Linear regression

Part 4: Model validity and performance evaluation

By: Noureddin Sadawi, PhD

University of London

Validity

- Always use a scatter plot to visualise the dependent variable vs the independent variable.
- Use a straight line regression if the scatter plot indicates a linear association exists in the data.
- The scatter of points around the determined line should be approximately the same along the line.
- The previous point means that the variance of the dependent variable Y is the same for all values of X (i.e. the independent variable).

Confidence interval for the line

- It is possible to calculate a confidence interval around the linear regression line (i.e. $y = a + bx$).
- The C.I. about the line is often at its narrowest at the mean of the dependent variable.
- Extrapolating beyond the data is dangerous.
- This is because the C.I. becomes very wide.
- Usually we calculate the 95% C.I.
- This interval will contain the true population line on 95% of occasions.

Estimation and prediction

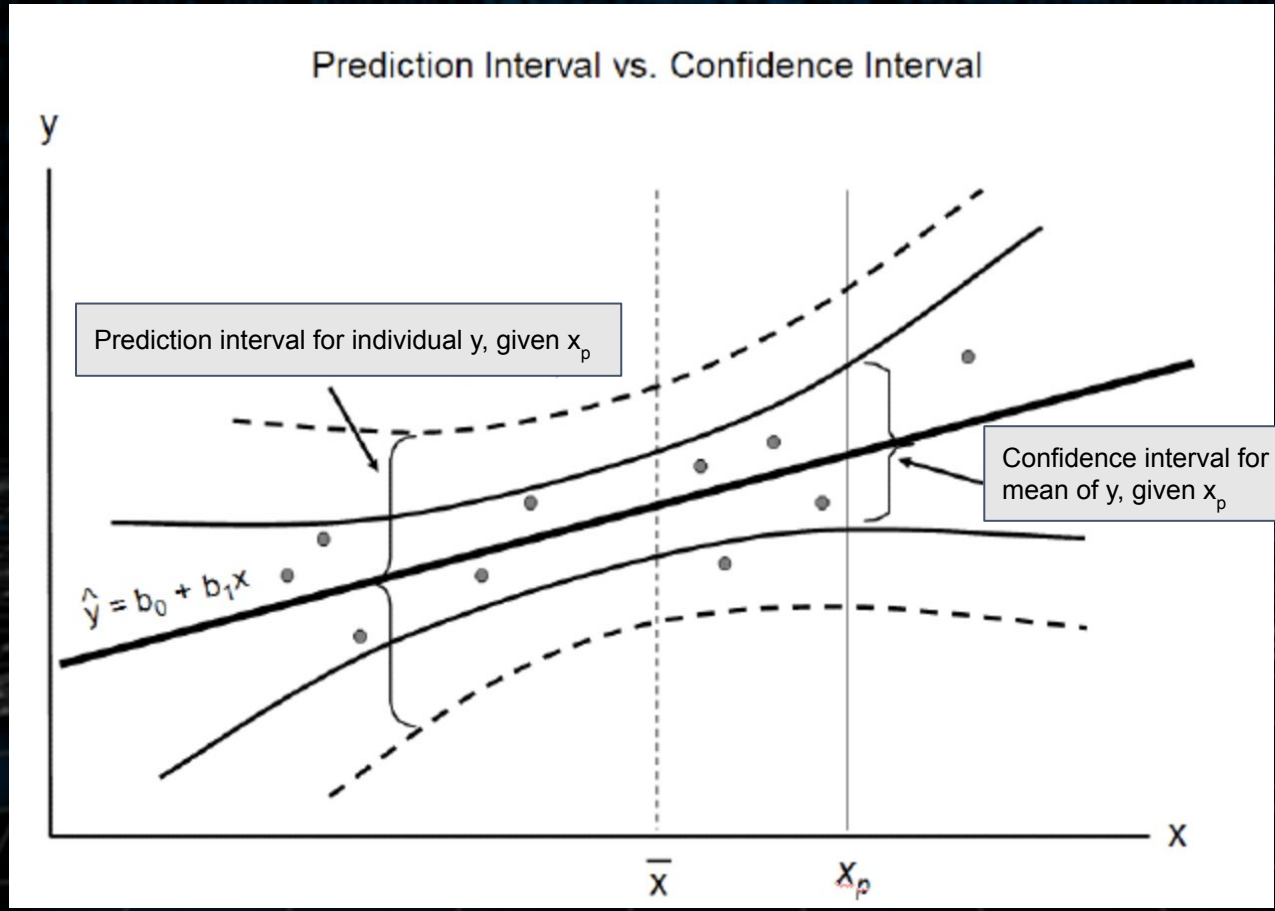
- If a significant relationship exists between *X* and *Y*, and the coefficient of determination shows it is a good fit, then the estimated regression equation would be useful for estimation and prediction.
- Point estimation: the estimated regression equation can be used to develop a point estimate of either the mean value of *Y* or an individual value of *Y* corresponding to a particular value of *X*.

Interval estimation

- Point estimates provide no information regarding the precision associated with an estimate. For that we require interval estimates. There are two types:
  a. The first type, known as *confidence interval*, is an interval estimate of the mean value of *Y* for a given value of *X*.
  b. The second type is a *prediction interval*, used whenever we want an interval estimate of an individual value of *Y* for a given value of *X*.
- The point estimate of the mean value of *Y* is the same as the point estimate of an individual value of *Y*. However, the interval estimates we obtain for the two cases are different. The prediction interval has a larger margin of error.

Prediction interval vs confidence interval

- The prediction interval predicts the range in which a future individual observation will be.
- The confidence interval shows the likely range of values associated with some statistical parameter of the data, such as the population mean.



Example

- xbar = 387.0
- ybar = 10.1
- b = sum((x - xbar)*(y - ybar)) / sum((x - xbar)^2)
- b = 0.029796

- a = ybar - b * xbar
- a = -1.431176

- yhat = -1.431176 + 0.029796 * x

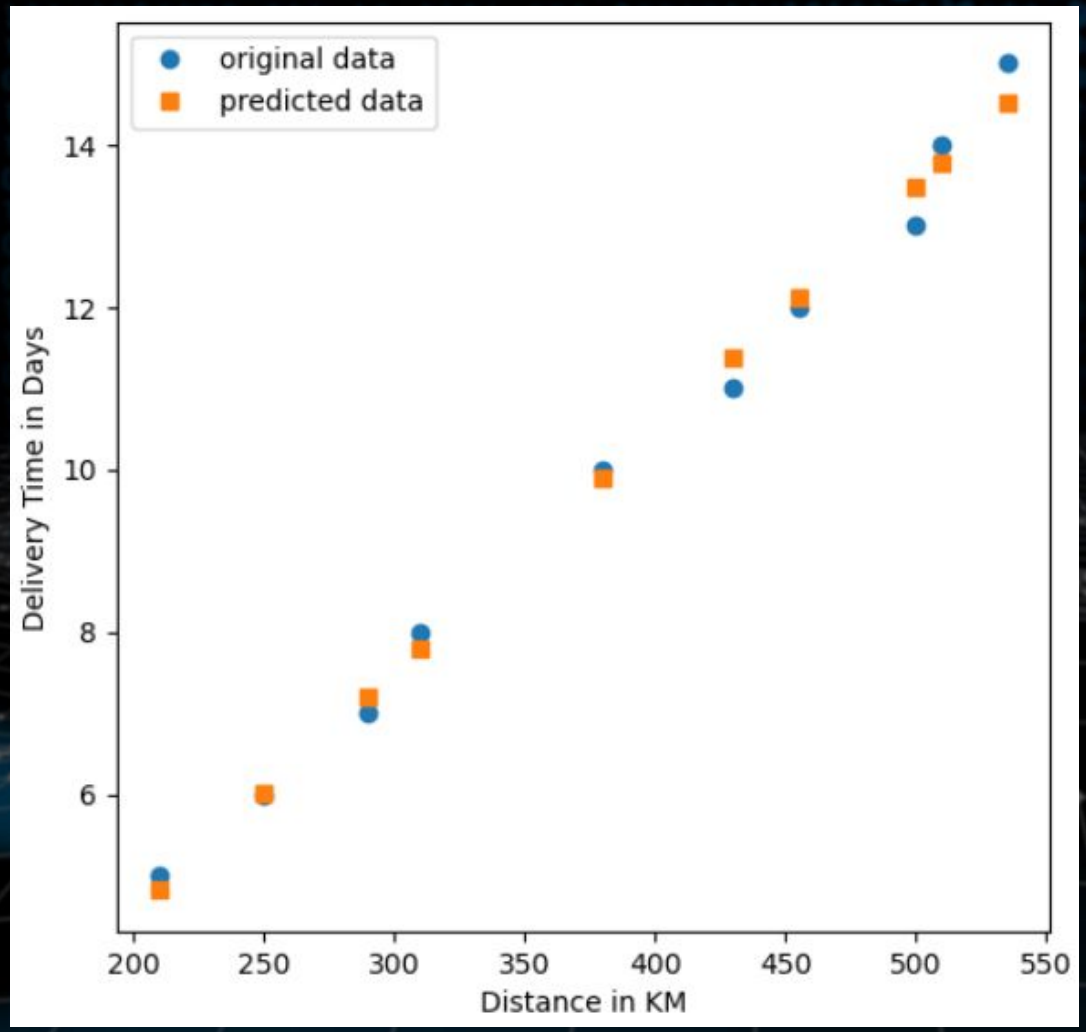
x	y	x-xbar	y-ybar
210	5	-177	-5.1
290	7	-97	-3.1
250	6	-137	-4.1
500	13	113	2.9
310	8	-77	-2.1
430	11	43	0.9
455	12	68	1.9
380	10	-7	-0.1
535	15	148	4.9
510	14	123	3.9

Example

- yhat = -1.431176 + 0.029796 * x
- error = y - yhat
- se = $error^2$
- n = 10
- mse = sum(se)/n = 0.07989
- rmse = sqrt(MSE) = 0.2826568
- R^2 = 1.0 - (var(error) / var(y)) = 0.99

x	y	yhat	e=y-yhat
210	5	4.82605125	0.17394875
290	7	7.2097569	-0.2097569
250	6	6.01790407	-0.01790407
500	13	13.46698423	-0.46698423
310	8	7.80568331	0.19431669
430	11	11.38124179	-0.38124179
455	12	12.1261498	-0.1261498
380	10	9.89142576	0.10857424
535	15	14.50985545	0.49014455
510	14	13.76494744	0.23505256

Original vs predicted



Python code

```python
import numpy as np
# Independent variable
x = np.array([210,290,250,500,310,430,455,380,535,510])
# Dependent variable
y = np.array([5,7,6,13,8,11,12,10,15,14])
# Means of indep and dep variables
xbar = np.mean(x)
ybar = np.mean(y)
# Apply equations to find b and a
b = np.sum((x - xbar)*(y - ybar)) / np.sum((x - xbar)**2)
a = ybar - b * xbar
# Plug in the values of the dep variable into the line equation to obtain yhat
yhat = a + b * x

# Compute the error (i.e. residuals)
error = y - yhat

# Compute metrics
SE = error**2 # squared error
MSE = np.mean(SE) # mean squared error
RMSE = np.sqrt(MSE) # Root Mean Squared Error, RMSE

SST = np.sum((y - ybar)**2)
SSR = np.sum((yhat - ybar)**2)
Rsquared = SSR/SST
# This should give you the same result for Rsquared
#Rsquared = 1.0 - (np.var(error) / np.var(y))
```

Better evaluation

- It is not always a good idea to evaluate models on the data used to build them (this data is known as the **training data**).
- One method is to randomly split the data into two non-overlapping parts:
  - Part 1: the **training data**, used to build the model
  - Part 2: the **test data**, used to evaluate the model

- Because we know the actual values of the outcome in the test data, we can compare this outcome against the predicted values.

Cross validation (CV)

- The k-fold cross-validation procedure involves splitting the data into *k* folds.
- The first *k-1* folds are used to train a model, and the holdout *k*th fold is used as the test set.
- This process is repeated and each of the folds is given an opportunity to be used as the holdout test set.
- A total of *k* models are built and evaluated, and the performance of the model is calculated as the mean of these runs.

Cross validation (CV)

Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

- Compute an evaluation metric for each iteration.
- In the end, compute the average and standard deviation for all those metric values.
- More on CV later in this module.

Key ideas

- ‘The regression equation models the relationship between a response variable *Y* and a predictor variable *X* as a line.
- A regression model yields fitted values and residuals — predictions of the response and the errors of the predictions.
- Regression models are typically fit by the method of least squares.
- Regression is used both for **prediction** and **explanation**.’

(Bruce and Bruce *Practical statistics for data scientists*, second edition, 2020).