✏️

Chapter 3: Assessing model fit

Created	@September 19, 2024 12:04 AM
Class	Introduction to Regression with statsmodels in Python

Quantifying Model Performance in Linear Regression

Key Details

- Important to assess whether model predictions are reliable
- Several metrics can quantify model performance
- Performance interpretation depends on the context of the dataset

Coefficient of Determination (R-squared)

Definition

- Proportion of variance in the response variable predictable from the explanatory variable
- Ranges from 0 (no better than random) to 1 (perfect fit)
- For simple linear regression, it's the square of the correlation between explanatory and response variables

Interpretation

- Depends on the dataset and field of study
- E.g., 0.5 might be high in psychology, while 0.9 might be poor in other fields

Obtaining R-squared

```
# From model summary
print(model.summary())
```

```
# Directly from model attribute
r_squared = model.rsquared
```

Residual Standard Error (RSE)

Definition

- Measure of the typical size of residuals
- Has the same unit as the response variable

Calculation

```
import numpy as np

# From model attribute
rse = np.sqrt(model.mse_resid)

# Manual calculation
residuals_squared = model.resid ** 2
sum_residuals_squared = np.sum(residuals_squared)
degrees_of_freedom = len(model.resid) - len(model.params)
rse_manual = np.sqrt(sum_residuals_squared / degrees_of_freedom)
```

Interpretation

- Indicates how much predictions are typically wrong
- E.g., RSE of 74 grams means predictions typically differ from observed values by about 74 grams

Related Metrics

Mean Squared Error (MSE)

- Square of the Residual Standard Error

- Available as `model.mse_resid`

Root Mean Square Error (RMSE)

- Similar to RSE but doesn't account for model coefficients in calculation
- Generally, RSE is preferred for model comparisons

Key Takeaways

- R-squared quantifies the strength of the linear relationship
- RSE quantifies the typical prediction error in the original units
- Interpretation of these metrics depends on the specific context of the data and field of study
- For model comparisons, prefer RSE over RMSE

Diagnostic Plots for Linear Regression Models

Key Details

- Several plots can quantify model performance
- These plots help assess if model assumptions are met
- Main assumptions: residuals are normally distributed with mean zero

Types of Diagnostic Plots

1. Residuals vs. Fitted Values Plot

- Purpose: Shows if residuals have non-linear patterns
- Ideal scenario: LOWESS trend line follows y=0 closely
- Interpretation:

- Bream model: Good fit, trend line close to y=0
- Perch model: Poor fit, trend line deviates from y=0

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.residplot(x='fitted_values', y='residuals', data=model_da
ta, lowess=True)
plt.xlabel('Fitted values')
plt.ylabel('Residuals')
plt.title('Residuals vs Fitted')
plt.show()
```

2. Q-Q Plot (Quantile-Quantile Plot)

- Purpose: Checks if residuals follow a normal distribution
- Ideal scenario: Points closely follow the diagonal line
- Interpretation:
  - Bream model: Most points follow the line, with some deviation at extremes
  - Perch model: Points deviate from the line, especially for larger residuals

```
import statsmodels.api as sm

sm.qqplot(model.resid, fit=True, line='45')
plt.title('Q-Q Plot')
plt.show()
```

3. Scale-Location Plot

- Purpose: Shows if residuals have constant variance (homoscedasticity)
- Ideal scenario: LOWESS trend line is approximately horizontal
- Interpretation:

- Bream model: Slight increase in residual size, but relatively constant
- Perch model: Trend line varies significantly, indicating poor fit

```
import numpy as np

normalized_residuals = model.get_influence().resid_studentize
d_internal
sqrt_normalized_residuals = np.sqrt(np.abs(normalized_residua
ls))

sns.regplot(x=model.fittedvalues, y=sqrt_normalized_residual
s, lowess=True)
plt.xlabel('Fitted values')
plt.ylabel('√|Standardized residuals|')
plt.title('Scale-Location')
plt.show()
```

Key Takeaways

- Diagnostic plots provide visual insights into model performance
- Residuals vs. Fitted plot checks for linearity
- Q-Q plot assesses normality of residuals
- Scale-Location plot examines homoscedasticity
- These plots complement numerical metrics like R-squared and RSE
- A good model should show random scatter in residuals, points following the diagonal in Q-Q plot, and constant variance in Scale-Location plot

Outliers in Regression Models

Key Details

- Outliers are unusual data points in datasets

- They can significantly impact regression models
- Two types of outliers:
    1. Extreme explanatory variable values
    2. Points far from model predictions
- Leverage and influence are important metrics for identifying outliers

Types of Outliers

Extreme Explanatory Variables

- Easy to identify in simple linear regression
- Example: Extreme short and long fish in the dataset

Points Far from Model Predictions

- Observations that deviate significantly from expected values
- Example: Fish with zero mass (biologically unlikely)

Metrics for Outlier Detection

Leverage

- Quantifies how extreme explanatory variable values are
- Stored in the 'hat_diag' column of the summary frame
- Measures the first type of outlier (extreme explanatory variables)

Influence

- "Leave one out" metric
- Measures how much the model would change if a data point was removed
- Analogous to torque in physics
- Based on the size of residuals and leverage
- Measured using Cook's distance (stored as 'cooks_d' in the summary frame)

Identifying Outliers

```
# Get influence metrics
influence_summary = model.get_influence().summary_frame()

# Get leverage values
leverage = influence_summary['hat_diag']

# Get Cook's distance (influence metric)
cooks_d = influence_summary['cooks_d']

# Find most influential points
most_influential = influence_summary.sort_values('cooks_d', a
scending=False)
```

Impact of Outliers

- Removing influential points can significantly change regression results
- Example: Removing the shortest fish altered the slope of the regression line

Visualization

```
# Plotting regression with and without influential point
fig, ax = plt.subplots()
ax.scatter(data['length'], data['mass'])
ax.plot(data['length'], model.predict(data['length']), color
='blue', label='Full dataset')
ax.plot(data_without_outlier['length'], model_without_outlie
r.predict(data_without_outlier['length']),
        color='red', label='Without outlier')
ax.set_xlabel('Length')
ax.set_ylabel('Mass')
ax.legend()
plt.show()
```

Key Takeaways

1. Outliers can be identified through extreme explanatory variable values or large deviations from predictions
2. Leverage measures the extremity of explanatory variables
3. Influence (Cook's distance) combines leverage and residual size to measure a point's impact on the model
4. Removing influential points can dramatically change regression results
5. Visualizing data and model predictions with and without outliers helps understand their impact