STA 3180 Statistical Modelling: Regression

Lecture Notes on Regression for STA 3180 Statistical Modelling

Regression is a statistical technique used to analyze the relationship between two or more variables. It is used to identify the effect of one or more independent variables on a dependent variable. It is also used to predict the value of the dependent variable based on the values of the independent variables.

Key Concepts:

• Linear Regression: Linear regression is a type of regression analysis that models the relationship between two or more variables using a linear equation. The equation is of the form y = mx + b, where y is the dependent variable, m is the slope of the line, x is the independent variable, and b is the intercept.

• Multiple Regression: Multiple regression is a type of regression analysis that models the relationship between two or more independent variables and a dependent variable. The equation is of the form y = b0 + b1x1 + b2x2 + ... + bnxn, where y is the dependent variable, b0 is the intercept, b1, b2, ..., bn are the coefficients of the independent variables x1, x2, ..., xn.

• Polynomial Regression: Polynomial regression is a type of regression analysis that models the relationship between two or more variables using a polynomial equation. The equation is of the form y = b0 + b1x1 + b2x2^2 + ... + bnxn^n, where y is the dependent variable, b0 is the intercept, b1, b2, ..., bn are the coefficients of the independent variables x1, x2, ..., xn.

• Logistic Regression: Logistic regression is a type of regression analysis that models the relationship between one or more independent variables and a binary dependent variable. The equation is of the form p(y=1|x) = 1/(1+e^-(b0 + b1x1 + b2x2 + ... + bnxn)), where p(y=1|x) is the probability of the dependent variable being 1 given the values of the independent variables x1, x2, ..., xn, b0 is the intercept, and b1, b2, ..., bn are the coefficients of the independent variables x1, x2, ..., xn.

Coding Examples:

```
Start of Code
# Linear Regression
# Import libraries
import numpy as np
from sklearn.linear_model import LinearRegression
# Create data
x = np.array([[1], [2], [3], [4], [5]])
y = np.array([1, 2, 3, 4, 5])
# Create model
model = LinearRegression()
# Fit model
model.fit(x, y)
```

```
# Print results
print('Intercept:', model.intercept_)
print('Coefficient:', model.coef_)
```

End of Code

Start of Code

```
# Multiple Regression
# Import libraries
import numpy as np
from sklearn.linear_model import LinearRegression
# Create data
x = np.array([[1, 2], [2, 4], [3, 6], [4, 8], [5, 10]])
y = np.array([1, 2, 3, 4, 5])
# Create model
model = LinearRegression()
# Fit model
model.fit(x, y)
# Print results
print('Intercept:', model.intercept_)
print('Coefficients:', model.coef_)
```

End of Code

Start of Code

```
# Polynomial Regression
# Import libraries
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
# Create data
x = np.array([[1], [2], [3], [4], [5]])
y = np.array([1, 2, 3, 4, 5])
# Create polynomial features
poly_features = PolynomialFeatures(degree=2)
x_poly = poly_features.fit_transform(x)
# Create model
model = LinearRegression()
# Fit model
model.fit(x_poly, y)
# Print results
print('Intercept:', model.intercept_)
print('Coefficients:', model.coef_)
```

End of Code

Start of Code

```
# Logistic Regression
# Import libraries
import numpy as np
from sklearn.linear_model import LogisticRegression
# Create data
x = np.array([[1], [2], [3], [4], [5]])
y = np.array([0, 0, 1, 1, 1])
# Create model
model = LogisticRegression()
# Fit model
model.fit(x, y)
# Print results
print('Intercept:', model.intercept_)
print('Coefficients:', model.coef_)
```

End of Code