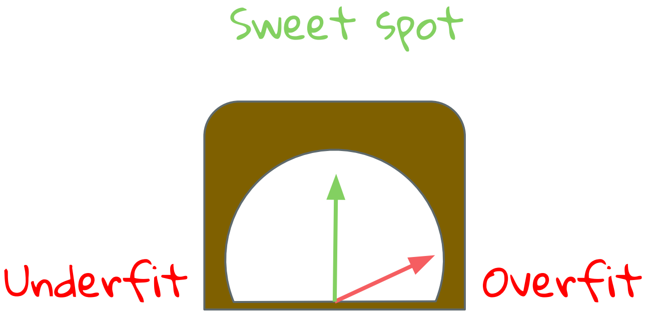
Performance

Overfitting occurs when we know our training data well but our model does not generalize to unseen data. Prediction performance on seen data is high and on unseen data is low.



If you go all out and bring sophisticated audio equipment, **you may catch all types of sounds that are not the concert,** like people shuffling in the seats and talking to each other.

In order to understand whether we are at our sweet spot we need to compare prediction performance on train and test datasets!



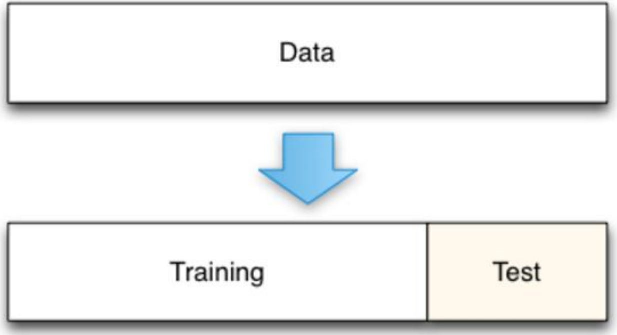
Training R2	Relationship	Test R2	Condition
high	>	low	Overfitting
high	~	high	Sweet spot
low	~	low	Underfitting
low	<	high	Shouldn't happen

It's unlikely that your model fits your test data better than your training data, but this can happen if you are working with small data or if you split your data in a non-random way

Performance

How do we measure underfitting/overfitting?

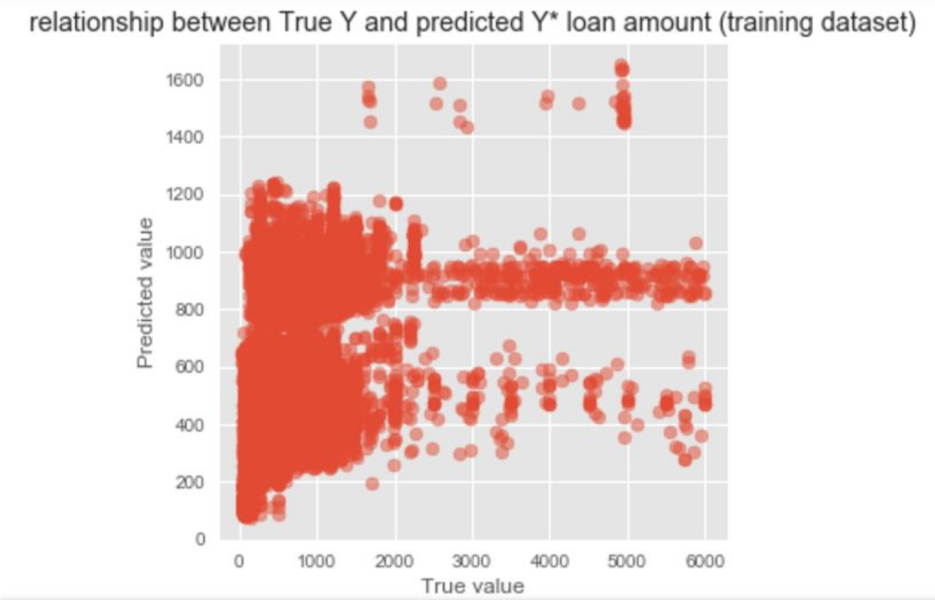
We apply the f(x) we got from the training data to score the test data.



1. Randomly split data into train/test
2. Run model on train data
3. Test model on test data
4. **Compare the model error (from train data) to the generalization error (from the test data)**

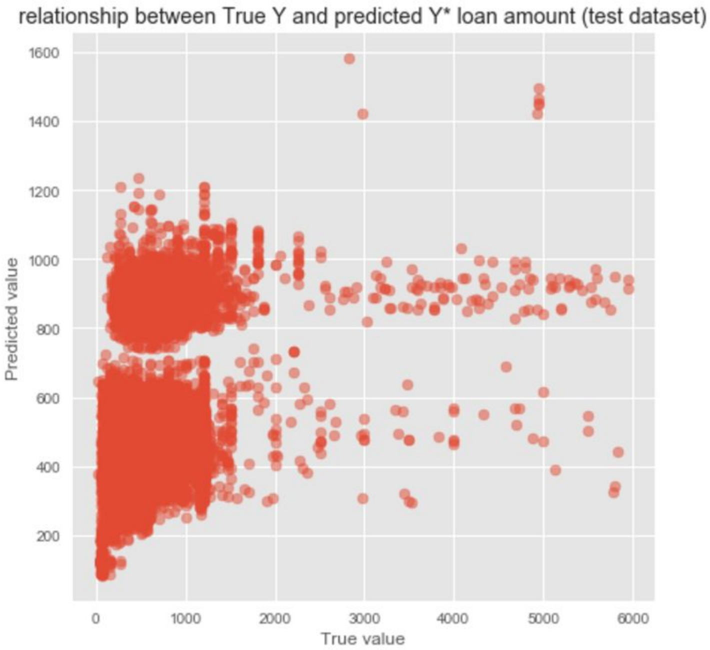
Comparing our model error on the training data to our generalization error on the test allows us to tell whether we have underfit or overfit on the data.

Training data



R2= 0.17

Test data



R2=0.16

Let's return to our linear regression. Is this model overfitting or underfitting?

Is our linear regression model under or over fitting?

Training Test

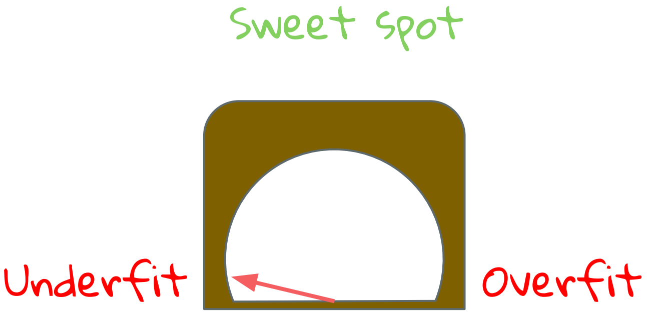
R2= 0.17 ~ R2=0.16

Training R2	Relationship	Test R2	Condition
high	>	low	Overfitting
high	~	high	Sweet spot
low	~	low	Underfitting
low	<	high	**Shouldn't happen**

Our linear model is underfitting the data. The R2 on both training and test are low.

Training R2= 0.17

Test R2=0.16



Training R2	Relationship	Test R2	Condition
high	>	low	Overfitting
high	~	high	Sweet spot
low	**~**	**low**	Underfitting
low	<	high	**Shouldn't happen**

What do you do when your model is underfitting?

Recall that in our rock concert example, underfitting occurs when you are not able to capture what is **truly happening** in reality.

A low adjusted R2 can mean a few different things:

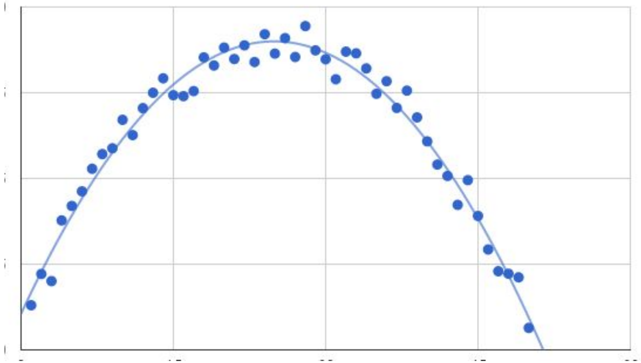
1) We need better feature engineering
2) We need more data
3) We need to do better feature selection (i.e., we are including too many useless features in our model)
4) We need a more complex model to capture the true f(x)

It is possible our model is underfitting because linear regression is **not complex enough** to capture the relationship that maps x to true Y.

We can test this by checking performance with a more complex model and seeing if the fit improves.

Some examples of models that are more complex than linear regression:

- Nonlinear regression
- Decision trees
- Bagging
- Boosting

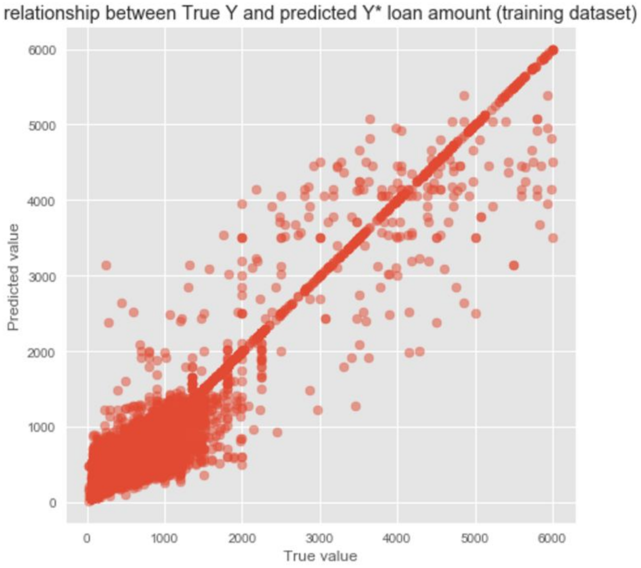


Recall our quadratic regression from Module 3.

We will go over the nuts and bolts of decision trees, bagging and boosting in future modules, but for now, let's consider how to interpret hypothetical R2 metrics to evaluate for overfitting and underfitting.

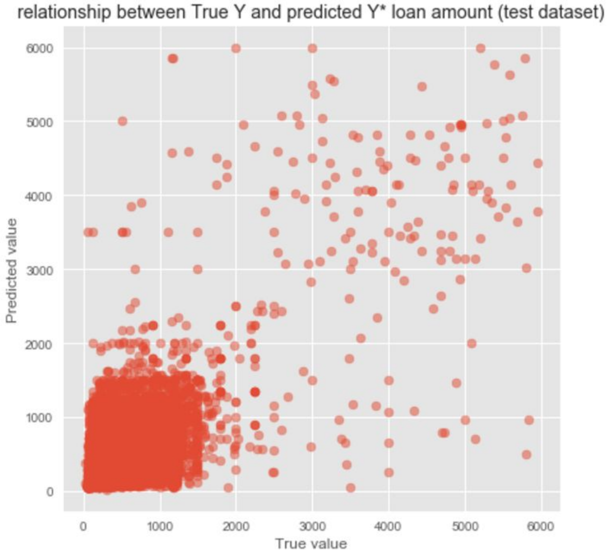
Results using Decision Tree Algorithm

Training data



R2= 0.93

Test data



R2=0.23

Is our decision tree model under or over fitting?

Training Test

R2= 0.93 ~R2=0.23

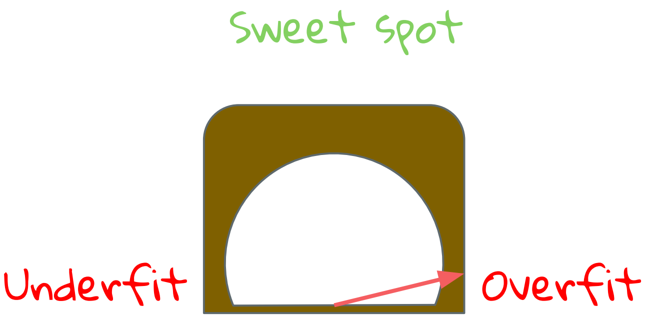
Training R2	Relationship	Test R2	Condition
high	>	low	Overfitting
high	~	high	Sweet spot
low	~	low	Underfitting
low	<	high	**Shouldn’t happen**

Our decision tree is **overfitting** the data. The R2 is very high on the training and low on test.

Training R2= 0.93

Test R2=0.23

Training R2	Relationship	Test R2	Condition
high	**>**	**low**	Overfitting
high	~	high	Sweet spot
low	~	low	Underfitting
low	<	high	**Shouldn't happen**



What do we do if our model is overfitted?

A large gap between the R^2 of the test and train datasets can be solved by a few different steps:

1) Get more data
2) Use ensemble approaches (like bagging)
3) Improve feature selection

Ensemble approaches are a powerful way to address overfitting!

We can all agree that more data is better.

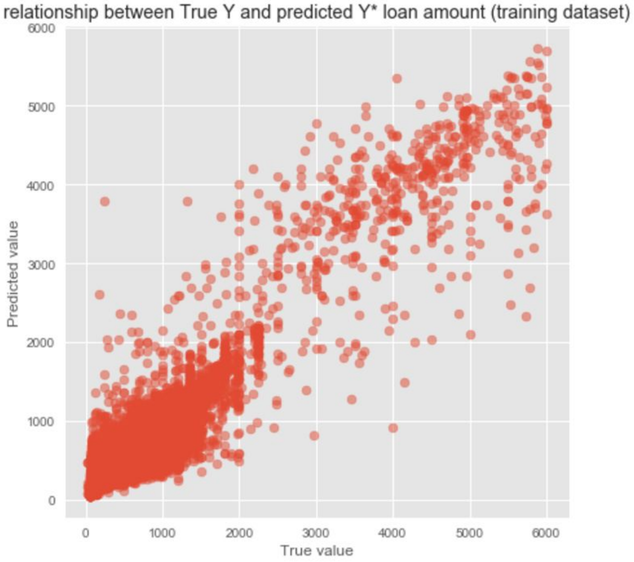
Imagine instead of running a single ML algorithm on a single dataset, you could run 100 different algorithms on 100 datasets! By leveraging “**the power of the crowd**,” bagging approaches can reduce generalization error.

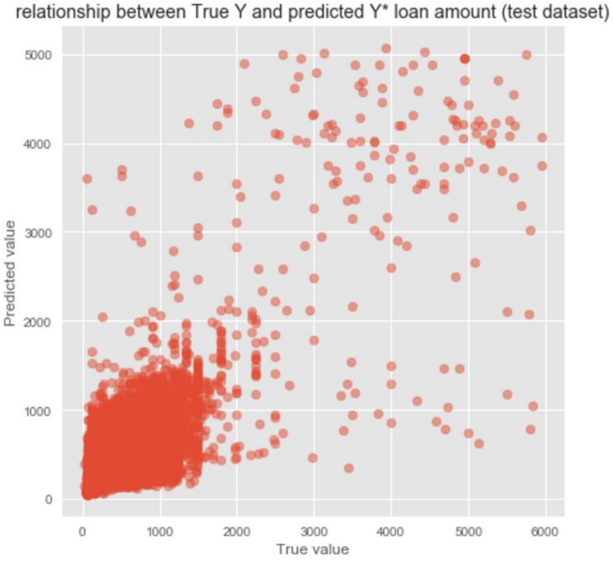
Bagging

We will do a deep dive into bagging regressions in the next lectures, but here is a teaser of how it improves results for our Kiva data.

Training data

Test data





R2= 0.86

R2=0.47

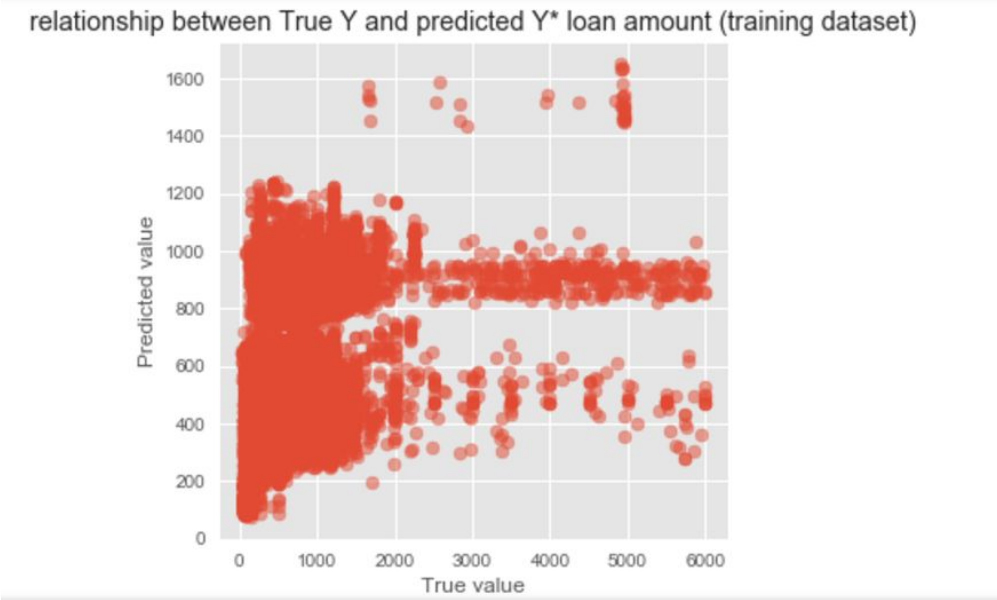
It's still not perfect, but it is a big improvement over our past models - performance on our test data has **doubled**!

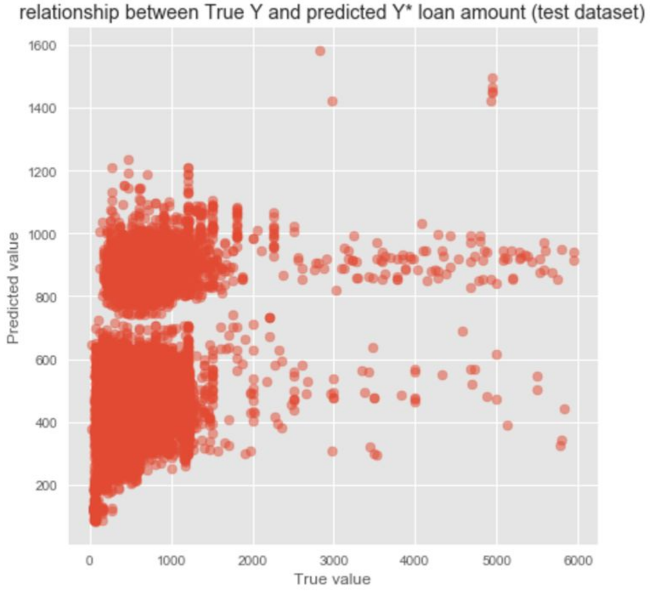
Let’s take a step back and review…

Our linear regression model **underfit** the data.

Training data

Test data


relationship between True Y and predicted Y* loan amount (training dataset)

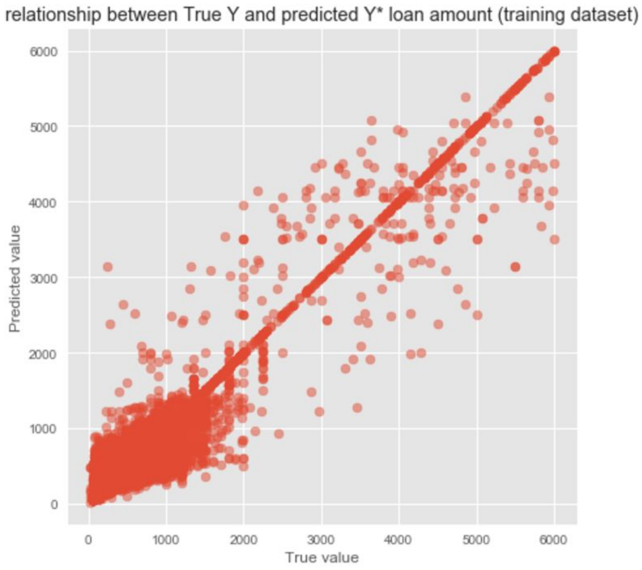

relationship between True Y and predicted Y* loan amount (test dataset)

R2= 0.17

R2=0.16

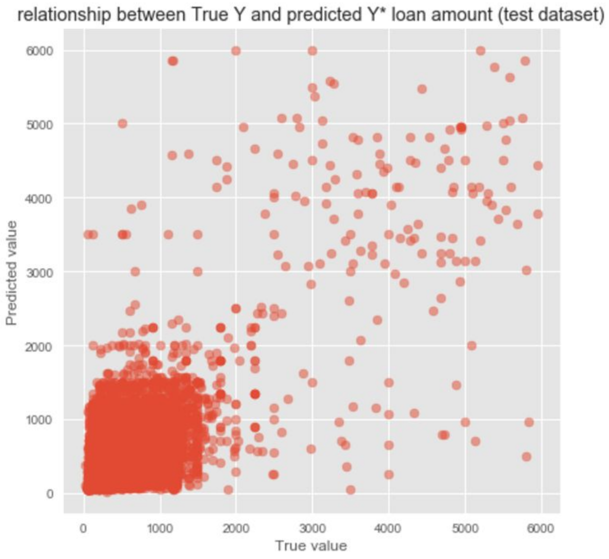
Our decision tree model **overfit** our data.

Training data



R2= 0.93

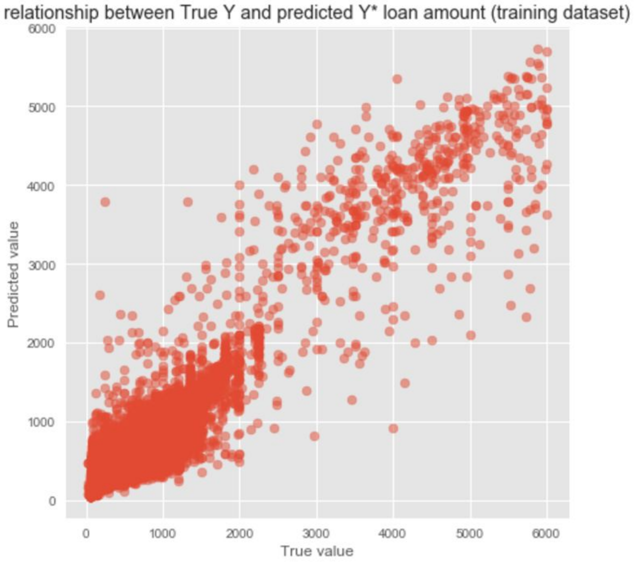
Test data



R2=0.23

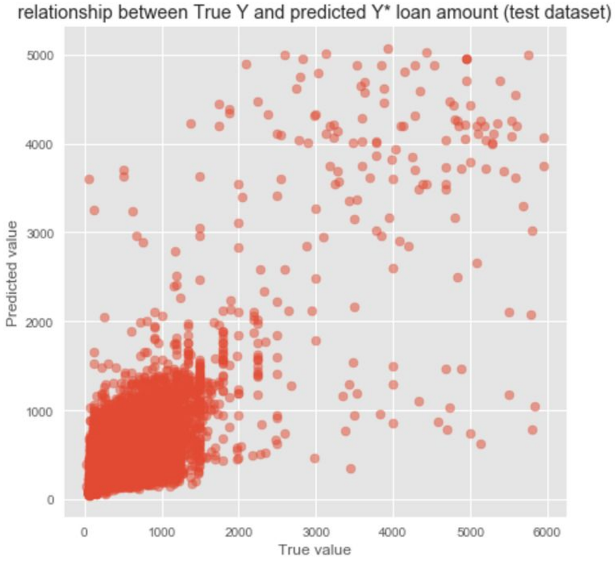
Bagging still **overfit** our data, but yielded a much better R2 than the previous model.

Training data



R2= 0.86

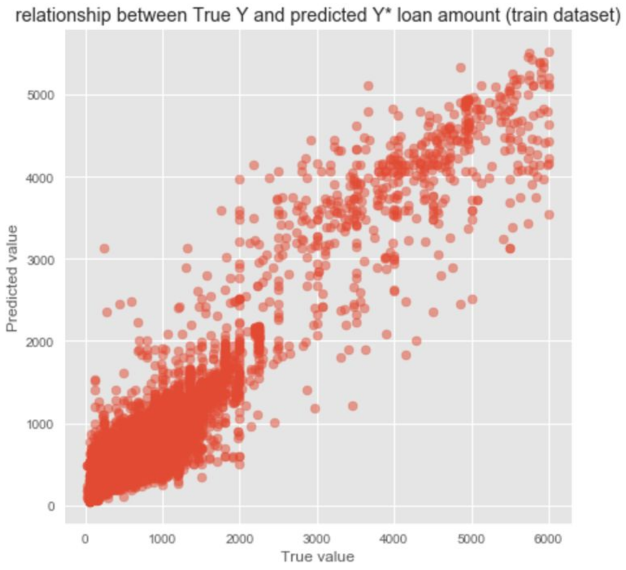
Test data



R2=0.47

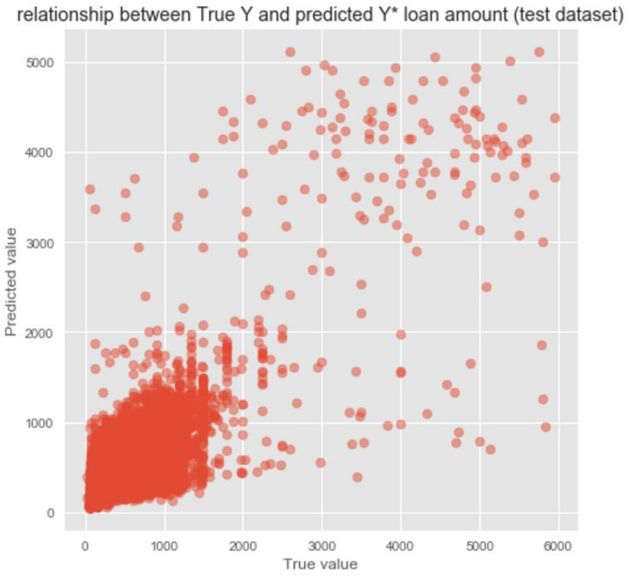
Random Forest, another ensemble approach we'll cover in later modules, yields results that are even better than bagging

Training data



R2= 0.88

Test data



R2=0.51

Using more complex models has resulted in a much better approximation of the true f(x)!

Model	Train R2	Test R2	Condition
OLS	0.17	0.16	Underfitting
Decision tree	0.93	0.23	Overfitting
Bagging	0.86	0.47	Overfitting
Random forest	0.88	0.51	Overfitting

In this case, the generalized model performance improves with complexity of model

However, even after choosing more complex representations of f(x), we are still overfitting.

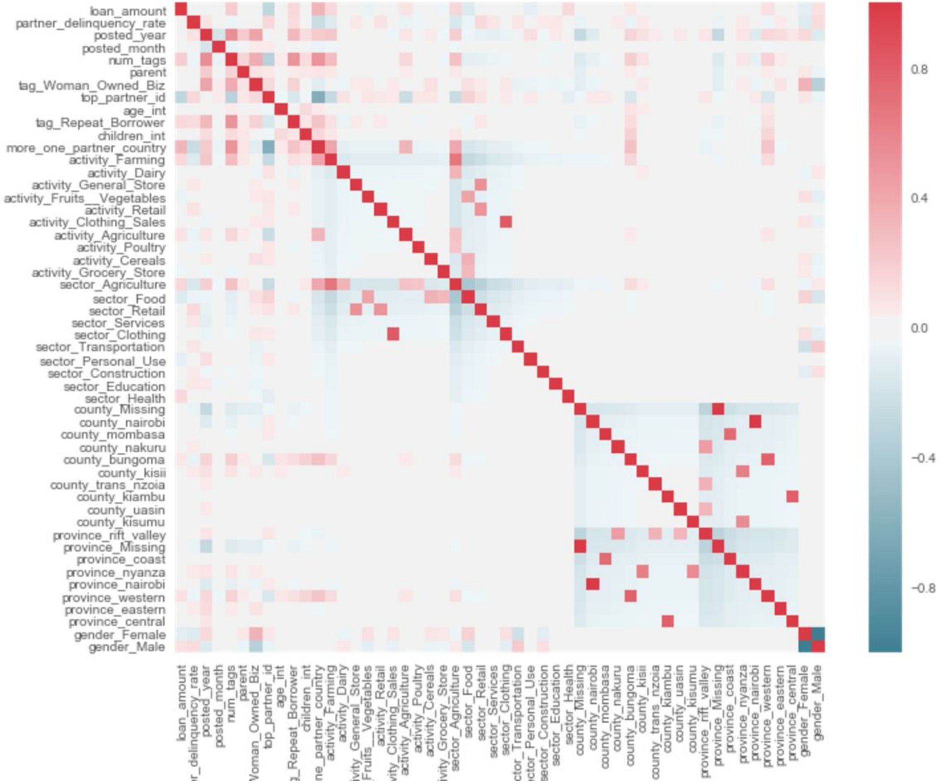
What else can we do to narrow the gap?

- Feature selection
- Find more data
- Hyperparameter optimization

Let’s talk about feature selection.

Feature selection

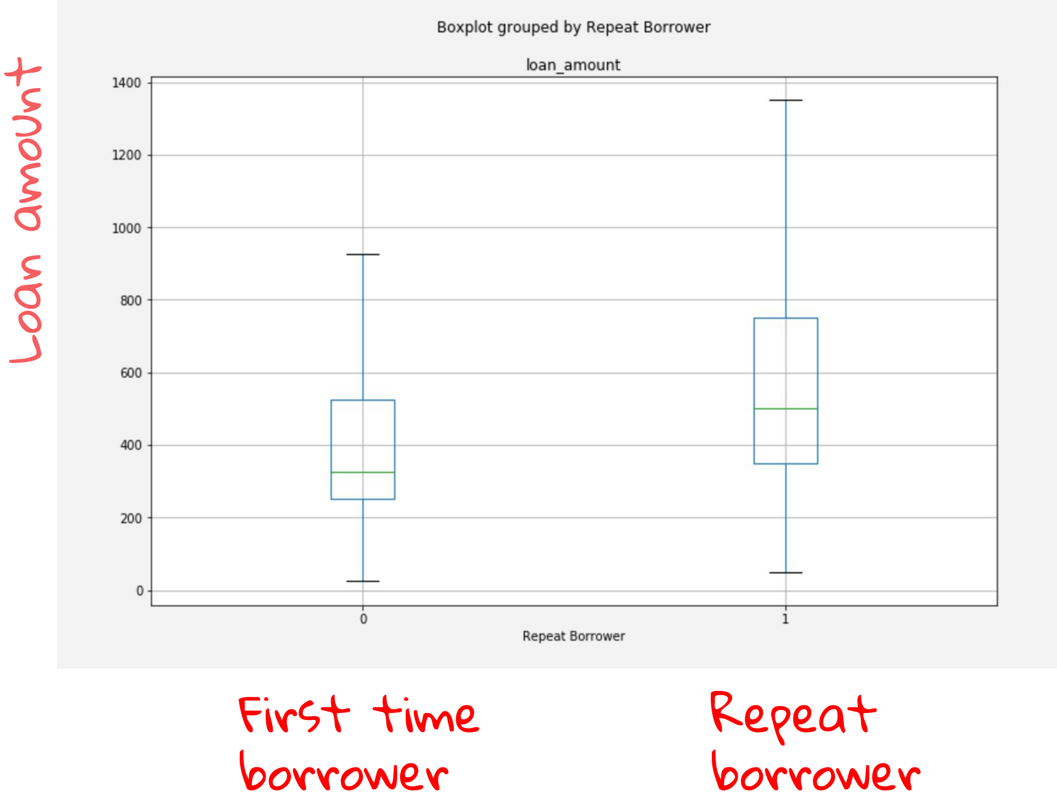
We already discussed excluding heavily correlated features.



What else can we do to improve what features we include in the model?

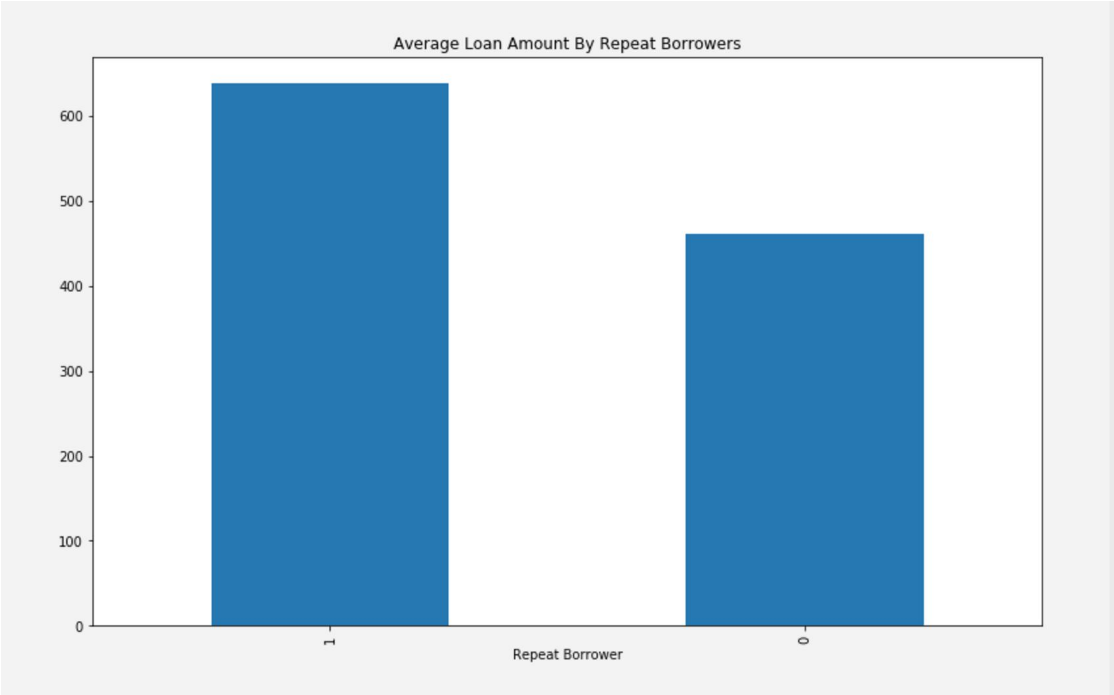
We can look at a few of our most interesting features in depth.

Feature investigation: repeat borrower



This boxplot shows us that repeat borrowers have a much higher distribution of requested loan amounts.

The 25th, 50th, 75th percentile of loan amounts requested by repeat borrowers are all higher.



Repeat borrower

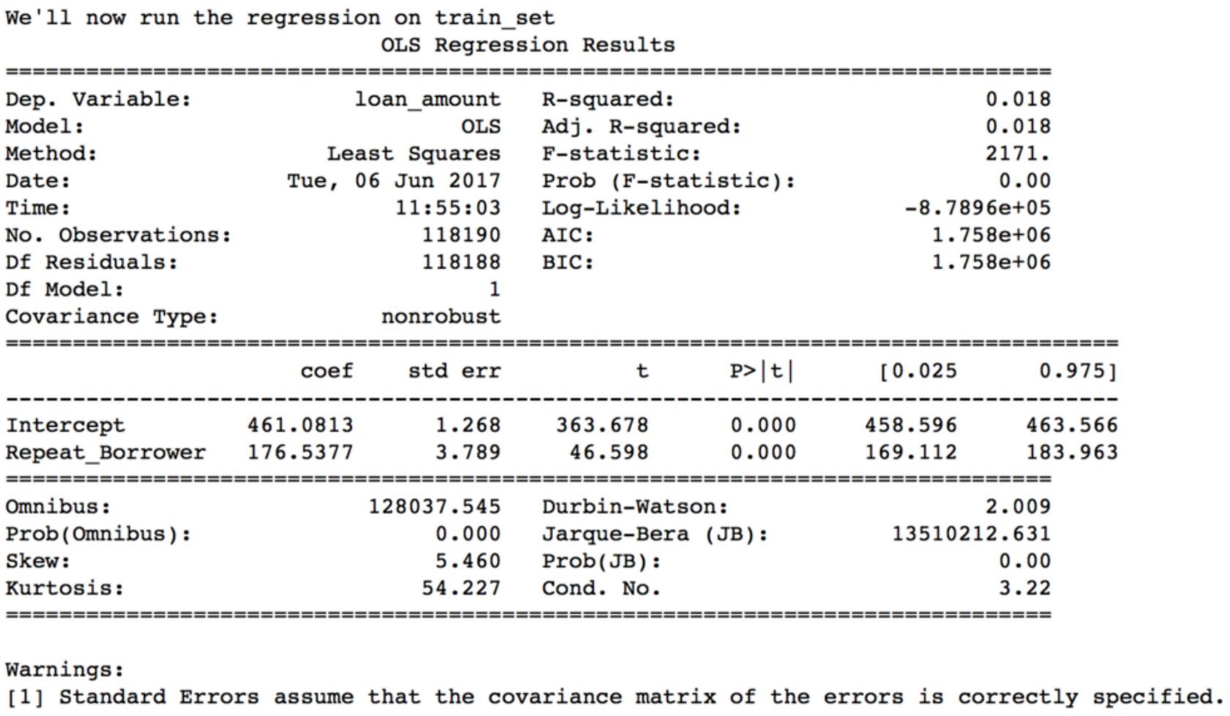
First time borrower

This chart shows that repeat borrowers request ~$120.00 more than first time borrowers.

Why would this be the case?

Univariate regression:

loan_amount~repeat_borrower



```
We'll now run the regression on train_set
                            OLS Regression Results
==============================================================================
Dep. Variable:            loan_amount   R-squared:                       0.018
Model:                            OLS   Adj. R-squared:                  0.018
Method:                 Least Squares   F-statistic:                     2171.
Date:                Tue, 06 Jun 2017   Prob (F-statistic):               0.00
Time:                        11:55:03   Log-Likelihood:            -8.7896e+05
No. Observations:              118190   AIC:                         1.758e+06
Df Residuals:                  118188   BIC:                         1.758e+06
Df Model:                           1
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept         461.0813      1.268    363.678      0.000     458.596     463.566
Repeat_Borrower   176.5377      3.789     46.598      0.000     169.112     183.963
==============================================================================
Omnibus:                   128037.545   Durbin-Watson:                   2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         13510212.631
Skew:                           5.460   Prob(JB):                         0.00
Kurtosis:                      54.227   Cond. No.                         3.22
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

R2= 0.17

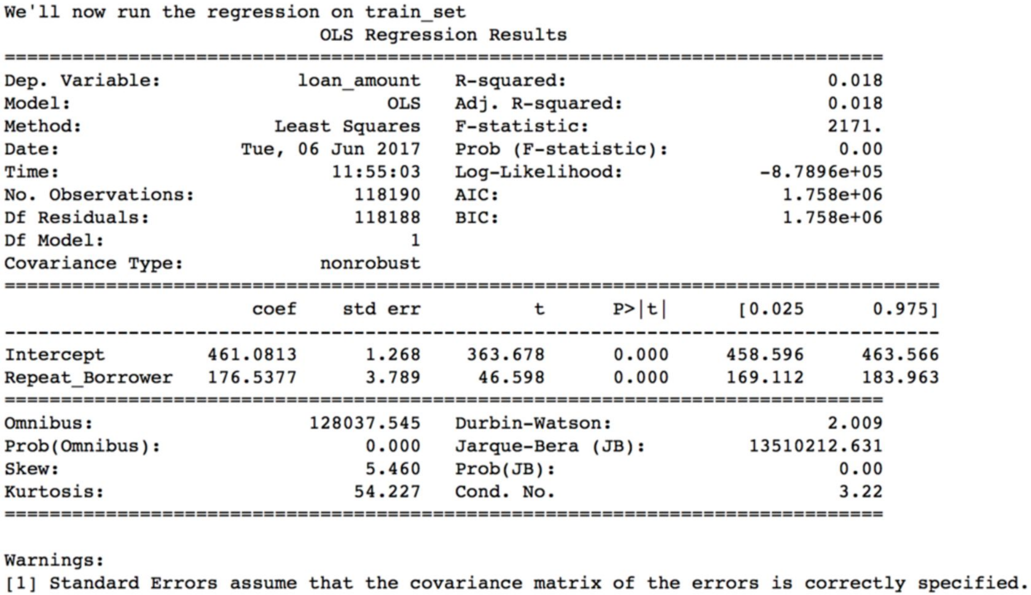
p-score=0.000

t-stat=46.5

intercept=461

repeat_borrower=177

Question: What is the predicted loan amount of a repeat borrower?



```
We'll now run the regression on train_set
                            OLS Regression Results
==============================================================================
Dep. Variable:            loan_amount   R-squared:                       0.018
Model:                            OLS   Adj. R-squared:                  0.018
Method:                 Least Squares   F-statistic:                     2171.
Date:                Tue, 06 Jun 2017   Prob (F-statistic):               0.00
Time:                        11:55:03   Log-Likelihood:            -8.7896e+05
No. Observations:              118190   AIC:                         1.758e+06
Df Residuals:                  118188   BIC:                         1.758e+06
Df Model:                           1
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept         461.0813      1.268    363.678      0.000     458.596     463.566
Repeat_Borrower   176.5377      3.789     46.598      0.000     169.112     183.963
==============================================================================
Omnibus:                   128037.545   Durbin-Watson:                   2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         13510212.631
Skew:                           5.460   Prob(JB):                         0.00
Kurtosis:                      54.227   Cond. No.                         3.22
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

intercept=461

repeat_borrower=177

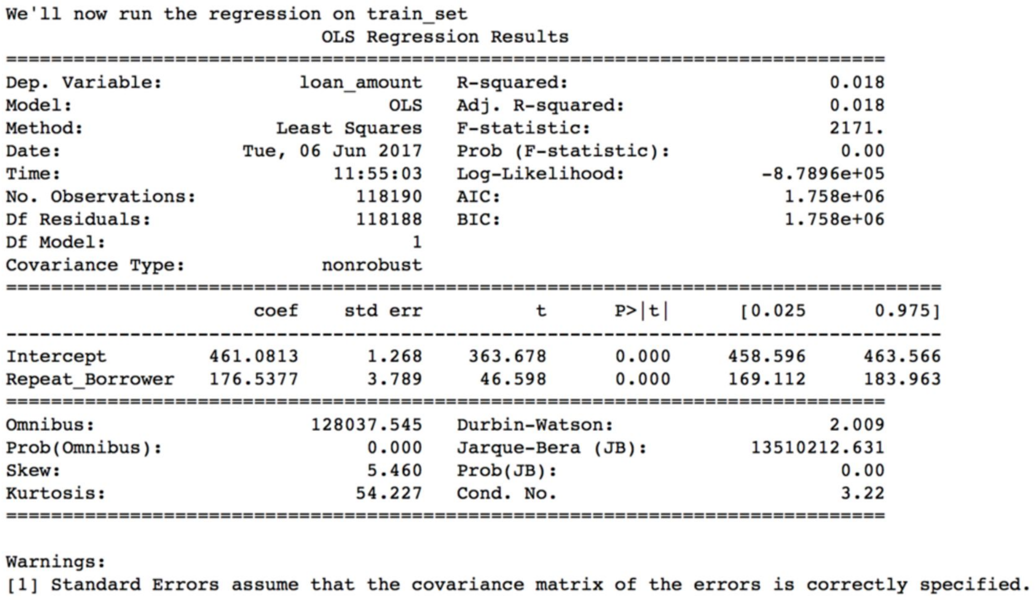
Our model predicts that a repeat borrower will request $637

Y*=461+176
Y*=637

intercept=461

repeat_borrower=177

Question: Is being a repeat borrower a statistically significant feature?



```
We'll now run the regression on train_set
                            OLS Regression Results
==============================================================================
Dep. Variable:            loan_amount   R-squared:                       0.018
Model:                            OLS   Adj. R-squared:                  0.018
Method:                 Least Squares   F-statistic:                     2171.
Date:                Tue, 06 Jun 2017   Prob (F-statistic):               0.00
Time:                        11:55:03   Log-Likelihood:            -8.7896e+05
No. Observations:              118190   AIC:                         1.758e+06
Df Residuals:                  118188   BIC:                         1.758e+06
Df Model:                           1
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept         461.0813      1.268    363.678      0.000     458.596     463.566
Repeat_Borrower   176.5377      3.789     46.598      0.000     169.112     183.963
==============================================================================
Omnibus:                   128037.545   Durbin-Watson:                   2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         13510212.631
Skew:                           5.460   Prob(JB):                         0.00
Kurtosis:                      54.227   Cond. No.                         3.22
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

R2= 0.17

p-score=0.000

t-stat=46.5

intercept=461.08

repeat_borrower=176

p-value=0.000

The P value, or calculated probability, is the probability of finding the observed, or more extreme, results when the **null hypothesis** (H0) of a study question is true.

The null hypothesis is the hypothesis that any observed relationship between x and y are just random noise.

t-stat=46.5

When you run a hypothesis test, you use the T statistic with a p value. The p-value tells you what the odds are that your results could have happened by chance.
The greater the T, the more evidence you have that your feature is significantly different from average. A smaller T value is evidence that your feature is not significantly different from average.

p-value=0.000

P VALUE < .05 we are reasonably confident (95% confidence level)

t-stat=46.5

At a t-stat>2 we are reasonably confident (95% confident level).

R2= 0.17

R-squared is a statistical measure of how close the data are to the fitted regression line.

R-squared = Explained variation / Total variation

R-squared is always between 0 and 100%:

0% indicates that the model explains none of the variability of the response data around its mean.

100% indicates that the model explains all the variability of the response data around its mean.

Repeat_borrower is statistically significant!

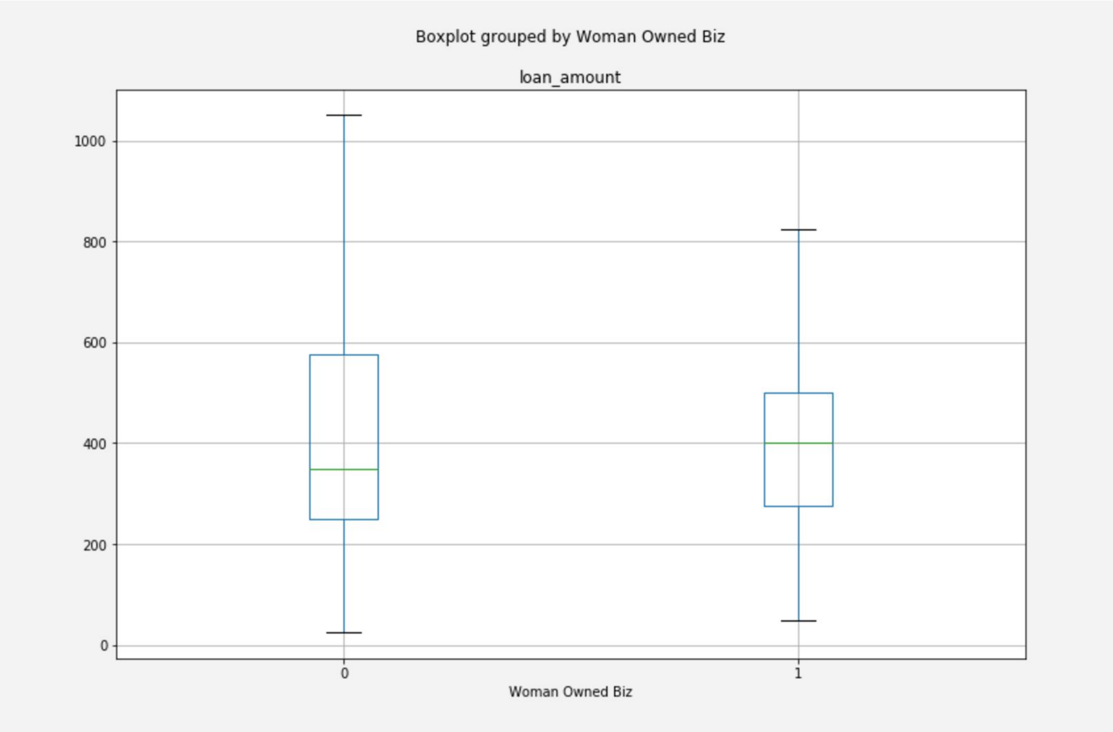
R2= 0.17

p-score=0.000

t-stat=46.5

We will be adding this feature to our our final model.

Feature investigation: woman-owned business

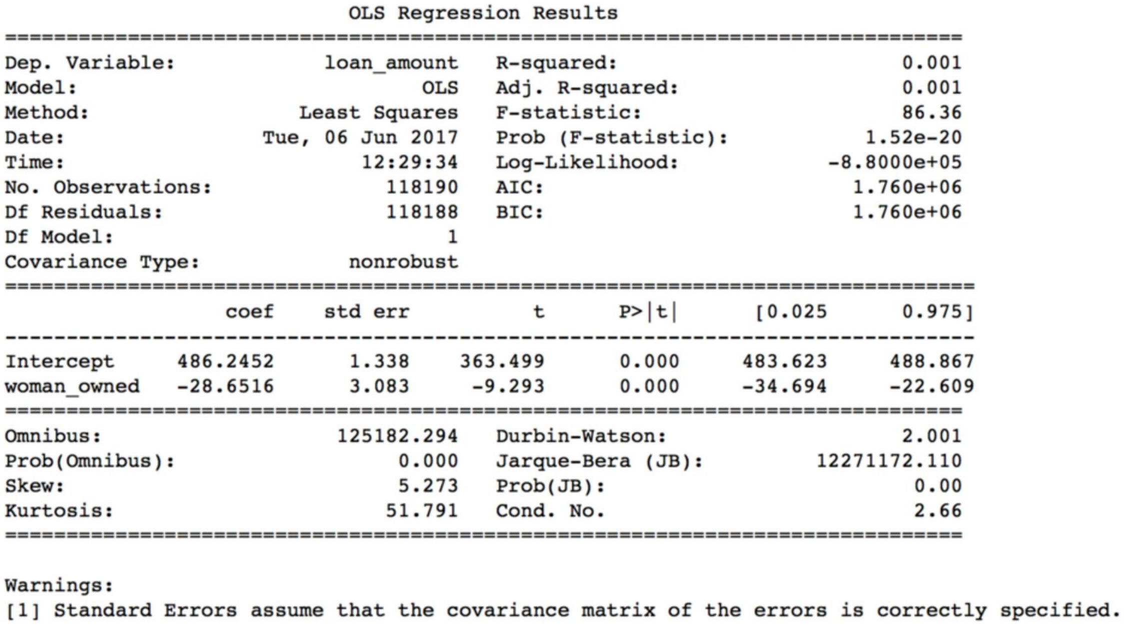


This boxplot shows that women owned businesses have a lower distribution of loan amounts requested.

What about the median?

Univariate regression:

loan_amount~woman_owned



```
                            OLS Regression Results
==============================================================================
Dep. Variable:            loan_amount   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                  0.001
Method:                 Least Squares   F-statistic:                     86.36
Date:                Tue, 06 Jun 2017   Prob (F-statistic):           1.52e-20
Time:                        12:29:34   Log-Likelihood:            -8.8000e+05
No. Observations:              118190   AIC:                         1.760e+06
Df Residuals:                  118188   BIC:                         1.760e+06
Df Model:                           1
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept     486.2452      1.338    363.499      0.000     483.623     488.867
woman_owned   -28.6516      3.083     -9.293      0.000     -34.694     -22.609
==============================================================================
Omnibus:                   125182.294   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         12271172.110
Skew:                           5.273   Prob(JB):                         0.00
Kurtosis:                      51.791   Cond. No.                         2.66
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

R2= 0.001

p-score=0.000

t-stat=-9.293

coefficient=-28.65

intercept=486.24

Question 1) What is the predicted loan amount of a female business owner?

R2= 0.001

p-score=0.000

t-stat=-9.293

coefficient=-28.65

intercept=486.24

Our model predicts that a female business owner will request $458.

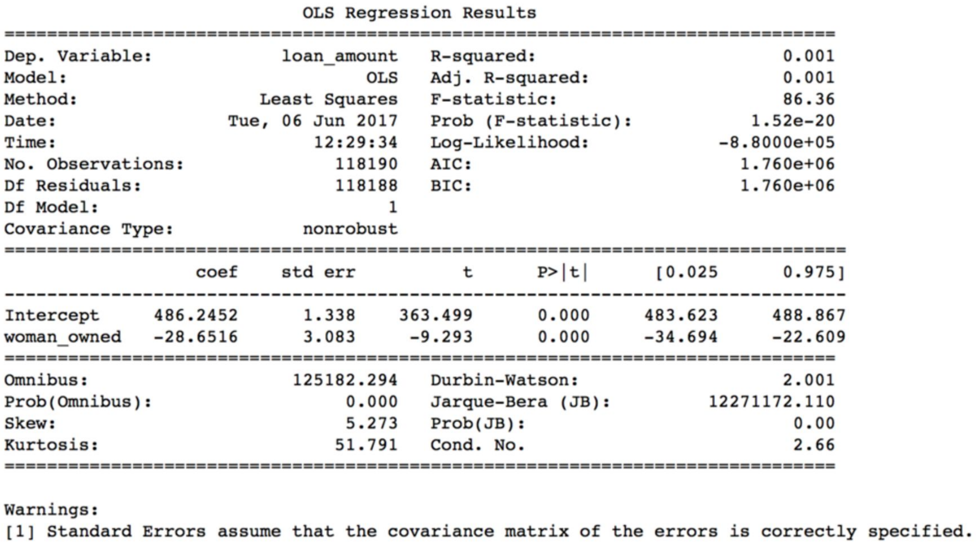
Y*=486-28
Y*=458

Y*=intercept+woman_business

intercept=
486.24

woman_business=-28.65

Question: What is the predicted loan amount of a male business owner?



```
                            OLS Regression Results
==============================================================================
Dep. Variable:            loan_amount   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                  0.001
Method:                 Least Squares   F-statistic:                     86.36
Date:                Tue, 06 Jun 2017   Prob (F-statistic):           1.52e-20
Time:                        12:29:34   Log-Likelihood:            -8.8000e+05
No. Observations:              118190   AIC:                         1.760e+06
Df Residuals:                  118188   BIC:                         1.760e+06
Df Model:                           1
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept     486.2452      1.338    363.499      0.000     483.623     488.867
woman_owned   -28.6516      3.083     -9.293      0.000     -34.694     -22.609
==============================================================================
Omnibus:                   125182.294   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         12271172.110
Skew:                           5.273   Prob(JB):                         0.00
Kurtosis:                      51.791   Cond. No.                         2.66
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

R2= 0.001

p-score=0.000

t-stat=-9.293

coefficient=-28.65

intercept=486.24

Our model predicts that a male business owner will request $486.

Y*=486-0
Y*=486

Y*=intercept

intercept=
486.24

woman_business=-28.65

Question: Is the female owned business feature statistically significant?

R2= 0.001

p-score=0.000

t-stat=-9.293

coefficient=-28.65

intercept=486.24

The female owned business is not statistically significant.

R2= 0.001

p-score=0.000

t-stat=-9.293

coefficient=-28.65

intercept=486.24

Should we still include it in our model?

Should we still include female owned business in our model?

Often, a feature that is not statistically significant by itself can provide a significant performance improvement when taken into account with other features.

However, a linear regression does not take into account interactions between terms unless we explicitly express this in f(x). More complex models like decision trees do account for this interaction.

No right answer, depends on relative importance but it is cheap to add to final model and re-evaluate importance.

Now that you know generally how to **evaluate and validate a model**, we can dive deep into the nuts and bolts of other algorithms.

Our next module will cover **decision trees**.

You are on fire! Go straight to the next module here.

Need to slow down and digest? Take a minute to write us an email about what you thought about the course. All feedback small or large welcome!

Email: sara@deltanalytics.org

Congrats! You finished module 4!

Find out more about Delta's machine learning for good mission here.