Data Analysis Using Statistical Methods: Case Study of Categorizing the Species of Penguin

Preprint · July 2022

CITATIONS
0

READS
1,591

1 author:

Aishwarya Pawar
Stevens Institute of Technology
2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Data Analysis Using Statistical Methods: Case Study of Categorizing the Species of Penguin

Aishwarya Pawar

Department of Mathematical Sciences, Stevens Institute of Technology, Hoboken, NJ

Project Supervisor: Dr. Hadi Safari Katesari

Abstract

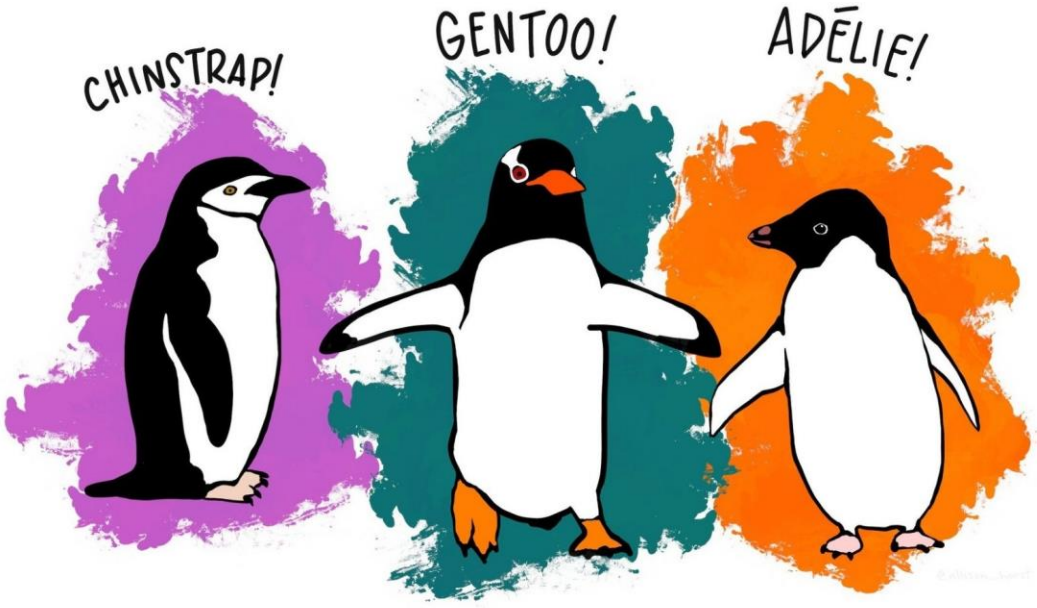
Statistical analysis is a scientific tool that helps collect and analyze large amounts of data to identify common patterns and trends to convert them into meaningful information. The goal is to use a dataset on which various statistical methods can be implemented so that one can get accurate predictions. The accuracy metric used in the project give us a 98.5% accuracy while categorizing the data.

INTRODUCTION AND MOTIVATION

The motivation of this project is to implement statistical methods encountered during analysis of real data. This includes and is not limited to F-test, Fisher information, ANOVA Analysis, Chi Squared Tests of Independence various distributions, categorical analysis of data, and regression models.

To achieve this, we will use the Palmer Archipelago (Antarctica) penguin dataset. First, we will implement various statistical methods to observe correlations and dependency between the various columns of the data. Then I will be implementing Forward and Backward elimination methods to select the best features from our data to fit a logistic regression model. We will test the accuracy of this logistic regression model on our test data.

MEET THE PENGUINS!



DATA DESCRIPTION:

Palmer Archipelago (Antarctica) penguin data were collected and made available by Dr. Kristen Gorman and the Palmer Station, Antarctica LTER, a member of the Long-Term Ecological Research Network.

penguins_size.csv: Simplified data from original penguin data sets.

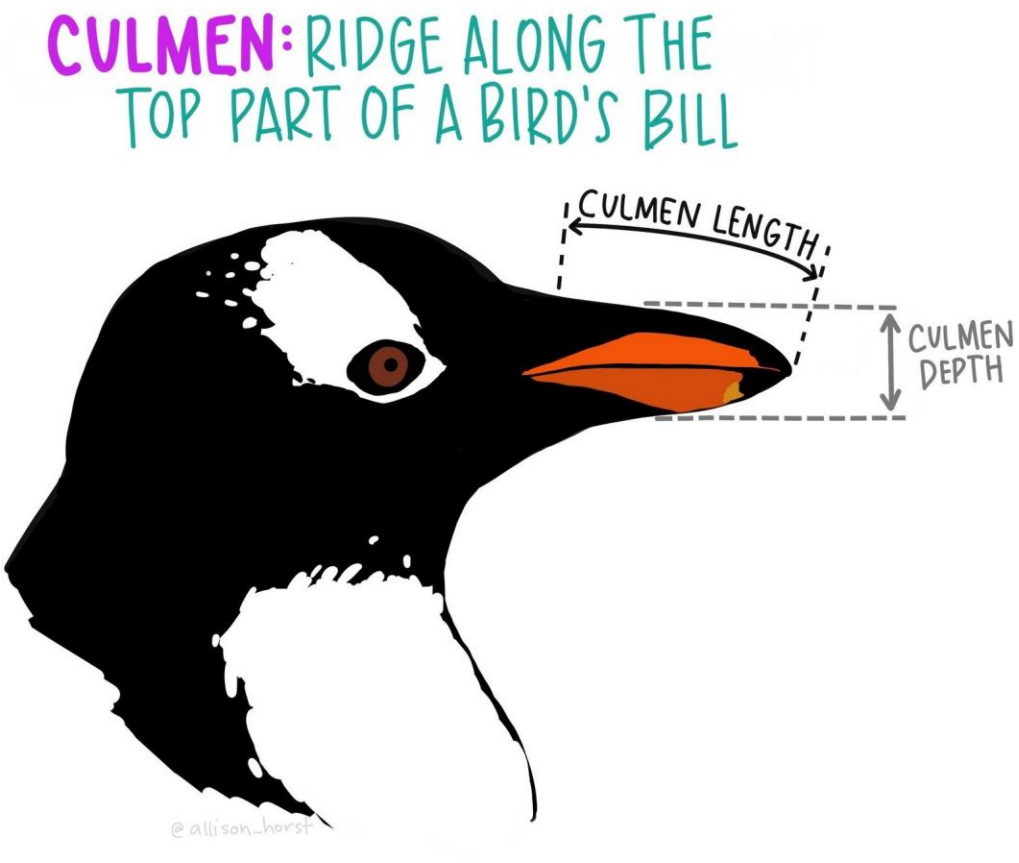
Contains variables:

Species: penguin species (Chinstrap, Adélie, or Gentoo) culmen_length_mm: culmen length (mm) culmen_depth_mm: culmen depth (mm) flipper_length_mm: flipper length (mm)

body_mass_g: body mass (g)

island: island name (Dream, Torgersen, or Biscoe) in the Palmer Archipelago (Antarctica) What are culmen length & depth?

The culmen is "the upper ridge of a bird's beak" (definition from Oxford Languages). For this penguin data, the culmen length and culmen depth are measured as shown below (thanks Kristen Gorman for clarifying!):



INITIAL THOUGHTS:

As we can see, the data set contains variables related to three different Penguin species. My aim for this project is to use these various features (Culmen length, Culmen depth, body mass, and sex) and build a logistic regression model to classify the species of Penguins depending on these features.

LOADING THE REQUIRED LIBRARIES

```
import numpy as np import pandas as pd

import matplotlib.pyplot as plt import seaborn as sns

from sklearn.preprocessing import LabelEncoder from scipy.stats import kruskal

from scipy.stats import mannwhitneyu

from statsmodels.stats.multicomp import pairwise_tukeyhsd

from sklearn.feature_selection import SelectKBest, chi2, f_classif from scipy.stats import chi2, chi2_contingency

from sklearn.model_selection import train_test_split from sklearn.metrics import r2_score

import joblib import sys
```

```
sys.modules['sklearn.externals.joblib'] = joblib

from mlxtend.feature_selection import SequentialFeatureSelector as SFS from sklearn.linear_model import LogisticRegression

from sklearn.preprocessing import StandardScaler from sklearn.metrics import accuracy_score

from sklearn.metrics import confusion_matrix

import warnings warnings.filterwarnings('ignore')

import pandas.util.testing as tm

filename = "/content/penguins_size.csv" df=pd.read_csv(filename)
```

Looking at the head/sample of the dataset, we can see that there are six columns that we can use to predict the species of the penguin. Since we want to predict the species of the Penguin, which is a categorical variable. Consequently, we'll fit a logistic regression model.

df.head()

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	MALE
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	FEMALE
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	FEMALE
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	FEMALE

df.describe()

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	342.000000	342.000000	342.000000	342.000000
mean	43.921930	17.151170	200.915205	4201.754386
std	5.459584	1.974793	14.061714	801.954536
min	32.100000	13.100000	172.000000	2700.000000
25%	39.225000	15.600000	190.000000	3550.000000
50%	44.450000	17.300000	197.000000	4050.000000
75%	48.500000	18.700000	213.000000	4750.000000
max	59.600000	21.500000	231.000000	6300.000000

DATA PRE-PROCESSING

CHECKING FOR NULL VALUE AND DROPPING THEM

df.isna().sum()

species	0
island	0
culmen_length_mm	2
culmen_depth_mm	2

flipper_length_mm	2
body_mass_g	2
sex	10

```
df = df.dropna()

df.isna().sum()
```

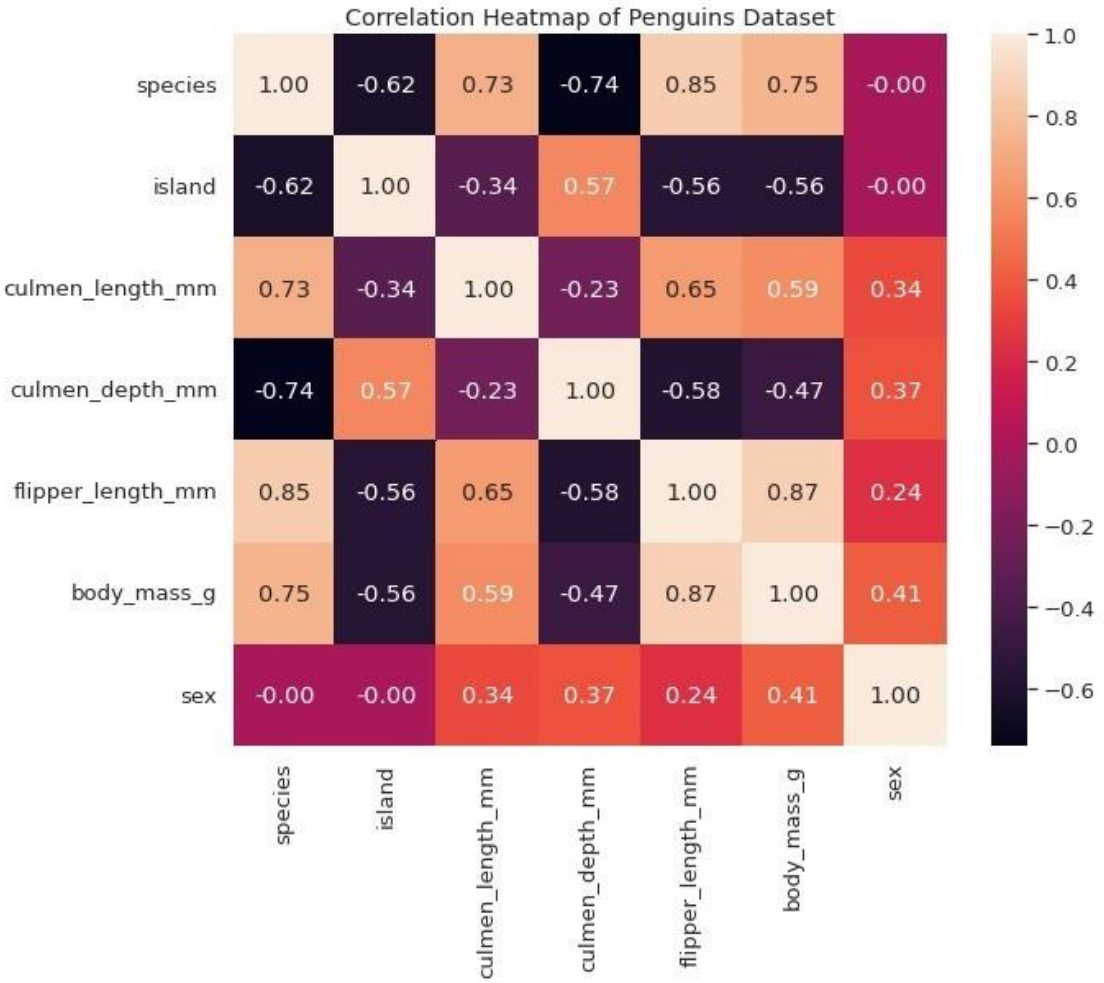
species 0

island 0
culmen_length_mm 0
culmen_depth_mm 0
flipper_length_mm 0
body_mass_g 0 sex 0
dtype: int64

Checking the correlation between all the variables

corr_matrix = df_cat.corr() corr_matrix

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
species	1.000000e+00	-0.623595	0.729262	-0.740803	0.851351	0.751020	-1.625802e-17
island	-6.235949e-01	1.000000	-0.337009	0.568885	-0.555759	-0.560518	-4.147089e-03
culmen_length_mm	7.292618e-01	-0.337009	1.000000	-0.228640	0.652126	0.589066	3.386764e-01
culmen_depth_mm	-7.408034e-01	0.568885	-0.228640	1.000000	-0.578730	-0.472987	3.740342e-01
flipper_length_mm	8.513508e-01	-0.555759	0.652126	-0.578730	1.000000	0.873211	2.411210e-01
body_mass_g	7.510201e-01	-0.560518	0.589066	-0.472987	0.873211	1.000000	4.115305e-01
sex	-1.625802e-17	-0.004147	0.338676	0.374034	0.241121	0.411531	1.000000e+00



FINDING CORRELATIONS GREATER THAN .9 AND DROPING THEM

```
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape),k=1).astype(bool))

to_drop = [column for column in upper.columns if any(upper[column] > 0.9)]
# NONE OF THE VALUES ARE HIGHLY CORRELATED SO WE CAN USE ALL OF THEM FOR ANALYSIS
```

[]

NONPARAMETRIC TESTS

WILCOXON MANN WHITNEY

Nonparametric statistics are those methods that do not assume a specific distribution to the data. Since our samples are independent and we have an unknown distribution, we can use the Wilcoxon Ranksum test. We can also use the kW test.

Mann-Whitney test is used for comparing differences between two independent groups. It tests the hypothesis that if the two groups come from same population or have the same medians. It does not assume any specific distribution (such as normal distribution of samples) for calculating test statistics and p values.

null hypothesis: All data samples were drawn from the same distribution.

Alternative Hypothesis: A rejection of the null hypothesis indicates that there is enough evidence to suggest that one or more samples dominate another sample

```
stat, p = mannwhitneyu(df_cat.species, df_cat.sex) print('Statistics=%.3f, p=%.3f'
% (stat, p))
alpha = 0.05
if p > alpha:
        print('Same distribution (fail to reject H0)')
  else:
        print('Different distribution (reject H0)')
```

Statistics=35751.000, p=0.000

Different distribution (reject H0)

KURSKAL-WALLIS TEST (ONE WAY ANOVA)

If we are having more than two groups to analyze, we will use Kruskal-Wallis test.

```
stat, p = kruskal(df_cat.species, df_cat.sex, df_cat.body_mass_g, df_cat.island, df_cat.culmen_depth_mm,
df_cat.culmen_length_mm, df_cat.flipper_length_mm)
print('Statistics=%.3f, p=%.3f' % (stat, p))
```

```
alpha = 0.05
if p > alpha:
print('Same distributions (fail to reject H0)')
  else:
print('Different distributions (reject H0)')
```

Statistics=2176.630, p=0.000 Different distributions (reject H0)

We can see that the overall p-value from the ANOVA table is not significant ie. less than .05, so we have sufficient evidence to say that the mean values across each group are not equal.
However, this doesn't tell us which groups are different from each other. It simply tells us that not all of the group means are equal. In order to find out exactly which groups are different from each other, we must conduct a post hoc test.

Thus, we can perform Tukey's Test to determine exactly which group means are different.

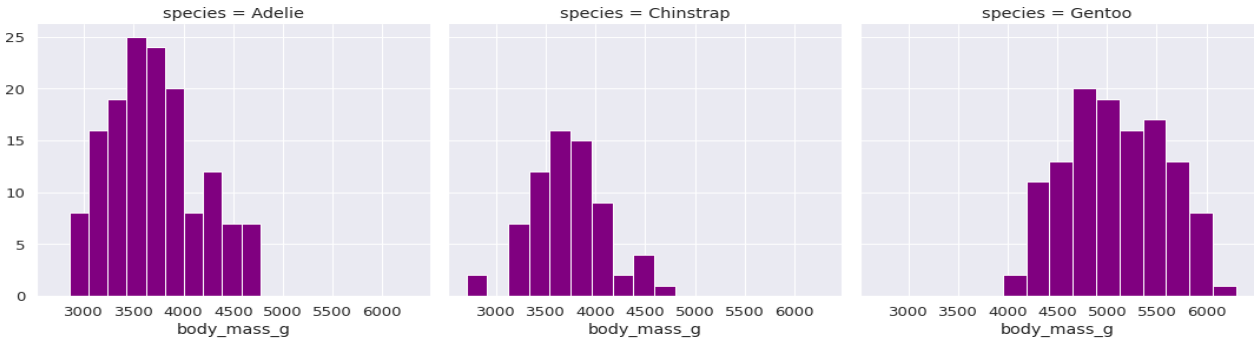
TUKEY'S HONEST SIGNIFICANT TEST

tukey = pairwise_tukeyhsd(endog=df['body_mass_g'], groups=df['species'], alpha=0.05)

Multiple Comparison of Means-Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
Adelie	Chinstrap	26.9239	0.9	-132.1708	186.0185	False
Adelie	Gentoo	1384.4606	0.001	1250.9385	1517.9828	True
Chinstrap	Gentoo	1357.5368	0.001	1193.0567	1522.0169	True

Adelie Chinstrap => False. Except for the body mass of these 2 species the rest all are different. Also, as we can see from the plot 1 and plot 2 below there is a significant overlap in the body mass of these two species.



tukey = pairwise_tukeyhsd(endog=df['culmen_length_mm'], groups=df['species'], alpha=0.05)

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
Adelie	Chinstrap	10.0099	0.001	8.9827	11.037	True
Adelie	Gentoo	8.7185	0.001	7.8564	9.5806	True

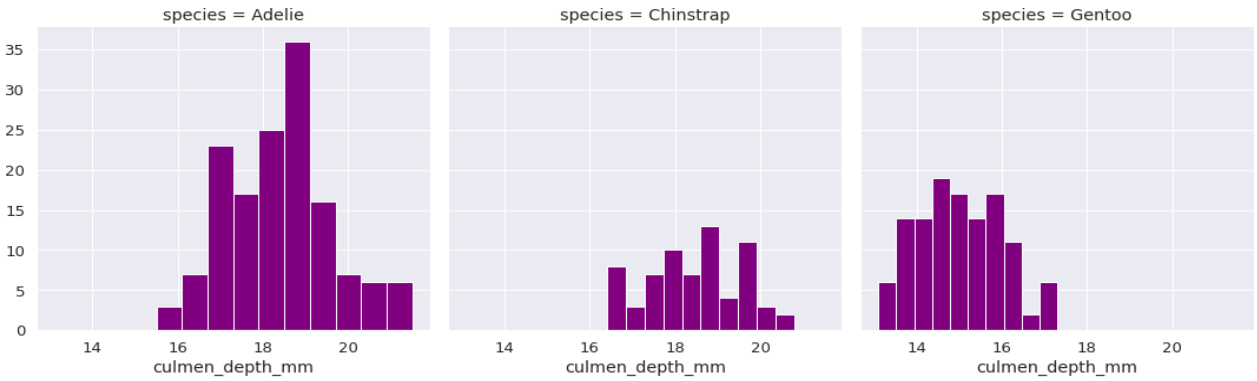
Chinstrap Gentoo -1.2913 0.0124 -2.3533 -0.2294 True

```
tukey = pairwise_tukeyhsd(endog=df['culmen_depth_mm'], groups=df['species'],
alpha=0.05)
```

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
Adelie	Chinstrap	0.0733	0.8896	-0.3147	0.4614	False
Adelie	Gentoo	-3.3448	0.001	-3.6704	-3.0191	True
Chinstrap	Gentoo	-3.4181	0.001	-3.8193	-3.0169	True

Adelie Chinstrap => False. Except for the culmen depth of these 2 species the rest all are different. Also, as we can see from the plot 1 and plot 2 below there is a significant overlap in the body mass of these two species.




```
tukey = pairwise_tukeyhsd(endog=df['flipper_length_mm'], groups=df['species'],
alpha=0.05)
```

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
Adelie	Chinstrap	5.7208	0.001	3.4178	8.0238	True
Adelie	Gentoo	27.1306	0.001	25.1978	29.0634	True
Chinstrap	Gentoo	21.4098	0.001	19.0288	23.7908	True

CATEGORICAL DATA ANALYSIS

CHI SQUARED TEST OF INDEPENDENCE

A Chi-Square Test of Independence is used to determine whether or not there is a significant association between two categorical variables. Checking if the categorical variables in the dataset, (sex and island) are related to the output categorical variable (species).

```
def ChiSqTest(data):

        stat, p, dof, expected = chi2_contingency(df_chi_test1)
        prob = 0.95
        critical = chi2.ppf(prob, dof)

    if abs(stat) >= critical:
                print('Dependent (reject H0)')
        else:
                print('Independent (fail to reject H0)')


 ChiSqTest(df_chi_test1)
```

Independent (fail to reject H0) 0.9999998092635104

H0: (null hypothesis) The two variables are independent.

H1: (alternative hypothesis) The two variables are not independent. Since the p-value (0.999) of the test is not less than 0.05, we fail to reject the null hypothesis. This means we do not have sufficient evidence to say that there is an association between island/sex and species. In other words, they are independent.

FEATURE SELECTION

SELECTING BEST FEATURES FOR FITTING THE LOGISTIC REGRESSION MODEL

Fisher Score (chi-square implementation)

It computes chi-squared stats between each non-negative feature and class.

This score can be used to evaluate categorical variables in a classification task. It compares the observed distribution of the different classes of target Y among the different categories of the feature, against the expected distribution of the target classes, regardless of the feature categories. We will use this to select the 4 best features based on Fisher score.

```
df_cat = df_cat.astype(int) target =
df_cat.copy() train_data = df_cat.copy()
target = df_cat.drop(['island', 'culmen_length_mm',
```

```
'culmen_depth_mm','flipper_length_mm', 'body_mass_g', 'sex'],1) train_data =
df_cat.drop(['species'],1)

# Select Features With Best F-Values
# Create an SelectKBest object to select features with four best ANOVA F-Values
fvalue_selector = SelectKBest(f_classif, k=4)

# Apply the SelectKBest object to the features and target
X_kbest = fvalue_selector.fit_transform(train_data,target)

print('Original number of features:', train_data.shape[1]) print('Reduced number of
features:', X_kbest.shape[1])
```

Original number of features: 6 Reduced number of features: 4

X_kbest

```
array([[      3       18,      181, 3750],
             9,
       [      3       17,      186, 3800],
             9,
       [      4       18,      195, 3250],
             0,
         ...,
         [    50,      15,      222, 5750],
         [    45,      14,      212, 5200],
         [    49,      16,      213, 5400]])
```

FORWARD SELECTION

Subset Selection. We identify a subset of the p predictors that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model. In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model.

```
X_train, X_test, y_train, y_test = train_test_split(df_cat.drop(labels=['species'], axis=1),
df_cat['species'], test_size=0.2, random_state=0)
```

X_train.shape, X_test.shape ((267,6), (67, 6))

Scaling the target value is a good idea in regression modelling. StandardScaler removes the mean and scales each feature/variable to unit variance making it easy for a model to learn and understand the problem

```
scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train) X_test =
scaler.fit_transform(X_test)

classifier = LogisticRegression(random_state = 0,multi_class='multinomial',
solver='lbfgs') classifier = LogisticRegression(random_state = 0)

sfs = SFS(classifier,
                    k_features=4,
                    forward=True,
                    floating=True, verbose=2,
                    scoring='r2', cv=3)

sfs = sfs.fit(np.array(X_train), y_train)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers. [Parallel(n_jobs=1)]: Done 1 out of
1 | elapsed: 0.0s remaining: 0.0s [Parallel(n_jobs=1)]: Done 6 out of 6 | elapsed: 0.1s finished [2022-05-16 22:44:23]

Features: 1/4 -- score: 0.672356128552036 2[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent
workers. [Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.0s remaining: 0.0s [Parallel(n_jobs=1)]: Done 5 out of 5 |
elapsed: 0.2s finished [Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.0s remaining: 0.0s [Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed:
0.0s finished [2022-05-16 22:44:23]

Features: 2/4 -- score: 0.932395569123946 6[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent
workers. [Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.0s remaining: 0.0s [Parallel(n_jobs=1)]: Done 4 out of 4 |
elapsed: 0.1s finished [Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.0s remaining: 0.0s [Parallel(n_jobs=1)]: Done 2 out of 2 | elapsed:
0.0s finished [2022-05-16 22:44:23]

Features: 3/4 -- score: 0.980647966949336 9[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent
workers. [Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.0s remaining: 0.0s [Parallel(n_jobs=1)]: Done 3 out of 3 |
elapsed: 0.1s finished [Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.0s remaining: 0.0s [Parallel(n_jobs=1)]: Done 3 out of 3 | elapsed:
0.1s finished [2022-05-16 22:44:23]

Features: 4/4 -- score: 0.980711809349970 6
sfs.k_feature_idx_ (1, 2, 3, 5)
```

These are the best features that forward elimination selected for us :
culmen_length_mm , culmen_depth_mm , flipper_length_mm, sex

BACKWARD SELECTION

Like forward stepwise selection, backward stepwise selection provides an efficient alternative to best subset selection. However, unlike forward stepwise selection, it begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.

```
sfs = SFS(classifier,
          k_features=4, forward=True,
          floating=False, verbose=2,
          scoring='r2', cv=3
sfs = sfs.fit(np.array(X_train), y_train)
```

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers. [Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.0s remaining: 0.0s [Parallel(n_jobs=1)]: Done 6 out of 6 | elapsed: 0.3s finished [2022-05-16 22:44:54]

Features: 1/4 -- score: 0.7110601946533626[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers. [Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.1s remaining: 0.0s [Parallel(n_jobs=1)]: Done 5 out of 5 | elapsed: 0.4s finished [2022-05-16 22:44:55]

Features: 2/4 -- score: 0.9131073784739172[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers. [Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.1s remaining: 0.0s [Parallel(n_jobs=1)]: Done 4 out of 4 | elapsed: 0.4s finished [2022-05-16 22:44:55]

Features: 3/4 -- score: 0.9854859752120025[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers. [Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.1s remaining: 0.0s [Parallel(n_jobs=1)]: Done 3 out of 3 | elapsed: 0.3s finished [2022-05-16 22:44:55]

Features: 4/4 -- score: 0.9903878258753022

These are the best features that Backward elimination selected for us:
culmen_length_mm, culmen_depth_mm, flipper_length_mm, body_mass_g

FITTING THE MODEL

LOGISTIC REGRESSION

```
X_train, X_test, y_train, y_test = train_test_split(df_cat.drop(labels=['species', 'island'], axis=1),
                                                    df_cat['species'],
                                                    test_size=0.2,
                                                    random_state=0)

X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)

classifier = LogisticRegression(random_state = 0,multi_class='multinomial', solver='lbfgs')
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
```

Checking how well the logistic regression model performed by looking at the confusion matrix and Accuracy of the model on the test set
Confusion Matrix:

```
 array([[36,  1,  0],
        [ 0,  9,  0],
        [ 0,  0, 21]])
```

Accuracy : 0.9850746268656716

CONCLUSION

We have successfully used statistical methods for the analysis of our data. We used non-parametric tests to identify the relation (check if their means are the same) between the numeric input columns and the categorical output columns. Then we did categorical data analysis using the chi squared test of independence to test the relation between input categorical variables and output categorical variables. Then to fit the logistic regression model, we used some techniques to reduce the features and select the best features accordingly. Then we fitted the logistic regression model to predict the species of the penguin. The fit of the model was tested on the test set, and we managed to get an accuracy of 98.5%.

REFERENCES

1. Gorman KB, Williams TD, Fraser WR (2014) Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (Genus *Pygoscelis*). PLoS ONE 9(3): e90081. doi:10.1371/journal.pone.0090081
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2015). An Introduction to Statistical Learning with Applications in R, Edn. 6th.
3. Safari-Katesari, H., & Zaroudi, S. (2020). Count copula regression model using generalized beta distribution of the second kind. *Statistics*, *21*, 1-12.
4. Katesari, H. S., & Vajargah, B. F. (2015). Testing adverse selection using frank copula approach in Iran insurance markets. *Mathematics and Computer Science*, *15*(2), 154-158.
5. Katesari, H. S., & Zarodi, S. (2016). Effects of coverage choice by predictive modeling on frequency of accidents. *Caspian Journal of Applied Sciences Research*, *5*(3), 28-33.
6. Kumar, P., Hashemi, M., Herbert, S. J., Jahanzad, E., Safari-Katesari, H., Battaglia, M., Zandvakili, O. R., & Sadeghpour, A. (2021). Integrated Management Practices for Establishing Upland Switchgrass Varieties. *Agronomy*, *11*(7), 1400.
7. Rice, J. A. (2006). *Mathematical statistics and data analysis*. Cengage Learning.
8. Safari-Katesari, H., & Zaroudi, S. (2021). Analysing the impact of dependency on conditional survival functions using copulas. *Statistics in Transition New Series*, *22*(1).
9. Safari-Katesari, H., Samadi, S. Y., & Zaroudi, S. (2020). Modelling count data via copulas. *Statistics*, *54*(6), 1329-1355.
10. Safari Katesari, H., (2021) Bayesian dynamic factor analysis and copula-based models for mixed data, PhD dissertation, Southern Illinois University Carbondale
11. Zaroudi, S., Faridrohani, M. R., Behzadi, M. H., & Safari-Katesari, H. (2022). Copula-based Modeling for IBNR Claim Loss Reserving. *arXiv preprint arXiv:2203.12750*.
12. AL-Amery, M., Battaglia, M., Serson, W., Sadeghpour, A., Lee, C. D., Knot, C., Swiggart, E., Safari-Katesari, H., & Hildebrand, D. (2022). Yield and growth characteristics of a high oil–protein soybean with enhanced diacylglycerol acyltransferase. *Agronomy Journal*, *114*(2), 1146-1154.