Statistics in Data Science

Amir Kiani

July 18, 2024

Contents

Part I
Descriptive Statistics

1 Descriptive Statistics

Descriptive statistics provide a powerful summary that can describe and reveal the underlying patterns within a dataset, without making any assumptions about its origin or accuracy.

1.1 Measures of Central Tendency

Measures of central tendency help identify the central or typical value in a dataset.

- *Mean*: The arithmetic average of the dataset, calculated as:

  $$\text{Mean} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

  Example: For a dataset of values 2, 3, 5, 7, the mean would be $(2+3+5+7)/4 = 4.25$.
- *Median*: The middle value in an ordered dataset. If the number of observations is even, it is the average of the two middle numbers. Example: For the dataset 2, 3, 5, 7, median is 4 (average of 3 and 5).
- *Mode*: The value that appears most frequently in the dataset. Example: In the dataset 2, 3, 5, 3, 3 is the mode.

1.2 Measures of Dispersion

These measures provide an understanding of how spread out the values in the dataset are.

- *Variance*: Measures the dispersion of the dataset from the mean.

  $$\text{Variance} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

  Example: For the dataset 2, 3, 5, 7, variance would show how each value varies from the mean 4.25.
- *Standard Deviation*: The square root of variance, providing a measure of dispersion in the same units as the data.

  $$\text{Standard Deviation} = \sqrt{\text{Variance}}$$

  Example: A higher standard deviation indicates a wider spread of values.
- *Range*: The difference between the maximum and minimum values. Example: For the dataset 2, 3, 5, 7, the range is $7 - 2 = 5$.
- *Interquartile Range (IQR)*: The range between the first and third quartile (25th and 75th percentiles), covering the middle 50% of data. Example: It helps in understanding the central bulk of the distribution without the influence of outliers.

1.3 Measures of Shape

These measures describe the shape of the data distribution.

- *Skewness*: Indicates the asymmetry of the data around the mean. Positive skewness indicates a tail on the right side.
- *Kurtosis*: Measures the 'tailedness' of the distribution, with high kurtosis showing heavy tails.

1.4 Measures of Dependence

Describing the relationship between pairs of data.

Covariance

Shows the directional relationship between two variables.

$$\text{Cov}(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})$$

Covariance can be positive (direct relationship), negative (inverse relationship), or zero (no linear relationship).

Pearson's Correlation

Provides a normalized measure of covariance, scaled between -1 and 1.

$$\rho(X,Y) = \frac{\text{Cov}(X,Y)}{\sigma_X \sigma_Y}$$

Example: A correlation of 0.9 between age and salary might indicate that higher age is generally associated with higher salary.

1.5 Descriptive Statistics in Python

Python's `pandas` library offers convenient functions to calculate descriptive statistics:

```
# Calculate basic descriptive statistics
df.describe()

# Calculate specific statistics
mean = df['column_name'].mean()
median = df['column_name'].median()
mode = df['column_name'].mode()
variance = df['column_name'].var()
std_dev = df['column_name'].std()
range_value = df['column_name'].max() - df['column_name'].min()
iqr = df['column_name'].quantile(0.75) - df['column_name'].quantile(0.25)
skewness = df['column_name'].skew()
kurtosis = df['column_name'].kurt()
```

This code snippet provides a hands-on example of how to quickly access key statistics, which are crucial for initial data analysis and decision-making processes.

Part II
Inferential Statistics

Hypothesis Testing is a rich collection of statistical techniques used to determine the statistical significance of certain experimental outputs.

The *p-value* is the probability of seeing the observed data given that the null hypothesis is true. We reject the null hypothesis if the p-value is less than a certain set threshold (typically 0.05). If not, we can't say anything.

There are various kinds of hypothesis tests for different types of data and use cases.

Hypothesis Testing: Continuous Data

- **One-sample t-test:** This test compares the mean of a measured group to a known mean.
- **Two-sample independent (unpaired) t-test:** This test compares the mean of a measured group to another measured group.
- **Two-sample paired t-test:** This test compares the mean of a measured group to itself.
- **Two-tailed tests:** If we don't have a strong suspicion that one mean is larger than another, we use a two-tailed t-test.
- **One-tailed t-test:** If we have a strong suspicion that one mean is larger than another, we use a one-tailed t-test.
- **ANOVA (Analysis of Variance):** We use this test when we're interested in the comparison between multiple groups (≥ 3)

Hypothesis Testing: Categorical Data

- **Chi-Square Test for Goodness of Fit:** This test is designed to assess if there's a statistically significant difference between the observed counts and the expected counts across a set of mutually exclusive categories (e.g., to see if a dice is fair).
- **Chi-Square Test for Independence:** This test evaluates whether two categorical variables are independent or if there is a significant association between them. It compares the observed frequencies of occurrences in the categories of a contingency table with the frequencies expected under the assumption that the variables are independent (e.g., to see whether people's favorite color is related to their favorite food).

Used to determine whether the data from two sources could have come from the same normal distribution.

2 Hypothesis Testing: Continuous Data

2.1 One-sample t-test

Compares the mean of a measured group to a known mean.

Example

If we have data about the average spend per customer for 100 customers in one of our stores, we know our average spend is \$14.5 per customer. Does our store have a different average spend than our competitor?

$$\begin{aligned} H_0 &: \mu = 14.5 \\ H_1 &: \mu \neq 14.5 \\ t &= \frac{\bar{x} - \mu_{\text{test}}}{s/\sqrt{n}} = \frac{x_{\text{our}\bar{\text{s}}\text{tore}} - 14.5}{s_{\text{ourstore}}/\sqrt{100}} \end{aligned}$$

If $t \leq t_{\text{critical}} = 0.05$, we can say that the average spend in our store is different from our competitors.

```
# Python code:
from scipy import stats

store_1 = np.array([...])
one_sample_test= stats.ttest_1samp(store_1, 14.5)
```

2.2 Unpaired Two-sample t-test

Compares the mean of one measured group to another measured group.

Example

We want to know if we have different average spends per customer in our two stores in downtown and suburb.

$$H_0 : \mu_{\text{suburb}} = \mu_{\text{downtown}}$$
$$H_1 : \mu_{\text{suburb}} \neq \mu_{\text{downtown}}$$
$$t = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{S_{\text{pooled}}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

```
# Python code:
from scipy import stats

store_1 = np.array([...])
store_2 = np.array([...])
unpaired_two_sample_test= stats.ttest_1samp(store_1, store_2)
```

2.3 Paired two-sample t-test

Example: Imagine the spending data we have is for the *same* 100 customers at both stores. We want to look at each individual's average spend between the two stores to see if there is a difference.

$$H_0 : \mu_{\text{difference}} = 0$$
$$H_1 : \mu_{\text{difference}} \neq 0$$

```
# Python code:
from scipy import stats

store_1 = np.array([...])
store_2 = np.array([...])
paired_two_sample_test= stats.ttest_rel(store_1, store_2)
```

2.4 ANOVA (Analysis of Variance)

ANOVA, or 'ANalysis Of VAriance', is a statistical test where we are interested in comparisons between multiple groups. In its simplest form, we aim to answer the following question: is any group different from the other groups? This boils down to the following null and alternative hypotheses:

- H_0: The means of group 1, group 2, group 3, ... are all equal.
- H_1: There is at least some difference between the means of groups.

Essentially, we want to know if we have drawn our data for each group from the same distribution, or from different ones.

Real-life examples for applying ANOVA include:

- Deciding if different placement of an item in a store results in better sales numbers;
- Determining which day to contact customers for higher response rates;
- Testing if there is a significant difference between run-times for various processes.

In the simplest case, when we have two groups, ANOVA essentially falls back to the A/B testing framework with a t-test. Its power comes from testing more than two groups when pairwise t-tests would give inaccurate results due to issues with multiple testing.

Let's go back to the first case study and the user engagement data. Using ANOVA, we can answer if the browser preference corresponds to significantly different time spent on the website.

Example

We want to see the average amount is different not just by accident. We use the following Python code to calculate ANOVA:

```
import pandas as pd
from scipy import stats

df = pd.DataFrame(...)
df.groupby("browser").aggregate({"time_on_page": ["mean", "var", "count"]})

>>>
            time_on_page
 Group | Mean | Variance | Count
 ---------------------------------
 Edge     | 5.23 | 42.0     | 41.0
 Chrome   | 8.19 | 41.01    | 42.1
 Firefox  | 9.07 | 39.0     | 40.1
 Safari   | 9.96 | 38.0     | 39.1


anova_data = {}
browser_types = df["browser"].unique()

# slice out the time spent for each browser type
for browser in browser_types:
    anova_data[browser] = df.loc[df["browser"] ==
    browser, "time_on_page"]

stats.f_oneway(anova_data["Chrome"],
               anova_data["Edge"],
               anova_data["Firefox"],
               anova_data["Safari"])
  >>> F_onewayResult(statistic=2.799978112368736, pvalue=0.04025821839055499)

# The test is conclusive with a threshold of 0.05, and we reject the null hypothesis.
# There is statistically significant evidence that browser choice affects
# the average time spent on our website.
```

2.5 Assumptions of Hypothesis Tests

- The samples are independent.
- The data is normally distributed.
- The variances are equal.

3 Hypothesis Testing: Categorical Data

3.1 χ^2-Test for Goodness of Fit

To test if two categorical values are independent, we use the χ^2 test for goodness of fit. It is a hypothesis test designed to assess whether there is a statistically significant difference between the observed counts and the expected counts across a set of mutually exclusive categories.

$$
\begin{aligned}
H_0 &: \text{The dice are fair} \\
H_1 &: \text{The dice are loaded}
\end{aligned}
$$

For example, if the null hypothesis is true, the expected count for 2000 rolls would be:

Outcome	1	2	3	4	5	6
Expected Count	333.33	333.33	333.33	333.33	333.33	333.33

Suppose our observed data on sample counts is this:

Outcome	1	2	3	4	5	6
Observed Count	315	336	334	331	337	347

$$
\begin{aligned}
\chi^2 &= \sum_{k=1}^{6} \frac{(\text{Observed}_k - \text{Expected}_k)^2}{\text{Expected}_k} \\
&= \ldots = 4.21
\end{aligned}
$$

```
# Python code:
from scipy import stats

biased_list = [315, 336, 334, 331, 337, 347]
stats.chisquare(biased_list) >>> (statistic = 4.1, pvalue = 0.51)

# 4.1 is chi_square, 0.51 is the p-value
# Since p-value > 5%, we can't make any conclusions.
```

3.2 χ^2-Test for Independence

A χ^2 test for independence tests across multiple categories (≥ 3).

$$
\begin{aligned}
H_0 &: \text{There is no relationship between the categorical variables} \\
H_1 &: \text{There is a relationship}
\end{aligned}
$$

Example

Let's investigate if BrainStation has a difference in student course preferences across different campuses.

$$H_0 : \text{There is no difference}$$
$$H_1 : \text{There is a difference}$$

Data:

Campus	Teach	Learn	Data
New York	160	190	70
London	86	70	70

```
# Python code:
from scipy import stats

# df = the data in the table above
stats.chi2_contingency(df) >>> (10.01, 0.0061, 2, ...)

# 10.01 is chi_square, 0.0061 is the p-value
# Since the p-value < 5%, the difference is significant.
# So we reject the null hypothesis.
```

4 A/B Testing

A/B testing is a common practical framework to test if two versions of a scenario lead to different outcomes. A/B tests are a modern business take on the idea of Randomized Controlled Trials. RCTs are experiments where participants are randomly assigned either a Control or a Treatment group. For instance, in clinical trials, the Control is usually some kind of a placebo pill, whereas the Treatment is the real drug. A/B testing is used widely in UX/UI design to measure the impact of design tweaks, or for selecting the most effective marketing campaigns.

Some examples of questions that we could decide using A/B testing involve:

- What message prompts more reviews from our users?
- Given two color schemes for a website, which would engage our users more?
- Given two (or more) possible ad placements, which results in higher click-through rates?

A/B testing consists of the following main steps:

1. Define the business question and performance metric.
2. Setup the experiment by selecting appropriate sample sizes and randomized groups.
3. Compare group performance and determine if the difference is statistically significant (with hypothesis testing).

Note: A/B testing refers to the design, execution, and evaluation of the experiment, not just the hypothesis test in Step 3. It includes how we deploy our different scenarios, how we collect the measurements, and the hypothesis testing as the final phase.

Example 1

We want to see if adding a certain feature on our main webpage leads to more user interaction. We calculate mean time spent on the webpage for each group.

Current	New
4.5 s	6.3 s

It seems that more time is spent on the new web page. We can now check if this is by accident or statistically significant.

$$H_0 : \text{The mean of group1} = \text{mean of group2}$$
$$H_1 : \text{The mean of group1} \neq \text{mean of group2}$$

We perform a t-test (two-sample independent):

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

```
from scipy import stats
set_1 = df[df['group']=='new'][time_on_page']
set_2␣=␣df[df['group']=='current'][time_on_page']

t_stat , p_value = stats.ttest_ind(set_1 , set_2)
```

Then we interpret the results as usual.

Example 2

We have two candidate email send-outs for testing new grid designs, A and B. We want to see if the difference in average amounts of customer response is significant.

$$\text{Group A responses: } = [1, 1, 0, 1, 0, 0, 0, 1, 0, ...]$$
$$\text{Group B responses: } = [0, 1, 1, 1, 1, 0, 0, 0, 1, ...]$$

Suppose the average conversion rates are as follows:

$$P_A := \bar{X}_A = 16$$
$$P_B := \bar{X}_B = 24.2$$

We want to see if the difference in average conversion rate is statistically significant. We use a *proportions z-test*[1]. Suppose P_A and P_B are the conversion rates for groups A and B, respectively.

$$H_0 : P_A - P_B = 0$$
$$H_1 : \neg H_0$$

[1] Note that t-test and proportions z-test are different. *Proportions z-test:* This test is used to determine if there is a significant difference between the proportions of two groups. It's typically used when dealing with categorical data and when the sample sizes are large enough for the Central Limit Theorem to apply, which allows the normal approximation to the binomial distribution. *t-test:* This test is used to compare the means of two groups. It's commonly used when the sample sizes are small or when the population variances are unknown. There are different types of t-tests, such as the independent t-test (comparing means of two independent groups), paired t-test (comparing means of the same group at different times), and one-sample t-test (comparing the sample mean to a known value).

```
# Python code for t-test for email response rates
from statsmodels.stats.proportions import
counts = [group_A.sum(), group_B.sum()]
count_obs = [group_A.shape[0], group_B.shape[0]]

proportions_ztest
proportions_ztest(counts, count_obs) >>> (-2, -.0031)
```

Part III
Linear Models

5 Linear Regression

5.1 Linear Model Evaluation

We can always fit and interpret a linear model, but how do we know if our model fits well? We can get a sense of this using the R^2 value, which is a measure of how much of the total variance in our dependent variable is explained by our model. The R^2 value ranges from 0 to 1, where a value closer to 1 indicates that a large proportion of the variance in the dependent variable is predictable from the independent variables.

However, R^2 alone does not provide a complete picture. It's also important to look at adjusted R^2, especially in multiple regression, as it adjusts for the number of predictors in the model, preventing overestimation of model fit due to many variables. Other metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are also useful to evaluate the performance of a linear regression model.

Additionally, visual diagnostics such as residual plots, Q-Q plots, and leverage plots help assess the assumptions of linear regression and identify potential outliers or influential points.

5.2 Equation for the Line of Best Fit

The equation for the line of best fit in a multiple linear regression model is given by:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Here, θ_0 is the y-intercept, $\theta_1, \theta_2, \ldots, \theta_n$ are the coefficients of the independent variables $x_1, x_2, \ldots, x_n$, respectively. These coefficients represent the change in the dependent variable for a one-unit change in the corresponding independent variable, holding all other variables constant.

The coefficients θ are estimated using methods like Ordinary Least Squares (OLS), which minimizes the sum of the squared residuals (the differences between the observed and predicted values). The interpretation of these coefficients is crucial for understanding the relationship between the independent and dependent variables.

5.3 Cost Function

The cost function, often referred to as the Mean Squared Error (MSE), is used to measure the difference between predicted values and actual values. It is given by:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

where m is the number of training examples, $h_\theta(x^{(i)})$ is the predicted value, and $y^{(i)}$ is the actual value. The cost function provides a single measure of the model's performance, and minimizing this function is the goal of the learning algorithm.

The factor $\frac{1}{2}$ is included for mathematical convenience when deriving the gradient descent equations, as it cancels out the 2 in the derivative of the squared term.

5.4 Gradient Descent

Gradient Descent is an optimization algorithm used to minimize the cost function. The parameters θ are updated iteratively in the direction that reduces the cost function the most. The update rule is:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

where α is the learning rate, which controls the step size of each iteration. The partial derivatives of the cost function are given by:

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

$$\frac{\partial J}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Gradient Descent can converge to a local minimum, which in the case of a convex cost function like MSE, is also the global minimum. Choosing an appropriate learning rate is crucial; too large a learning rate can cause the algorithm to diverge, while too small a rate can make the convergence process very slow.

5.5 Normal Equation

The normal equation provides a closed-form solution to linear regression, which can be computed without iterative optimization:

$$\theta = (X^T X)^{-1} X^T y$$

Here, X is the matrix of independent variables with each row representing a training example, y is the vector of dependent variables, and θ is the vector of coefficients. The normal equation directly computes the values of θ that minimize the cost function.

While the normal equation provides an exact solution, it is computationally expensive for large datasets because it involves matrix inversion, which has a time complexity of $O(n^3)$. Therefore, for large datasets, iterative methods like Gradient Descent are preferred.

5.6 Linear Regression Evaluation

5.6.1 Total Sum of Squares

The Total Sum of Squares (SS_{total}) measures the total variance in the dependent variable. It is defined as:

$$SS_{\text{total}} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

where $\bar{y}$ is the mean of the observed values y_i.

The Residual Sum of Squares (SS_{res}) measures the variance that is not explained by the model:

$$SS_{\text{res}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ are the predicted values.

The explained sum of squares (SS_{reg}) is the difference between SS_{total} and SS_{res}:

$$SS_{\text{reg}} = SS_{\text{total}} - SS_{\text{res}}$$

These metrics are fundamental in calculating the R^2 value and assessing the model's explanatory power.

5.6.2 R-Squared

The R^2 value, or coefficient of determination, is calculated as:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{total}}}$$

To help with the interpretation of R^2, let's consider the following examples:

1. **A "perfect" model** – all values of our dependent variable lie exactly on our best fit line. In this case, $SS_{\text{res}} = 0.0$ and $R^2 = 1.0$.

2. **A less than perfect model** – some of our Y values deviate a bit from the best fit line. Here, $SS_{\text{res}} > 0.0$ and $R^2 < 1.0$ but greater than zero.[2]

3. **The model has equal SS_{res} and SS_{total}** – the model is no better at predicting Y than simply guessing the mean of Y. Either our model is not properly fit or our independent variables do not have a linear relationship with Y due to nonlinearity or a lack of correlation between the two. In either case, $R^2 = 0.0$.

4. **An** $R^2 < 0.0$ implies that our model is **worse** at predicting Y than simply guessing the mean of Y. Perhaps our model is some random line completely removed from our data or there has been some fitting error.

A quick think will reveal that this value can never exceed 1.0 because neither SS_{res} nor SS_{total} can be negative.

Fun fact: in a model with a single independent variable, the R^2 is simply the square of the correlation coefficient ρ. In a more complicated model, we usually try to separate out the sources of variability in the data and analyze each individually.

5.7 Hypothesis Testing for Linear Regression

When we fit a linear regression model, our goal is to model the linear relationship of age and weight for all children in the population. Since we can't build a model on the entire population, we build a model on a sample of data to approximate the patterns in the underlying population.

Hence, we can perform some statistical tests to assess whether the coefficient estimates we have calculated based on the sample apply to the population. Specifically, for all β_i values, we can perform the following test:

$$H_0 : \beta_i = 0 \quad \text{vs} \quad H_a : \beta_i \neq 0$$

Possible conclusions:

[2] It's important to note that a high R^2 value does not necessarily indicate that the model is good; it could be due to overfitting. Therefore, R^2 should be interpreted in conjunction with other diagnostic measures and the context of the data.

- *Reject the null hypothesis* - The population model parameter β_i is not 0, so the independent variable associated with it is relevant in predicting the dependent variable.
- *Fail to reject the null hypothesis* - We do not have sufficient evidence to say that β_i is not 0. We *cannot* conclude that β_i is 0, but since we cannot say otherwise, it is common to conclude that the associated independent variable is not as relevant in predicting the dependent variable.

Why do we care if β_i is 0? If the coefficient in front of x_i in the equation is 0, x_i has no effect on y. No matter what you set the value x_i to, y will not change.

You can think of considering these hypothesis tests as: We only want to include variables that are certain to have an effect in predicting the dependent variable. If we are not sure (their β_i might be 0), we consider dropping them from the model.

Very conveniently, the p-values for all of these hypothesis tests are in the `P>|t|` column of the `statsmodels` summary.

5.8 Linear Regression Assumptions

Assumption 1: Linear Relationship Between Variables

Before fitting a model, it is crucial to identify whether a linear relationship exists between the independent and dependent variables. This can be checked using the following methods:

- **Visualize:** Create scatterplots between each independent variable and the dependent variable. If a straight line appears to adequately describe the relationship, this suggests the presence of a linear relationship.
- **Calculate:** Compute the correlation coefficients for each pairing of independent and dependent variables. Any non-zero values indicate the existence of a linear relationship.

Assumption 2: Independent and Identically Distributed (i.i.d)

To ensure the validity of regression results, the i.i.d. assumption must be verified:

- **Independence:** The independent variables should not be correlated. Correlation coefficients between them should be zero. Despite individual variables appearing uncorrelated, multicollinearity can still occur if a combination of variables correlates with another variable.
- **Identically Distributed:** Variables should follow the same statistical distribution during random sampling. Slight deviations from this assumption can still permit the fitting of linear regression models but may affect confidence in the model.

Multicollinearity: This occurs when one variable can be linearly predicted with high accuracy from others. Simple examples include the creation of a variable as a sum of two others. Such redundancy can lead to computational errors in regression. Detection methods include:

1. Analysis of variables that are insignificant in hypothesis tests but may still predict the dependent variable.
2. Variance Inflation Factor (VIF), which we will cover in detail in another session.

Assumption 3: Normally Distributed Residuals

After model fitting, check for normal distribution of residuals:

- **Visualize:** Plot a histogram of the residuals. It should resemble a normal distribution.
- **Shapiro-Wilk Test:** Though controversial, this test is commonly used to assess normality.
- **Q-Q Plot:** This plot compares the quantiles of residuals with the expected quantiles from a normal distribution.

Assumption 4: Homoscedasticity

Homoscedasticity means that the variance of error terms is consistent across all values of the independent variables. To verify this:

- Plot the residuals versus the fitted values. This plot should look like random noise, indicating that residuals are consistent across all values of predicted $\hat{Y}$.

5.9 Linear Regression in Statsmodels

```
import pandas as pd
import statsmodels.api as sm

kids = pd.DataFrame({
    ‘age’: [2.0, 6.0, 7.1],
    ‘weight’: [5.0, 20.0, 30.0]
})

X = kids[[‘age’]]
y = kids[‘weight’]

X_with_constant = sm.add_constant(X)
model = sm.OLS(y, X_with_constant)
results = model.fit()

print(results.summary())

>>>
```

```
Dep. Variable:           Weight          R−squared:            0.534
Model:                   OLS             Adj. R−squared:       0.476
Method:                  Least Squares F−statistic:            9.175
Date:                    Sun, 16 Jun 2024 Prob (F−statistic): 0.0163
Time:                    09:57:00         Log−Likelihood:      −25.395
No. Observations:        10              AIC:                  54.79
Df Residuals:            8               BIC:                  55.39
Df Model:                1               Covariance Type:      nonrobust

                 coef     std err      t       P>|t|    [0.025    0.975]
--------------------------------------------------------------------------
const           2.6198    4.788       0.547    0.599    −8.420    13.660
Age             3.0054    0.992       3.029    0.016     0.717     5.293

Omnibus:        1.130     Durbin−Watson:       2.295
```

```
Prob(Omnibus):   0.568     Jarque-Bera (JB):   0.695
Skew:            0.196     Prob(JB):           0.706
Kurtosis:        1.769     Cond. No.:          22.2
```

- Since p-value < 0.05, we reject the null hypothesis that $\beta_1 = 0$.
- This means there is likely a linear relationship between weight and age.
- The line of best fit for this data is now
$$y = 2.6198 + 3.0054x.$$
- A child that is 0 years old (newborn) has a predicted weight of 2.6 kg.
- For one unit increase in age, we expect the weight to go up by 3 kg.
- Important Point: We can conclude that the age variable can be used to predict weight to a statistically significant degree because we have rejected the null hypothesis that β_1 is zero. This is *not* the same concept as quantifying the error of our estimate of β_1; we are saying only that it is highly unlikely that this coefficient is zero given our data.

6 Logistic Regression

Logistic regression is a statistical method used to analyze a data set where the dependent variable is qualitative or discrete. This method is widely used in situations where the outcome to be predicted is binary, such as yes/no, pass/fail, win/lose, etc.

In order to model the relationship between the predictor variables and the outcome variable, the logistic regression model assumes that the relationship between the predictor variables and the outcome variable is linear on the logit scale. The logit function (log-odds) is given by the equation:

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n)}}$$

where P is the probability of the event occurring. This function maps the linear combination of the predictors to the [0,1] interval, making it suitable for modeling probabilities.

6.1 Cost Function

In logistic regression, we use the likelihood function to estimate the model parameters. The likelihood function is the product of the probabilities assigned to each observed outcome:

$$L(\beta) = \prod_{i=1}^{n} P(y_i|\beta) = \prod_{i=1}^{n} p(y_i)^{y_i}(1 - p(y_i))^{(1-y_i)}$$

where $p(y_i)$ is the predicted probability of the i-th observation being in the positive class, and y_i is the actual class label (0, 1) for the i-th observation.

To simplify computation, we take the natural logarithm of the likelihood function, known as the log-likelihood:

$$l(\beta) = \sum_{i=1}^{n} (y_i \ln(p(y_i)) + (1 - y_i)\ln(1 - p(y_i)))$$

The partial derivative of the log-likelihood with respect to each parameter β_j is used to update the coefficients during the gradient descent optimization:

$$\frac{\partial l(\beta)}{\partial \beta_j} = \sum_{i=1}^{n} (y_i - P(y_i))x_{ij}$$

We can update coefficients using gradient descent to maximize the log-likelihood.

6.2 Evaluating Models

Evaluating the performance of logistic regression models involves assessing how well the model predicts the actual class labels. The key metrics used for this purpose include accuracy, precision, recall, F1-score, and the area under the ROC curve. Each metric serves a specific purpose and provides insights into different aspects of model performance.

- **Accuracy:** This is the simplest and most intuitive performance measure. Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined. It is calculated as:

  $$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

  However, accuracy alone can be misleading, especially in datasets with an uneven class distribution, known as imbalanced datasets.

- **Precision:** Precision, also called the positive predictive value, is the proportion of positive identifications that were actually correct. It is a critical measure when the cost of false positives is high. Precision is calculated as:

  $$\text{Precision} = \frac{TP}{TP + FP}$$

  Precision helps us understand the accuracy of the positive predictions made by the model.

- **Recall (Sensitivity):** Recall is the ability of the model to find all the relevant cases within a dataset. It is especially important in situations where the cost of false negatives is high. Recall is calculated as:

  $$\text{Recall} = \frac{TP}{TP + FN}$$

  Recall provides insights into the model's ability to detect positive samples.

- **F1-Score:** The F1-score is the harmonic mean of precision and recall. It is a better measure than accuracy for models with imbalanced datasets. The F1-score is calculated as:

  $$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

  This score balances the concerns of precision and recall, supporting scenarios where both false positives and false negatives have significant costs.

- **Area Under the ROC Curve (AUC-ROC):** The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (Recall) against the false positive rate (1 - Specificity) at various threshold settings. The area under the curve (AUC) is a measure of the ability of the classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s.

Each of these metrics should be considered when evaluating the effectiveness of a logistic regression model to ensure it meets the specific criteria of the application. For instance, in medical diagnostics, high recall might be more desirable than high precision, while in spam detection, high precision might be more critical.

Furthermore, the trade-offs between precision and recall can be adjusted by altering the classification threshold. Lowering the threshold increases recall but reduces precision, whereas raising the threshold increases precision but reduces recall. This adjustment is crucial in applications where the costs of false positives and false negatives vary significantly.

6.3 Logistic Regression in Statsmodels

```
import statsmodels.api as sm

X = cr[['Hours Researched']]
y = cr['Hired']

X_with_constant = sm.add_constant(X)
mylogreg = sm.Logit(y, X_with_constant)
mylogreg_results = mylogreg.fit()

print(mylogreg_results.summary())

>>>
```

```
\begin{lstlisting}
Dep. Variable:          Hired
No. Observations:       25
Model:                  Logit
Df Residuals:           23
Method:                 MLE
Date:                   Tue, 17 May 2022
Pseudo R-squ.:           0.3059
Time:                   13:07:44
Log-Likelihood:         -11.678
converged:               True
LL-Null:                -16.825
Covariance Type:        nonrobust
LLR p-value:            0.001334

                 coef     std err       z        P>|z|      [0.025     0.975]
--------------------------------------------------------------------------------
const           -4.8223     2.077     -2.322     0.020     -8.893      -0.752
Hours Researched   2.3694   1.023   2.316   0.021   0.364   4.375
```

The logistic regression model in `statsmodels` can be used to fit and evaluate a logistic regression model. The summary output provides detailed information about the model, including the coefficients and their statistical significance.

The predicted probability function is given by:

$$S(X) = P(y|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1)}}$$

The odds ratio, which represents the change in odds of the outcome occurring for a one-unit change in the predictor variable, is calculated as:

$$\text{odds ratio} = \frac{P}{1 - P}$$

- For the intercept: odds ratio = $e^{4.81} \approx 122.36$
- So, for every hour of research put in, the odds of getting hired increase by $e^{2.36} \approx 10.69$.

6.4 Model Evaluation

		Predicted Label	
		0	1
True Label	0	85292	16
	1	59	76

If our goal is to identify as many of the frauds as possible and measure how well we achieved that goal, we need to look at the number of true positives versus the number of total frauds.

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{76}{76 + 59}$$

If our goal is to make sure that if a package is predicted as a bomb, it actually is a bomb because it is very costly to neutralize a bomb, we want to reduce the number of false positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

But here, it is even more important to make sure there are as few false negatives as possible because that is even more costly if a bomb goes off. So we want as high a recall as possible.

6.5 Threshold Adjustment

In logistic regression, the decision threshold is a critical parameter that determines the classification of probability outcomes into discrete class labels. The default threshold of 0.5 classifies probabilities greater than or equal to this value as positive (1) and less than this value as negative (0). Adjusting this threshold can significantly alter the model's performance metrics, particularly its precision and recall.

Lowering the threshold means that the model classifies instances as positive more liberally. This is beneficial when the cost of missing a positive case (false negative) is high but can result in an increase in false positives. Conversely, increasing the threshold results in a conservative model that only classifies an instance as positive if the evidence is strong, thereby reducing false positives but increasing the risk of false negatives.

The choice of threshold should be based on the specific requirements and costs associated with false positives and false negatives in the application context. It is often chosen by evaluating the trade-offs between metrics on a validation set or by using techniques like ROC curve analysis to select an optimal balance.

Example 1: Credit Card Fraud Detection

In the context of credit card fraud detection, the cost of missing a fraudulent transaction (false negative) can be very high, leading to significant financial losses. Therefore, a lower threshold might be preferred to ensure high recall. For example, setting the threshold at 0.1 could increase the model's sensitivity to potential fraud, flagging more transactions as suspicious. This approach prioritizes security over user convenience, accepting the higher false positive rate which might lead to blocking some legitimate transactions but ensures that fraud is likely detected.

In practice, while this might inconvenience some customers, the financial security it provides could justify the approach, especially for high-value transactions. Banks often manage customer relations by swiftly resolving false alarms, maintaining a balance between security and customer satisfaction.

Example 2: Email Spam Filtering

Contrastingly, in email spam filtering, where the cost of mistakenly classifying an important email as spam (false positive) is high, a higher threshold may be preferable. This conservative approach ensures that only emails with a high probability of being spam are filtered out.

For instance, setting the threshold at 0.9 means the model is very confident before classifying an email as spam, reducing interruptions to important communications. The trade-off here is that some spam emails might slip through into the inbox (lower recall), but crucial emails are less likely to be misclassified as spam, enhancing user experience by ensuring that important emails remain in the inbox.

This strategy is particularly advantageous in business environments where missing a legitimate email could have significant repercussions, such as missing critical deadlines or failing to respond to important client communications.