Statistics unit 6- on interpreting linear regression runs

What’s the point?

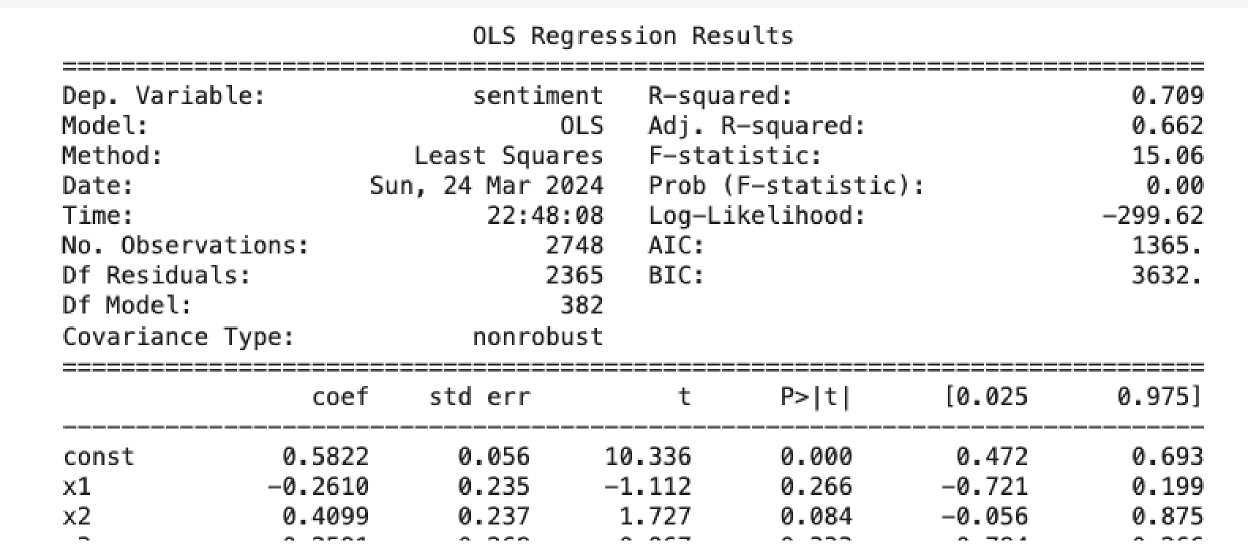
Regression is perhaps the most widely used statistical procedure for interpreting a phenomenon in the “real world” by collecting data about it.   It recovers a linear function a “model” — of possibly high input dimension — that resembles as close as possible the unknown function that is “hidden” in the data. This function can be used for *prediction,* to estimate values for inputs that are not in the data or for *inference,* to attribute importance to the input features.

“Prediction” is evaluated by accuracy, for which there are several measures that measure slightly different things. Inference is evaluated by the sensitivity and significance of a variable’s coefficient — or directly by it’s effect on prediction when adding or removing it from the model.

All of the concepts that apply to linear regression can be applied to more advanced machine learning models.

Understanding the outputs

The OLS `statsmodels` output are similar to the diagnostics found in many statistical packages. Unfortunately since they are not as widely used in machine learning, packages such as `scikit learn` don’t include them.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:              sentiment   R-squared:                       0.709
Model:                            OLS   Adj. R-squared:                  0.662
Method:                 Least Squares   F-statistic:                     15.06
Date:                Sun, 24 Mar 2024   Prob (F-statistic):               0.00
Time:                        22:48:08   Log-Likelihood:                -299.62
No. Observations:                2748   AIC:                             1365.
Df Residuals:                    2365   BIC:                             3632.
Df Model:                         382
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.5822      0.056     10.336      0.000       0.472       0.693
x1            -0.2610      0.235     -1.112      0.266      -0.721       0.199
x2             0.4099      0.237      1.727      0.084      -0.056       0.875
```

- *DF Model:* The model degrees of freedom equals the number of variables in the model, with a correction for the number of duplicates. It is derived from the size of the basis of the Gram Matrix. This is important because increasing the number of variables, even if they are entirely irrelevant (e.g. “insignificant”) will lead to better measures of R-squared and log-likelihood, but not necessarily to a better model.
- *R-squared:* An “*analysis of variance.” S*imply the fraction of variance in the outcome variable *y* that is “explained” by the regression. Equivalently, 1 -

RSS(residuals) / RSS(y). This suffers from the fault that it always increases as variables are added, even if they are irrelevant to the model.

- *Adjusted R-squared:* A feeble attempt to compensate for the number of variables.
- *F-Statistic* It gives the probability that the function learned is statistically insignificant. It's not that useful.
- *Log-Likelihood:* Also called "cross entropy" - the sum of the probabilities assigned to the output data-points by the model. Assuming the residual error is normally distributed in linear regression, this works just like *Mean Square Error. Log-Likelihood* is used as a objective function to be maximized for any probability model.
- ***AIC* A correction to Log-Likelihood for the number of variables (strictly the degrees of freedom) in the model. It penalizes the log likelihood as the number of variables increases, to obtain an "effective" log-likelihood. It can be used to compare different models applied to the same data. Lower AIC is better.**
- ***BIC* Like AIC, it adjusts the Log-Likelihood for the number of variables. The subtle difference between AIC and BIC is that AIC attempts to choose the model that makes the best *prediction*, whereas BIC chooses the model with the most relevant set of variables.**
- There are other measures to evaluate model predictions. In the binary case, where results are reduced to 0-1, then *0-1* error is simply the number of correct answers over the total number of cases. Since there are two ways the model can be wrong, (and two ways it can be right), the full visualization of the model performance is a 2 × 2 *confusion matrix.* This gives a detailed breakdown of the error.

Out of sample error.

All these evaluation measures use the same data to evaluate the model that it was trained on. Measures, except R-squared and log-likelihood make adjustments so that the measure approximates the value that would be obtained on new, unseen data; that is they approximate the "out-of-sample" error, to avoid the notorious problem of "over-fitting." An alternative, if enough data is available, is to "hold-out" a test subset of the data not yet seen by the model to run for evaluation purposes. Accuracy on the hold-out data set is the gold standard for machine learning modelling.

Alternately one could use bootstrap resampling to estimate out-of-sample errors.

Variable Selection methods

"Inference" — Finding the right model means picking the relevant subset of variables, or a subset of some combination of them. This is subtle, because inferring which are the "true" variables that most likely generated the data does not necessarily give the set of variables that give the best prediction.

What kinds of variables?

The input to a regression model is the design matrix, often created from the "raw data." Most of the modelling process comes down to creating the "features" that go into the design matrix from the raw data. The hope is that with a wide enough variety of features, one will capture relevant aspects of the data with some of them. With more features one has expanded the number of possible variables for the model. So after generating new features, the next task is to select those most relevant.

Here are common ways variables are created:

- Convert "unstructured" data into vectors. This comes in many flavors; we have been working with a recently developed method for text called "embedding vectors." One could also create features directly from the distribution of words in the text.
- Rescaling, centering (e.g. "affine" transformations), or non-linear transformations of numeric values, e.g. as in polynomial regression.
- Regression terms that are functions of multiple features. For instance, "Interaction terms" are the product of two (or more) features.
- "One hot" encodings of variables that have a "small" number of possible values.
- "Low rank" approximations, such as *principal component analysis* or other dimensionality reduction methods.
- Recognizing other characteristics to embellish the data, such as the various sources different samples came from.

Scoring variables

Linear regression provides a straightforward if flawed way to compare the contribution of each variable, by computation of it's "standard error". If the estimation interval is "significantly" different than zero, the premise is that the variable is relevant. This fails in two ways:

1. Beta coefficients are derived from covariance, so they only detect linear relationships with the outcome variable
2. Variables can have dependencies (one variable is able to predict another). In linear models this is called "multi-collinearity". This is avoided if the variables are orthogonal, as with PCA components. Otherwise the design matrix columns the regression coefficients are no longer unique and tend to be unstable.

So one approach is to orthogonalize variables by Gram-Schmidt or PCA transformations so that standard error methods work. However PCA rates components only by their contribution to the total design matrix variance, so determining their relevance to the outcome takes another step.

Scoring sets of variables

Often to keep interpretability one wants to keep the original variables, and find a subset of them. AIC and BIC scores can guide selection, since they penalize error measures (e.g. accuracy) for extra variables. Similarly, one could use cross validation.

But how does one choose a subset?

- Looking at coefficient standard errors followed by testing is a conventional if flawed approach. Or using other measures of relevance with the outcome variable, such as mutual information.
- Recently developed *regularization* methods such as *ridge regression* will “shrink” coefficient values toward zero, often improving prediction accuracy at the cost of biasing coefficients “downward.” The same idea can be applied to variable selection — presuming those coefficients that suffer less shrinkage may be the more relevant ones. One could then plug the selected set of “shrunk” variables into a non-penalized version of the model to try to recover their “true” values.

Any and all of these methods can work in combination, assuming one has a “gold standard” by which to compare their performance.