hon-for-data-24-hypothesis-testing

April 27, 2023

1 Python for Data 24: Hypothesis Testing

back to index

Point estimates and confidence intervals are basic inference tools that act as the foundation for another inference technique: statistical hypothesis testing. Statistical hypothesis testing is a framework for determining whether observed data deviates from what is expected. Python's scipy.stats library contains an array of functions that make it easy to carry out hypothesis tests.

2 Hypothesis Testing Basics

Statistical hypothesis tests are based a statement called the null hypothesis that assumes nothing interesting is going on between whatever variables you are testing. The exact form of the null hypothesis varies from one type test to another: if you are testing whether groups differ, the null hypothesis states that the groups are the same. For instance, if you wanted to test whether the average age of voters in your home state differs from the national average, the null hypothesis would be that there is no difference between the average ages.

The purpose of a hypothesis test is to determine whether the null hypothesis is likely to be true given sample data. If there is little evidence against the null hypothesis given the data, you accept the null hypothesis. If the null hypothesis is unlikely given the data, you might reject the null in favor of the alternative hypothesis: that something interesting is going on. The exact form of the alternative hypothesis will depend on the specific test you are carrying out. Continuing with the example above, the alternative hypothesis would be that the average age of voters in your state does in fact differ from the national average.

Once you have the null and alternative hypothesis in hand, you choose a significance level (often denoted by the Greek letter .). The significance level is a probability threshold that determines when you reject the null hypothesis. After carrying out a test, if the probability of getting a result as extreme as the one you observe due to chance is lower than the significance level, you reject the null hypothesis in favor of the alternative. This probability of seeing a result as extreme or more extreme than the one observed is known as the p-value.

The T-test is a statistical test used to determine whether a numeric data sample of differs significantly from the population or whether two samples differ from one another.

3 One-Sample T-Test

A one-sample t-test checks whether a sample mean differs from the population mean. Let's create some dummy age data for the population of voters in the entire country and a sample of voters in Minnesota and test the whether the average age of voters Minnesota differs from the population:

[1]:

```
%matplotlib inline

import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import math
```

[ ]:

```
np.random.seed(6)

population_ages1 = stats.poisson.rvs(loc=18, mu=35, size=150000)
population_ages2 = stats.poisson.rvs(loc=18, mu=10, size=100000)
population_ages = np.concatenate((population_ages1, population_ages2))

minnesota_ages1 = stats.poisson.rvs(loc=18, mu=30, size=30)
minnesota_ages2 = stats.poisson.rvs(loc=18, mu=10, size=20)
minnesota_ages = np.concatenate((minnesota_ages1, minnesota_ages2))

print( population_ages.mean() )
print( minnesota_ages.mean() )
```

Notice that we used a slightly different combination of distributions to generate the sample data for Minnesota, so we know that the two means are different. Let's conduct a t-test at a 95% confidence level and see if it correctly rejects the null hypothesis that the sample comes from the same distribution as the population. To conduct a one sample t-test, we can the stats.ttest_1samp() function:

[ ]:

```
stats.ttest_1samp(a = minnesota_ages,               # Sample data
                 popmean = population_ages.mean())  # Pop mean
```

The test result shows the test statistic "t" is equal to -2.574. This test statistic tells us how much the sample mean deviates from the null hypothesis. If the t-statistic lies outside the quantiles of the t-distribution corresponding to our confidence level and degrees of freedom, we reject the null hypothesis. We can check the quantiles with stats.t.ppf():

[ ]:

```
stats.t.ppf(q=0.025,  # Quantile to check
            df=49)  # Degrees of freedom
```

[ ]:

```
stats.t.ppf(q=0.975,  # Quantile to check
            df=49)  # Degrees of freedom
```

We can calculate the chances of seeing a result as extreme as the one we observed (known as the p-value) by passing the t-statistic in as the quantile to the stats.t.cdf() function:

[ ]:

```
stats.t.cdf(x= -2.5742,      # T-test statistic
               df= 49) * 2   # Multiply by two for two tailed test *
```

Note: The alternative hypothesis we are checking is whether the sample mean differs (is not equal to) the population mean. Since the sample could differ in either the positive or negative direction we multiply the by two.

Notice this value is the same as the p-value listed in the original t-test output. A p-value of 0.01311 means we'd expect to see data as extreme as our sample due to chance about 1.3% of the time if the null hypothesis was true. In this case, the p-value is lower than our significance level (equal to 1-conf.level or 0.05) so we should reject the null hypothesis. If we were to construct a 95% confidence interval for the sample it would not capture population mean of 43:

[ ]:

```
sigma = minnesota_ages.std()/math.sqrt(50)  # Sample stdev/sample size

stats.t.interval(0.95,                        # Confidence level
                 df = 49,                     # Degrees of freedom
                 loc = minnesota_ages.mean(), # Sample mean
                 scale= sigma)                # Standard dev estimate
```

On the other hand, since there is a 1.3% chance of seeing a result this extreme due to chance, it is not significant at the 99% confidence level. This means if we were to construct a 99% confidence interval, it would capture the population mean:

[ ]:

```
stats.t.interval(alpha = 0.99,                # Confidence level
                 df = 49,                     # Degrees of freedom
                 loc = minnesota_ages.mean(), # Sample mean
                 scale= sigma)                # Standard dev estimate
```

With a higher confidence level, we construct a wider confidence interval and increase the chances that it captures to true mean, thus making it less likely that we'll reject the null hypothesis. In this case, the p-value of 0.013 is greater than our significance level of 0.01 and we fail to reject the null hypothesis.

4 Two-Sample T-Test

A two-sample t-test investigates whether the means of two independent data samples differ from one another. In a two-sample test, the null hypothesis is that the means of both groups are the same. Unlike the one sample-test where we test against a known population parameter, the two sample test only involves sample means. You can conduct a two-sample t-test by passing with the stats.ttest_ind() function. Let's generate a sample of voter age data for Wisconsin and test it against the sample we made earlier:

[ ]:

```
np.random.seed(12)
wisconsin_ages1 = stats.poisson.rvs(loc=18, mu=33, size=30)
wisconsin_ages2 = stats.poisson.rvs(loc=18, mu=13, size=20)
wisconsin_ages = np.concatenate((wisconsin_ages1, wisconsin_ages2))
```

```python
print( wisconsin_ages.mean() )
```

[ ]:

```python
stats.ttest_ind(a= minnesota_ages,
                b= wisconsin_ages,
                equal_var=False)    # Assume samples have equal variance?
```

The test yields a p-value of 0.0907, which means there is a 9% chance we'd see sample data this far apart if the two groups tested are actually identical. If we were using a 95% confidence level we would fail to reject the null hypothesis, since the p-value is greater than the corresponding significance level of 5%.

5 Paired T-Test

The basic two sample t-test is designed for testing differences between independent groups. In some cases, you might be interested in testing differences between samples of the same group at different points in time. For instance, a hospital might want to test whether a weight-loss drug works by checking the weights of the same group patients before and after treatment. A paired t-test lets you check whether the means of samples from the same group differ.

We can conduct a paired t-test using the scipy function stats.ttest_rel(). Let's generate some dummy patient weight data and do a paired t-test:

[ ]:

```python
np.random.seed(11)

before= stats.norm.rvs(scale=30, loc=250, size=100)

after = before + stats.norm.rvs(scale=5, loc=-1.25, size=100)

weight_df = pd.DataFrame({"weight_before":before,
                          "weight_after":after,
                          "weight_change":after-before})

weight_df.describe()             # Check a summary of the data
```

The summary shows that patients lost about 1.23 pounds on average after treatment. Let's conduct a paired t-test to see whether this difference is significant at a 95% confidence level:

[ ]:

```python
stats.ttest_rel(a = before,
                b = after)
```

6 Type I and Type II Error

The result of a statistical hypothesis test and the corresponding decision of whether to reject or accept the null hypothesis is not infallible. A test provides evidence for or against the null hypothesis and then you decide whether to accept or reject it based on that evidence, but the evidence may lack the strength to arrive at the correct conclusion. Incorrect conclusions made from hypothesis tests fall in one of two categories: type I error and type II error.

Type I error describes a situation where you reject the null hypothesis when it is actually true. This type of error is also known as a “false positive” or “false hit”. The type 1 error rate is equal to the significance level , so setting a higher confidence level (and therefore lower alpha) reduces the chances of getting a false positive.

Type II error describes a situation where you fail to reject the null hypothesis when it is actually false. Type II error is also known as a “false negative” or “miss”. The higher your confidence level, the more likely you are to make a type II error.

Let’s investigate these errors with a plot:

[ ]:

```
plt.figure(figsize=(12,10))


plt.fill_between(x=np.arange(-4,-2,0.01),
                 y1= stats.norm.pdf(np.arange(-4,-2,0.01)) ,
                 facecolor='red',
                 alpha=0.35)

plt.fill_between(x=np.arange(-2,2,0.01),
                 y1= stats.norm.pdf(np.arange(-2,2,0.01)) ,
                 facecolor='grey',
                 alpha=0.35)

plt.fill_between(x=np.arange(2,4,0.01),
                 y1= stats.norm.pdf(np.arange(2,4,0.01)) ,
                 facecolor='red',
                 alpha=0.5)

plt.fill_between(x=np.arange(-4,-2,0.01),
                 y1= stats.norm.pdf(np.arange(-4,-2,0.01),loc=3, scale=2) ,
                 facecolor='grey',
                 alpha=0.35)

plt.fill_between(x=np.arange(-2,2,0.01),
                 y1= stats.norm.pdf(np.arange(-2,2,0.01),loc=3, scale=2) ,
                 facecolor='blue',
                 alpha=0.35)

plt.fill_between(x=np.arange(2,10,0.01),
                 y1= stats.norm.pdf(np.arange(2,10,0.01),loc=3, scale=2),
                 facecolor='grey',
                 alpha=0.35)

plt.text(x=-0.8, y=0.15, s= "Null Hypothesis")
plt.text(x=2.5, y=0.13, s= "Alternative")
plt.text(x=2.1, y=0.01, s= "Type 1 Error")
plt.text(x=-3.2, y=0.01, s= "Type 1 Error")
plt.text(x=0, y=0.02, s= "Type 2 Error");
```

In the plot above, the red areas indicate type I errors assuming the alternative hypothesis is not different from the null for a two-sided test with a 95% confidence level.

The blue area represents type II errors that occur when the alternative hypothesis is different from the null, as shown by the distribution on the right. Note that the Type II error rate is the area under the alternative distribution within the quantiles determined by the null distribution and the confidence level. We can calculate the type II error rate for the distributions above as follows:

[ ]:

```
lower_quantile = stats.norm.ppf(0.025)  # Lower cutoff value
upper_quantile = stats.norm.ppf(0.975)  # Upper cutoff value

# Area under alternative, to the left the lower cutoff value
low = stats.norm.cdf(lower_quantile,
                     loc=3,
                     scale=2)

# Area under alternative, to the left the upper cutoff value
high = stats.norm.cdf(upper_quantile,
                      loc=3,
                      scale=2)

# Area under the alternative, between the cutoffs (Type II error)
high-low
```

With the normal distributions above, we'd fail to reject the null hypothesis about 30% of the time because the distributions are close enough together that they have significant overlap.

7 Statistical Power

The power of a statistical test is the probability that the test rejects the null hypothesis when the alternative is actually different from the null. In other words, power is the probability that the test detects that there is something interesting going on when there actually *is* something interesting going on. Power is equal to one minus the type II error rate. The power of a statistical test is influenced by:

1. The significance level chosen for the test.
2. The sample size.
3. The effect size of the test.

When choosing a significance level for a test, there is a trade-off between type I and type II error. A low significance level, such as 0.01 makes a test unlikely to have type I errors (false positives), but more likely to have type II errors (false negatives) than a test with larger value of the significance level . A common convention is that a statistical tests should have a power of at least 0.8.

A larger sample size reduces the uncertainty of the point estimate, causing the sample distribution to narrow, resulting in lower type II error rates and increased power.

Effect size is a general term that describes a numeric measure of the size of some phenomenon.

There are many different effect size measurements that arise in different contexts. In the context of the T-test, a simple effect size is the difference between the means of the samples. This number can be standardized by dividing by the standard deviation of the population or the pooled standard deviation of the samples. This puts the size of the effect in terms of standard deviations, so a standardized effect size of 0.5 would be interpreted as one sample mean being 0.5 standard deviations from another (in general 0.5 is considered a "large" effect size).

Since statistical power, the significance level, the effect size and the sample size are related, it is possible to calculate any one of them for given values of the other three. This can be an important part of the process of designing a hypothesis test and analyzing results. For instance, if you want to conduct a test with a given significance level (say the standard 0.05) and power (say the standard 0.8) and you are interested in a given effect size (say 0.5 for standardized difference between sample means), you could use that information to determine how large of a sample size you need.

In python, the statsmodels library contains functions to solve for any one parameter of the power of T-tests. Use statsmodels.stats.power.tt_solve_power for one sample t-tests and statsmodels.stats.power.tt_ind_solve_power for a two sample t-test. Let's check the sample size we should use need to use given the standard parameter values above for a one sample t-test:

[ ]:

```
from statsmodels.stats.power import tt_solve_power

tt_solve_power(effect_size = 0.5,
               alpha = 0.05,
               power = 0.8)
```

In this case, we would want a sample size of at least 34 to make a study with the desired power and signifiance level capable of detecting a large effect size.

8 Wrap Up

The t-test is a powerful tool for investigating the differences between sample and population means. T-tests operate on numeric variables; in the next lesson, we'll discuss statistical tests for categorical variables.

1.What is Null hypothesis?

Ans: In statistics, a null hypothesis is a statement that there is no significant difference or relationship between two or more groups or variables being compared. It is typically denoted by the symbol H0.

The null hypothesis is a starting point for statistical hypothesis testing, where a researcher proposes an alternative hypothesis (denoted by H1 or Ha) and tests whether there is sufficient evidence to reject the null hypothesis in favor of the alternative hypothesis.

For example, suppose a researcher wants to test whether there is a significant difference in the mean weight of two groups of people. The null hypothesis would be that there is no significant difference in the mean weight between the two groups, while the alternative hypothesis would be that there is a significant difference in the mean weight between the two groups.

The null hypothesis is important because it provides a reference point for statistical analysis and helps to ensure that any observed differences or relationships are not due to chance or random variation. By testing the null hypothesis, researchers can determine whether their results are statistically significant and provide evidence to support their research hypotheses.

2.What is alternate hypothesis?

Ans: In statistics, the alternative hypothesis (also known as the research hypothesis or Ha) is a statement that contradicts or opposes the null hypothesis (H0). It represents the hypothesis that the researcher is trying to prove or find evidence for through statistical hypothesis testing.

The alternative hypothesis is typically formulated based on prior research, theoretical considerations, or observations of the data. It is usually denoted by the symbol Ha and represents a range of possible values or outcomes that are different from what is expected under the null hypothesis.

For example, if a researcher is conducting a study to test the effectiveness of a new drug, the null hypothesis might be that the drug has no effect on the outcome being measured. The alternative hypothesis, in this case, would be that the drug does have an effect on the outcome being measured, either positive or negative.

In statistical hypothesis testing, the alternative hypothesis is used to determine whether there is sufficient evidence to reject the null hypothesis in favor of the alternative hypothesis. If the evidence supports the alternative hypothesis, then the researcher can conclude that there is a statistically significant relationship or difference between the variables being tested.

3.Define Type 1 and Type 2 errors.

Ans: Type 1 and Type 2 errors are two types of errors that can occur in hypothesis testing in statistics.

Type 1 error, also known as a false positive, occurs when the null hypothesis is rejected even though it is actually true. This means that the researcher concludes that there is a significant relationship or difference between the variables being tested when there is actually no such relationship or difference. The probability of making a Type 1 error is denoted by the symbol alpha ( ) and is typically set at 0.05 or 0.01.

Type 2 error, also known as a false negative, occurs when the null hypothesis is not rejected even though it is actually false. This means that the researcher concludes that there is no significant relationship or difference between the variables being tested when there is actually a relationship or difference. The probability of making a Type 2 error is denoted by the symbol beta ( ) and is influenced by factors such as the sample size, effect size, and level of significance.

Both Type 1 and Type 2 errors can have important implications for statistical inference and decision making. A Type 1 error can lead to false conclusions and wasted resources, while a Type 2 error can result in missed opportunities and incorrect decisions. The goal of hypothesis testing is to balance the risks of these two types of errors by selecting an appropriate level of significance and sample size, and by interpreting the results of the test in context of the research question and study design.

9 Next Lesson: Python for Data 25: Chi-Squared Tests

back to index

python-for-data-26-anova

April 27, 2023

1 Python for Data 26: ANOVA

back to index

In lesson 24 we introduced the t-test for checking whether the means of two groups differ. The t-test works well when dealing with two groups, but sometimes we want to compare more than two groups at the same time. For example, if we wanted to test whether voter age differs based on some categorical variable like race, we have to compare the means of each level or group the variable. We could carry out a separate t-test for each pair of groups, but when you conduct many tests you increase the chances of false positives. The analysis of variance or ANOVA is a statistical inference test that lets you compare multiple groups at the same time.

2 One-Way ANOVA

The one-way ANOVA tests whether the mean of some numeric variable differs across the levels of one categorical variable. It essentially answers the question: do any of the group means differ from one another? We won't get into the details of carrying out an ANOVA by hand as it involves more calculations than the t-test, but the process is similar: you go through several calculations to arrive at a test statistic and then you compare the test statistic to a critical value based on a probability distribution. In the case of the ANOVA, you use the "f-distribution".

The scipy library has a function for carrying out one-way ANOVA tests called scipy.stats.f_oneway(). Let's generate some fake voter age and demographic data and use the ANOVA to compare average ages across the groups:

[1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
```

[ ]:

```
np.random.seed(12)

races =   ["asian","black","hispanic","other","white"]

# Generate random data
voter_race = np.random.choice(a= races,
                              p = [0.05, 0.15 ,0.25, 0.05, 0.5],
                              size=1000)
```

```python
voter_age = stats.poisson.rvs(loc=18,
                              mu=30,
                              size=1000)

# Group age data by race
voter_frame = pd.DataFrame({"race":voter_race,"age":voter_age})
groups = voter_frame.groupby("race").groups

# Etract individual groups
asian = voter_age[groups["asian"]]
black = voter_age[groups["black"]]
hispanic = voter_age[groups["hispanic"]]
other = voter_age[groups["other"]]
white = voter_age[groups["white"]]

# Perform the ANOVA
stats.f_oneway(asian, black, hispanic, other, white)
```

The test output yields an F-statistic of 1.774 and a p-value of 0.1317, indicating that there is no significant difference between the means of each group.

Another way to carry out an ANOVA test is to use the statsmodels library, which allows you to specify a model with a formula syntax that mirrors that used by the R programming language. R users may find this method more familiar:

[ ]:

```python
import statsmodels.api as sm
from statsmodels.formula.api import ols

model = ols('age ~ race',                 # Model formula
            data = voter_frame).fit()

anova_result = sm.stats.anova_lm(model, typ=2)
print (anova_result)
```

As you can see, the statsmodels method produced the same F statistic and P-value (listed as PR(<F)) as the stats.f_oneway method.

Now let's make new age data where the group means do differ and run a second ANOVA:

[ ]:

```python
np.random.seed(12)

# Generate random data
voter_race = np.random.choice(a= races,
                              p = [0.05, 0.15 ,0.25, 0.05, 0.5],
                              size=1000)

# Use a different distribution for white ages
white_ages = stats.poisson.rvs(loc=18,
                              mu=32,
                              size=1000)

voter_age = stats.poisson.rvs(loc=18,
                              mu=30,
                              size=1000)

voter_age = np.where(voter_race=="white", white_ages, voter_age)

# Group age data by race
voter_frame = pd.DataFrame({"race":voter_race,"age":voter_age})
groups = voter_frame.groupby("race").groups

# Extract individual groups
asian = voter_age[groups["asian"]]
black = voter_age[groups["black"]]
hispanic = voter_age[groups["hispanic"]]
other = voter_age[groups["other"]]
white = voter_age[groups["white"]]

# Perform the ANOVA
stats.f_oneway(asian, black, hispanic, other, white)
```

[ ]:

```
# Alternate method
model = ols('age ~ race',                 # Model formula
            data = voter_frame).fit()

anova_result = sm.stats.anova_lm(model, typ=2)
print (anova_result)
```

The test result suggests the groups don't have the same sample means in this case, since the p-value is significant at a 99% confidence level. We know that it is the white voters who differ because we set it up that way in the code, but when testing real data, you may not know which group(s) caused the test to throw a positive result. To check which groups differ after getting a positive ANOVA result, you can perform a follow up test or "post-hoc test".

One post-hoc test is to perform a separate t-test for each pair of groups. You can perform a t-test between all pairs using by running each pair through the stats.ttest_ind() we covered in the lesson on t-tests:

[ ]:

```
# Get all race pairs
race_pairs = []

for race1 in range(4):
    for race2  in range(race1+1,5):
        race_pairs.append((races[race1], races[race2]))

# Conduct t-test on each pair
for race1, race2 in race_pairs:
    print(race1, race2)
    print(stats.ttest_ind(voter_age[groups[race1]],
                          voter_age[groups[race2]]))
```

The p-values for each pairwise t-test suggest mean of white voters is likely different from the other groups, since the p-values for each t-test involving the white group is below 0.05. Using unadjusted pairwise t-tests can overestimate significance, however, because the more comparisons you make, the more likely you are to come across an unlikely result due to chance. We can adjust for this multiple comparison problem by dividing the statistical significance level by the number of comparisons made. In this case, if we were looking for a significance level of 5%, we'd be looking for p-values of 0.05/10 = 0.005 or less. This simple adjustment for multiple comparisons is known as the Bonferroni correction.

The Bonferroni correction is a conservative approach to account for the multiple comparisons problem that may end up rejecting results that are actually significant. Another common post hoc-test is Tukey's test. You can carry out Tukey's test using the pairwise_tukeyhsd() function in the statsmodels.stats.multicomp library:

[ ]:

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd

tukey = pairwise_tukeyhsd(endog=voter_age,     # Data
                          groups=voter_race,   # Groups
                          alpha=0.05)          # Significance level

tukey.plot_simultaneous()    # Plot group confidence intervals
plt.vlines(x=49.57,ymin=-0.5,ymax=4.5, color="red")

tukey.summary()              # See test summary
```

The output of the Tukey test shows the average difference, a confidence interval as well as whether you should reject the null hypothesis for each pair of groups at the given significance level. In this case, the test suggests we reject the null hypothesis for 3 pairs, with each pair including the "white" category. This suggests the white group is likely different from the others. The 95% confidence interval plot reinforces the results visually: only 1 other group's confidence interval overlaps the white group's confidence interval.

3 Wrap Up

The ANOVA test lets us check whether a numeric response variable varies according to the levels of a categorical variable. Python's scipy library makes it easy to perform an ANOVA without diving too deep into the details of the procedure.

Next time, we'll move on from statistical inference to the final topic of this guide: predictive modeling.

4.When to go for Anova instead of t-test or chi square test?

Ans: In statistics, analysis of variance (ANOVA) is used to compare the means of three or more groups, while t-tests and chi-square tests are used to compare the means of two groups and the proportions of categorical variables, respectively. The choice of which test to use depends on the type of data being analyzed and the research question being addressed.

Generally, ANOVA is used when there are three or more independent groups being compared on a continuous outcome variable. It is typically used when the independent groups are categorical (e.g., treatment groups, age groups, or geographic regions) and the dependent variable is continuous (e.g., blood pressure, height, or income). ANOVA allows for the testing of multiple group means simultaneously, and can detect differences between groups that may not be detected by pairwise comparisons using t-tests.

On the other hand, t-tests are used to compare the means of two independent groups on a continuous outcome variable, while chi-square tests are used to compare the proportions of two or more categorical variables. T-tests are appropriate when the sample size is small, and the data is approximately normally distributed and the variances are equal. Chi-square tests are appropriate when the data consists of categorical variables and the sample size is sufficiently large.

In summary, the choice of test between ANOVA, t-tests, and chi-square tests depends on the research question, the type of data being analyzed, and the number of groups being compared. If the research question involves comparing the means of three or more groups on a continuous outcome variable, ANOVA may be the most appropriate test. If the research question involves comparing the means of two groups, t-tests may be more appropriate. If the research question involves comparing proportions of categorical variables, chi-square tests may be more appropriate.

5.What role does Anova play in a ML Project ppipeline?

Ans: In a machine learning (ML) project pipeline, ANOVA (Analysis of Variance) can be used to perform feature selection, which is the process of selecting the most important features (i.e., variables) in a dataset that contribute the most to the performance of a predictive model.

ANOVA can help identify features that have a significant effect on the outcome variable, and those that do not. By comparing the variance between groups (i.e., levels of a categorical variable) and within groups (i.e., variation within each group), ANOVA can determine whether there is a statistically significant difference in the means of the outcome variable across the different groups.

In the context of feature selection, ANOVA can be used to evaluate the significance of each feature with respect to the outcome variable, and to select the features with the highest F-scores, which indicate the strength of the relationship between the feature and the outcome variable. Features with high F-scores are more likely to be important predictors of the outcome variable, and can be used to train more accurate predictive models.

ANOVA can be used as a pre-processing step in the ML pipeline, where it helps to reduce the dimensionality of the feature space and remove irrelevant features, which can improve the performance of the predictive model and reduce overfitting.

4 Next Lesson: Python for Data 27: Linear Regression

back to index