Penguin Weight Estimator

A predictive model using modern Machine Learning techniques to accurately estimate penguin weight.

David Krupar

Contents

Letter of Transmittal

David Krupar, Developer

Pandas and Penguins LLC

420 Panda Lane

Seattle, WA 98101

January 23, 2023

Hans Schmidt, President

Arctic Solutions LLC

Sölvhólsgötu 7

101 Reykjavík

ICELAND

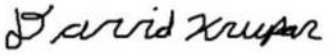
Dear Mr. Schmidt,

I submit herewith a proposal of a data driven software tool set to be designed and maintained by Pandas and Penguins LLC.

Questions relating to this letter of transmittal shall be directed to myself. I can be reached anytime by email at dkrupar@my.wgu.edu.

Thank you for your consideration, and please review the project proposal attached.

Sincerely,



David Krupar

Section A - Project Proposal/Suggested Solutions

Problem Summary

Following past conversations and also in depth market research, it has been made clear that Arctic Solutions has encountered a variety of issues regarding accuracy while obtaining penguin weight data points. The issue that has been discussed and highlighted in various meetings is that Arctic Solutions is having problems obtaining access to the use of reliable and simple to use weight scales for their penguin weight data collection. It has been made clear to us at Pandas and Penguins LLC, that with the variety of clientele Arctic Solutions has fought for and earned, that this is providing a lower quality of service to their customers. Being that the majority of customers for Arctic Solutions are at an institutional and government level, this issue demands a reliable and thoughtful solution.

Product Benefits

Following an in depth study into the processes and excursions in which Arctic Solutions has to take for its customers, we believe there will be many benefits for using our data driven product. The current system employed by Arctic solutions is to use expensive and potentially unreliable weight scales to gather data on penguin weight for their customers. It has also been discovered that there are various supply chain and availability issues for such weight scaling systems. Using Pandas and Penguins proposed software product, Arctic Solutions stands to benefit from a cost savings perspective and also will benefit from a time efficiency standpoint. The tool designed by Pandas and Penguins will offer an accurate, reliable, lower cost solution for obtaining penguin weight when compared to Arctic Solutions current mode of operation.

Product Description

The proposed solution is a data driven, machine learning trained user interface which may be run on any computing device that can run the popular python IDE known as PyCharm. The entirety of the product will be two files, one being the main interface, the other for data training. It is critical that this product is accurate and also stays true to the time and cost savings forecast. The data product proposed by Pandas and Penguins will have a security feature in the form of a user login upon use. The product will be trained and proven statistically accurate for future resulting outputs using a highly detailed and reliable dataset. It will be designed in an efficient manner for simple and quick response time while in use by Arctic Solutions on their many times dangerous excursions.

Data Description

The penguin weight estimator software tool will be trained using modern machine learning techniques. This will be accomplished using the expertise and knowledge of the data scientists at Pandas and Penguins LLC. The original dataset used to train this data product will be a dataset of 344 penguins comparing the physical measurements and their corresponding weights. This particular set of data will be relevant and useful to provide an accurate source of training and testing data for the product.

Project Objectives

The objectives of the proposed data product for arctic solutions include the following points.

1. Provide a data driven and trained software product with proven statistical accuracy.
2. Interface must be simplified and robust enough for multiple user effectiveness with minimal training.
3. The software product proposed must save Arctic Solutions resources in the context of time and money.

Project Hypotheses

Using the expertise of Pandas and Penguins data science and software development team, arctic solutions will have a proven accurate and fixed cost method of obtaining the weight of penguins for their expeditions. The interface will be simple to use, quick responding, and provide statistically proven accurate results. This will be accomplished by using modern techniques in data organization which will simplify the input process. Also, using modern machine learning techniques, an algorithmic model will be trained behind the scenes, to provide the customer a highly accurate output data point.

Project Methodology Outline

The project methodology used for the proposed data product will be a classical waterfall model.

1. *Feasibility study* In this phase the financial and technical feasibility will be discussed. A variety of solutions will be identified and weighted on their various benefits and drawbacks as proposed solutions.
2. *Requirement analysis and specification* The customer, Arctic Solutions, will have a requirements analysis done for the specific issue at hand. Documenting the various stages and progress of these discussions and the observations and further conclusions made will be critical in this phase. This phase will be concluded by both parties completing a software requirement specification (SRS) document. This will serve as a contract between Pandas and Penguins LLC and Arctic Solutions LLC to provide clarity on the project.
3. *Design* The software requirement specification (SRS) document will serve as a reference for this phase in which the requirements will be driven into a source code style of solution. The result of this stage will be a high level thoughtful design of the overall software architecture to be used in the solution.

4. *Coding and unit testing* The phase of coding and unit testing can be described as the translation of the system design into a functioning source code. Each module that was constructed during the design phase will have its own committed source code using a modern software framework, in this case Python. Each module will be checked using modern testing methods to confirm that they are working as intended in the design phase.

5. *Integration and system testing* Following unit testing the integration and system testing must be completed. Various modules will have their integration capabilities tested in an incremental sense, in order to provide a reliable and well structured data product for Arctic Solutions. Once the successful integration and testing has been completed for the various modules, the system as a whole will now be constructed. Three methods of system testing will be used for further quality assurance. Alpha testing will be conducted by the software development team at Pandas and Penguins. The beta testing phase is conducted by a user base, in this case it could be Arctic Solutions themselves, or another group of non-technical users. Finally the acceptance testing must be completed by the end customer to determine whether it will be a finished delivered product or will lead to a rejected product.

6. *Maintenance* Maintenance is thought to be the most important phase of a software development cycle. There will be three types of maintenance provided by Pandas and Penguins for this product. Corrective maintenance will be conducted to correct errors not found during the development phase. Perfective maintenance will be done to enhance functionalities of the data product per the request of the customer. And finally adaptive maintenance will be available to Arctic Solutions as a way to adapt the data product to be used in a different computing platform or operating system if needed.

Funding Requirements

Using modern and efficient tools, the funding of this project has been kept to a lower level when compared to its functional capabilities. The funding will require two separate installments of $15,000 for the design and finished product provided by Pandas and Penguins to its customer, Arctic Solutions. The first installment will be due following the software requirement specification (SRS) document creation. The second installment will be due on a successful round of acceptance testing on behalf of Arctic Solutions. There will also be a yearly maintenance fee of $5,000, which will include up to 50 hours of customer support for any issues that may arise, or potential modifications requested on behalf of Arctic Solutions.

Stakeholder Impact

The stakeholders of Arctic Solutions will have access to the software design and testing phases. The entire project will be open for questioning and probing due to the extensive effort by Pandas and Penguins to have a well documented and thoughtful development process. The overall impact however, will be in line with the project's goals of providing a product to save the resources of Arctic Solutions in the forms of time and money.

Data Sensitivity Precautions

The task at hand is creating a data driven software product to predict the weight of penguins without the use of weight scales. A potential ethical issue would be the handling of the penguins on behalf of arctic solutions and their excursions. This will be communicated in the design phase as Pandas and Penguins discusses the inputs of measurement chosen for the software input. Also, Arctic Solutions has many government and institutional customers that will be using the data produced by this product. The data driven tool set must have a well communicated and well documented accuracy standard, in order to maintain a high quality of data for such large scale and demanding customers.

Relevant Expertise

The partners at Pandas and Penguins LLC have over 10 years of experience in developing software tools for many fortune 500 companies. The software development and business team have worked together on many projects of various sizes to provide solutions that are used today in many functioning and profitable organizations. Pandas and Penguins LLC prides itself on customer satisfaction and repeat business, leading to a great reputation in the software industry.

Section B - Technical Executive Summary

Decision-Support System (DSS) Problem Statement

Decision Support Systems (DSS), when properly assessed, are aimed at less well structured problems of various upper level managers. Following communications and a variety of analytical studies between Pandas and Penguins LLC and Arctic Solutions, a decision support opportunity has been discovered. A proper solution to this opportunity must feature ease of use in an interactive mode, while also emphasizing flexibility and adaptability. The discovered opportunity is the use of a software tool built on a modern framework to streamline the process of weighing penguins by staff of Arctic Solutions.

Customer Summary

The customer in which Pandas and Penguins is working with for this data driven product is continually providing services to large scale groups and organizations. These future users of this product work in very challenging terrains and for very demanding clients with tight timelines and competitive bidding for services. Because of these realities, Pandas and Penguins will offer a software data product that will provide accurate and quick data, on a statistically proven, reliable, and fixed cost basis.

Gaps in Current Data Products

Current processes used by many service companies in the industry leave much to be desired in the sense of simplicity, reliability, and cost. Using weight scales on such treacherous expeditions has led to a large loss of time while also increasing the financial burden of such excursions. The current system of using weight scales has proven difficult as it employs many third party providers which have not adapted to the modern business requirement of flexibility and simplicity of use. This is quite common, and a solution that software often offers to the business world. This is yet another occurrence of such a trend, using machine learning to predict the weight of penguins without the use of a complex network of third party providers of physical weight scales.

Data Analysis

The steps used in the data product life cycle for the proposed solution on behalf of Pandas and Penguins LLC will include the following.

1. *Experiment*

The use of penguins.csv, which is a dataset of 344 penguins and their weight based on a large variety of inputs will be used. A variety of processes will be used to organize and assign these variables as useful for our output values of penguin weight. The original source of the penguin dataset used can be found at https://www.kaggle.com/datasets/mustafabozka/palmers-penguins. Using the pandas library within Python, ~~the data will be read from CSV format and placed into a~~ dataframe format. Following those steps, the variables will be compared using Pearson's Correlation Coefficient, in order to use the least amount of inputs while still providing the most accurate output.

2. *Implementation*

Using software best practices and modern tools, Pandas and Penguins will implement a finalized system which will provide a simple and reliable user interface which outputs the weight of a penguin given a variety of physical measurements.

3. *Deploy*

The software tool set will be tested on a variety of physical computers. This software must be deployed into real world use in order to complete the task at hand and will include such considerations as versioning, acceptance, and maintenance.

4. *Monitor*

Following a successful deployment, a continual effort will be made on the technical and business performance of the software product.

Product Methodology

The project methodology used for the proposed data product will be a classical waterfall model.

1. *Feasibility study* In this phase the financial and technical feasibility will be discussed. A variety of solutions will be identified and weighted on their various benefits and drawbacks as proposed solutions.
2. *Requirement analysis and specification* The customer, Arctic Solutions, will have a requirements analysis done for the specific issue at hand. Documenting the various stages and progress of these discussions and the observations and further conclusions made will be critical in this phase. This phase will be concluded by both parties completing a software requirement specification (SRS) document. This will serve as a contract between Pandas and Penguins LLC and Arctic Solutions LLC to provide clarity on the project.
3. *Design* The software requirement specification (SRS) document will serve as a reference for this phase in which the requirements will be driven into a source code style of solution. The result of this stage will be a high level thoughtful design of the overall software architecture to be used in the solution.
4. *Coding and unit testing* The phase of coding and unit testing can be described as the translation of the system design into a functioning source code. Each module that was constructed during the design phase will have its own committed source code using a modern software framework, in this case Python. Each

module will be checked using modern testing methods to confirm that they are working as intended in the design phase.

5. *Integration and system testing* Following unit testing the integration and system testing must be completed. Various modules will have their integration capabilities tested in an incremental sense, in order to provide a reliable and well structured data product for Arctic Solutions. Once the successful integration and testing has been completed for the various modules, the system as a whole will now be constructed. Three methods of system testing will be used for further quality assurance. Alpha testing will be conducted by the software development team at Pandas and Penguins. The beta testing phase is conducted by a user base, in this case it could be Arctic Solutions themselves, or another group of non-technical users. Finally the acceptance testing must be completed by the end customer to determine whether it will be a finished delivered product or will lead to a rejected product.

6. *Maintenance* Maintenance is thought to be the most important phase of a software development cycle. There will be three types of maintenance provided by Pandas and Penguins for this product. Corrective maintenance will be conducted to correct errors not found during the development phase. Perfective maintenance will be done to enhance functionalities of the data product per the request of the customer. And finally adaptive maintenance will be available to Arctic Solutions as a way to adapt the data product to be used in a different computing platform or operating system if needed.

Product Deliverables

The success of the project will be dependent on the following list of deliverables:

1. Python script designed and created to operate within PyCharm IDE.

2. Included within the Python script will be a penguins.csv file for machine learning and training.

4. A zip file containing all of the above documents and files, for ease of use and sharing.

Implementation Plan

The following steps illustrates the implementation plan for the creation of a predictive model data product.

1. A data set will be used in CSV file format. This file is named “penguins.csv” and will be the sole source of data for the proposed software product. This specific file was chosen as it is sourced from a reliable website, kaggle.com, and also highly relevant to solving the task at hand, obtaining weights of penguins without the use of scale equipment being onsite.
2. Using the pandas library within python, the CSV file will be reorganized and cleaned from any null values. Also, again using the pandas library within the python language, the file will be read and processed into a dataframe format for use in our machine learning training and exploration.

3. Using the preprocessing package provided by sklearn, any columns in the dataset that are in a text format will be converted into an integer value. An example of this would be transforming datapoints of “Species” or “male” and “female” into integers of 0 and 1 for future use of the machine learning model.

4. An analysis will be done on the input variables using Pearsons correlation index. This will allow us to make design decisions to balance the accuracy of the output while also provided a simple to use product for the user base.

5. A variety of visual aids will be generated during this stage of the implementation plan. These will help our team understand the relationship between variables and our goal of providing an accurate weight prediction.

6. The penguins.csv dataset, now in dataframe form, will be broken into X and y variables and assigned to training and testing variants of X and y for use in our algorithm.
7. A random forest algorithm will be initialized from the sklearn library to build a regression based predictor for our output variable which is the weight of a penguin given physical dimensions as inputs. This particular algorithm was first proposed as a machine learning solution in the mid 1990’s.
8. A new variable will be initialized, “y_pred”, this will be the output of our random forest regression powered prediction of penguin weight.
9. An analysis will be done using between our algorithm based output and the output test arrays. The tool used for this analysis will be the r^2 score, also known as the coefficient of determination.
10. Following the steps listed above, the main class will be initialized, which will be a python script written for a simple and effective user interface. This user interface will be the final product which will provide our customers with a quick and reliable cost effective solution to finding the weight of a penguin without the use of a weight scale on-site.
11. The entire python script will undergo a round of acceptance testing.

12. Pending successful customer acceptance testing, a maintenance and update plan will be initialized for customer support and satisfaction.

Evaluation Plan

The requirements and needs of the customer will be validated and verified by the following methods.

1. Using Pearson’s correlation coefficient on the input variables will validate the decision of providing a tool set that only includes input variables that lead to great accuracy. This will provide a good analysis of a balance between potential user inputs and their ability to provide an accurate penguin weight output as required by the customer

2. The Algorithm output will be analyzed using an R^2 score, known as the coefficient of determination. This will validate and verify the requirement of the customer for accurate output data given a simplified minimal input process.

3. The provided python script will provide a simple to use and effective user interface for the customer. There will be a round of acceptance testing by Arctic Solutions LLC to verify and validate that the requirements are met.

Resources and Costs

1. Programming Environment

The programming environments for the project are as follows.

-Python 3.9
Libraries used within Python 3.9 are listed below.
-sklearn
-matplotlib
-seaborn
-pandas

2. *Environment Costs* The software solution proposed on behalf of Pandas and Penguins LLC to Arctic

Solutions is a
native python script that will run within the PyCharm IDE. This keeps the costs low for the end customer, and the environment costs reflect that design choice. The end user is required to have a license for PyCharm in order to use this particular product, which as of today is listed at $250 per year. JetBrains offers lower prices on a yearly basis for returning customers. This will lead to a lower price per use over time. The Python script is created to run in an efficient and cost saving manner, therefore, the hardware requirements will be based on the PyCharm system requirements. It is expected that the hardware already used by the customer, arctic solutions, will be more than enough to run the software data solution. This leaves the environmental costs to be $250/yr per user station, and will decrease with each consecutive year.

3. *Human resources requirements*

The human resource requirements for the proposed project are as follows. The project will require two separate installments of $15,000 for the design and finished product provided by Pandas and Penguins to Arctic Solutions. The first installment will be due following the SRS document creation. The second installment will be due on a successful round of acceptance testing on behalf of Arctic Solutions. There will also be a yearly maintenance fee of $5,000, which will include up to 50 hours of customer support for any issues that may arise, or potential modifications requested on behalf of arctic solutions.

Projected Timeline and Milestones

Phase	Start	End	Duration	Dependencies	Assigned Resources
Planning and Design….					
Analysis of Requirements	1-2-23	1-3-23 EOD	2 days	None	Project Manager, Stakeholders
Collection of Data	1-4-23	1-5-23 EOD	2 days	Requirements	Project Manager, Developers
Analysis of Data	1-6-23	1-6-23 EOD	1 day	Data Collected	Developers
Development…					
Development	1-9-23	1-13-23 EOD	5 days	Planning and Design	Project Manager, Developers
Deployment of Product	1-16-23	1-17-23 EOD	2 days	Initial Development	Project Manager, Developers
Testing	1-18-23	1-20-23 EOD	3 days	Deployed Product	Developers
Documentation…					
Documentation	1-23-23	1-25-23 EOD	3 days	Tested Product	Project Manager, Developers
Maintenance	1-26-23	1-27-23 EOD	2 days	Documented Product	Developers, SQA

Section C - Product Development and Design

Descriptive Method (Pearson's Correlation Coefficient)

Our descriptive method for variable selection and potential variable elimination will be to use Pearson's Rank Correlation. It is important to note that correlation does not mean causation. However in this scenario, this descriptive tool will allow us to select a minimal number of input variables while still providing an accurate output.

```
corr = descriptive.corr(method='pearson')
```

This python code was used to create the correlation values on the "descriptive" dataframe.

	bill_length_mm	flipper_length_mm	bill_depth_mm	sex	species	body_mass_g
bill_length_mm	1.000000	0.653096	-0.228626	0.344078	0.730548	0.589451
flipper_length_mm	0.653096	1.000000	-0.577792	0.255169	0.850737	0.872979
bill_depth_mm	-0.228626	-0.577792	1.000000	0.372673	-0.740346	-0.472016
sex	0.344078	0.255169	0.372673	1.000000	0.010964	0.424987
species	0.730548	0.850737	-0.740346	0.010964	1.000000	0.750434
body_mass_g	0.589451	0.872979	-0.472016	0.424987	0.750434	1.000000

Printed values of Pearson's Correlation Coefficient

	bill_length_mm	flipper_length_mm	bill_depth_mm	sex	species	body_mass_g
bill_length_mm	1.000000	0.653096	-0.228626	0.344078	0.730548	0.589451
flipper_length_mm	0.653096	1.000000	-0.577792	0.255169	0.850737	0.872979
bill_depth_mm	-0.228626	-0.577792	1.000000	0.372673	-0.740346	-0.472016
sex	0.344078	0.255169	0.372673	1.000000	0.010964	0.424987
species	0.730548	0.850737	-0.740346	0.010964	1.000000	0.750434
body_mass_g	0.589451	0.872979	-0.472016	0.424987	0.750434	1.000000

The area that is notated in the above image is the data that will help us decrease our user input requirements as required by the customer.

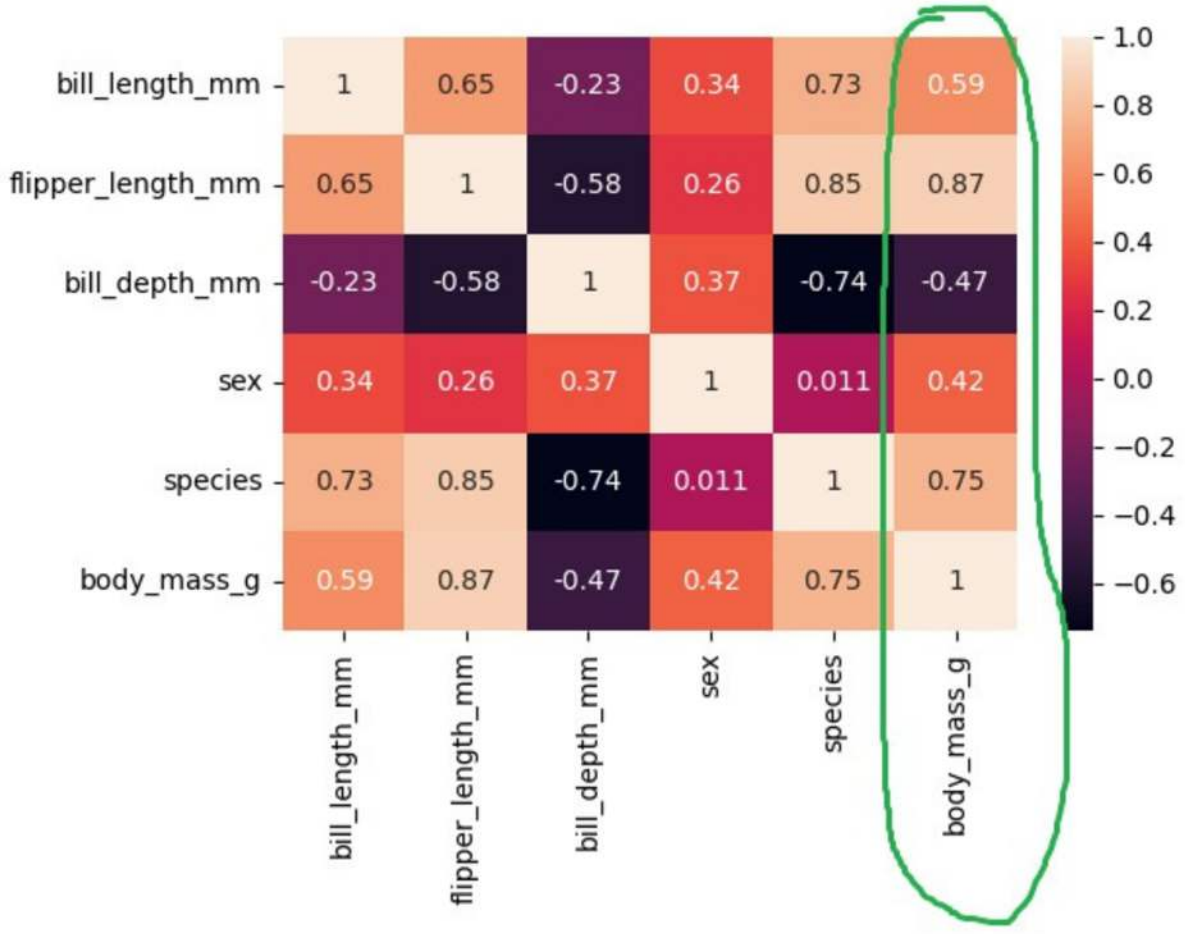
Using the seaborn library for python will allow us to visually display this information in many interesting and useful ways.

```
import seaborn as sb
```

The library must first be imported into the “Main.py” file

```
sb.heatmap(corr, annot=True)
plt.tight_layout()
plt.show()
```

Following successful import, the python code snippet above allows us to retrieve the chart shown below



The data that is circled in green will provide us with information to make a decision on which variables to keep. It will be a balance between the customer’s requirements of a simple quick input process, but it must also be accurate as well.

Predictive Method (Random Forest Algorithm)

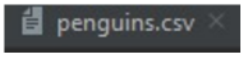
For this particular project, a random forest algorithm was used to create a predictive method for our output variable, penguin weight in grams. Reasons for this decision include the ease of application, low computational cost, high predictive accuracy, flexibility, and interpret ability of random forest machinery (Linguin, 2018). This algorithm was first conceived in the mid 1990's and we believe it is a good choice for the data product. There were many hours spent into research and inquiry into different data sets and differing algorithms to use as a final product. Pandas and Penguins will review some of the information found which makes this algorithm a wise choice. One report noted, Random Forest is a powerful tool capable of delivering performance that is among the most accurate methods to date (Svetnik et al., 2003). It is important to mention why the algorithm was selected, besides the following notes listed above which we found very impressive, one particular journal discovery stuck out to us. Random forests provide consistent pairwise similarity measures for multiple modalities, thus facilitating the combination of different types of feature data (Gray et al., 2013). While it is true that the random forest algorithm truly shines with large data sets with an incredible scale of input variables, it is still a good fit for the product at hand. The efficacy of this algorithm on very simple and smaller scale data sets will be proven later in the document, however it should be noted now that it performed very well.

Datasets

The dataset used for creating the data product was found at Kaggle.com. Below is a link for the dataset.

https://www.kaggle.com/datasets/mustafabozka/palmers-penguins

Shown below is a screenshot of the datafile in the PyCharm IDE. This is the only dataset used for the data product.



A list of *CRITICAL* discussions in upcoming areas of section C is listed below.

1. Data preparation
2. Data test/training split
3. Model performance
4. Success/Failure of model

Decision-Support Functionality

The fully functional data product produced by Pandas and Penguins LLC will offer a variety of decision support functions to the customer, Arctic Solutions LLC. For its first iteration, it will provide an output of a specific penguin weight based upon highly correlated input data without the use of weight scales. Using this product, Arctic Solutions will be able to control the input variables, hence providing them the ability to tailor fit their measuring processes for efficiency. The first iteration will include two input values, for the sake of speed and efficiency. However, further adaptions can be made to provide arctic solutions the ability to customize the output of the product, in order to assist in decision support functionality.

Data Cleaning

Using the pandas library within Python, there have been many steps that support cleaning, parsing and wrangling of datasets.

```
df = pd.read_csv('CSV/penguins.csv', encoding='unicode_escape')
```

Shown above is the python code which used the pandas library to read the data file

```
df.drop('year', inplace=True, axis=1)

df.drop('island', inplace=True, axis=1)
```

Shown above is the python function to drop columns from the penguins dataset

```
if df.isnull().values.any() == True:
    newdf = df.dropna()
```

The main cleaning method used is the isnull() function. This python code is used for the purpose of, if any null values are found in a row, for all rows, then we must delete the entire row.

```
le = preprocessing.LabelEncoder()

newdf['species'] = le.fit_transform(newdf['species'])
newdf['sex'] = le.fit_transform(newdf['sex'])
```

Listed above is the LabelEncoder() method used from the sklearn preprocessing library. This is helpful for data cleaning and organization as it takes any column value that is in text format and changes to an integer value. This must be done for our testing and training of the machine learning algorithm, in this case, the random forest.

```
     species  bill_length_mm  ...  body_mass_g  sex
0          0            39.1  ...       3750.0    1
1          0            39.5  ...       3800.0    0
2          0            40.3  ...       3250.0    0
4          0            36.7  ...       3450.0    0
5          0            39.3  ...       3650.0    1
..       ...             ...  ...          ...  ...
339        1            55.8  ...       4000.0    1
```

Listed above is the output after using the label encoder. As you can see the ‘species’ and ‘sex’ have been changed from text based data points, to integer values.

Data Exploration and Preparation

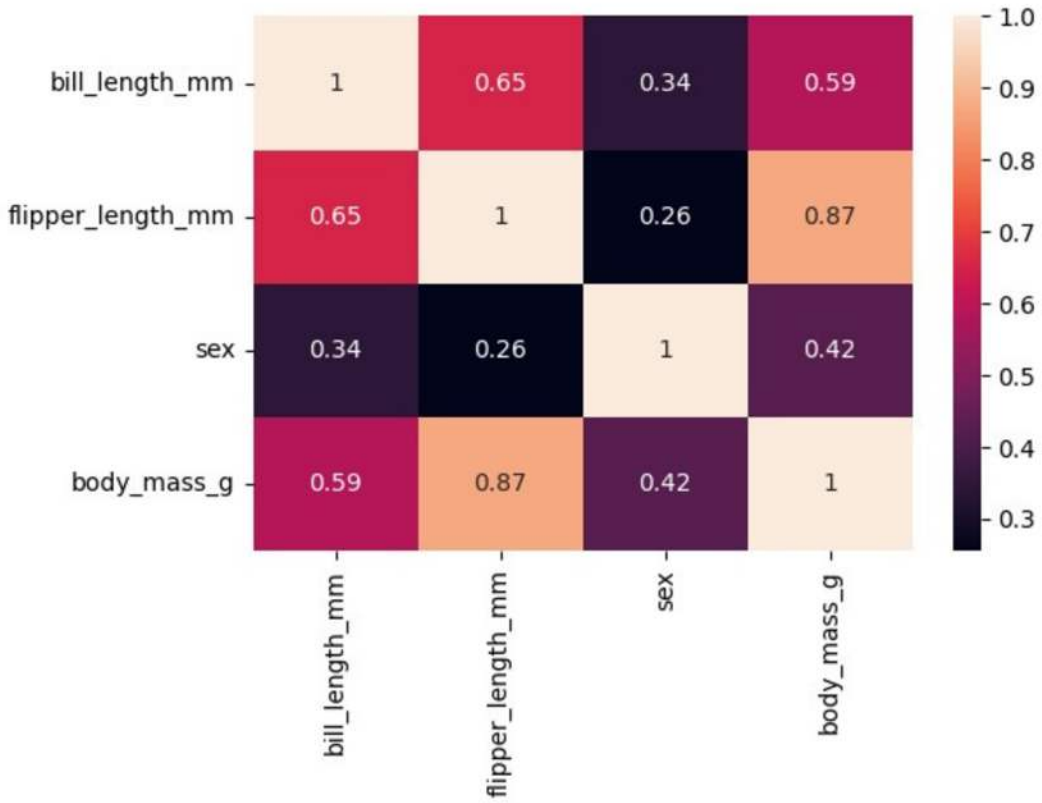
Many methods were used to support the exploration and preparation of the dataset. Most of the data preparation was done using the pandas library within python. The method used from pandas was pd.read_csv. Following that the sklearn library provided us with the labelencoder() method for preprocessing. Pearson's correlation coefficient was the main tool used to explore the dataset. And finally sklearn provided us a method to prepare the data for use in our random tree regression functions.

```
data = descriptive.values
X, y = data[:, :-1], data[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

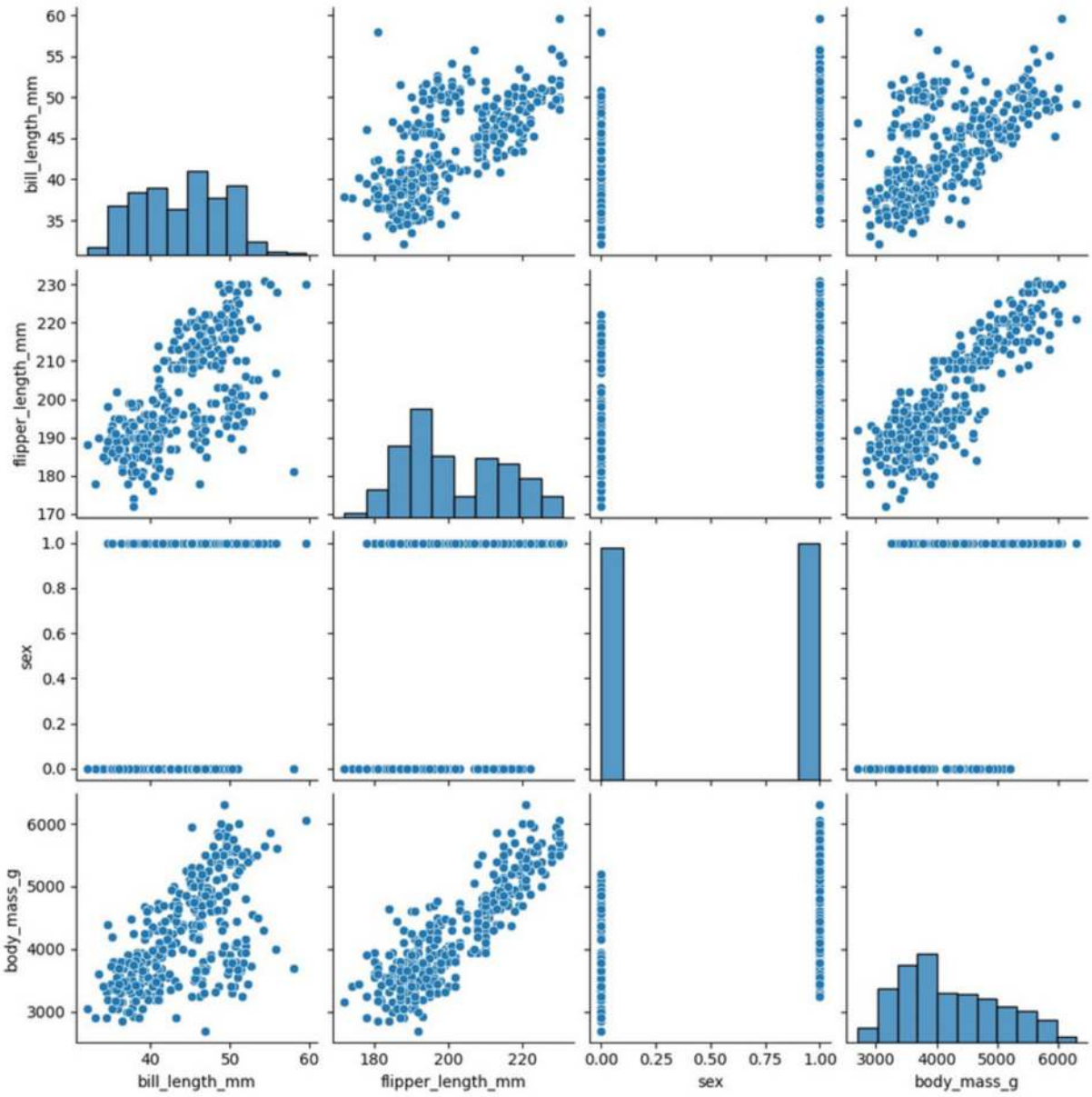
Shown above is the process used to prepare the data for the random forest algorithm. The train/test split is shown in the third line. In order to perform the train/test split, we must assign information into X and y values, that is shown in the second line of code.

Data Visualization

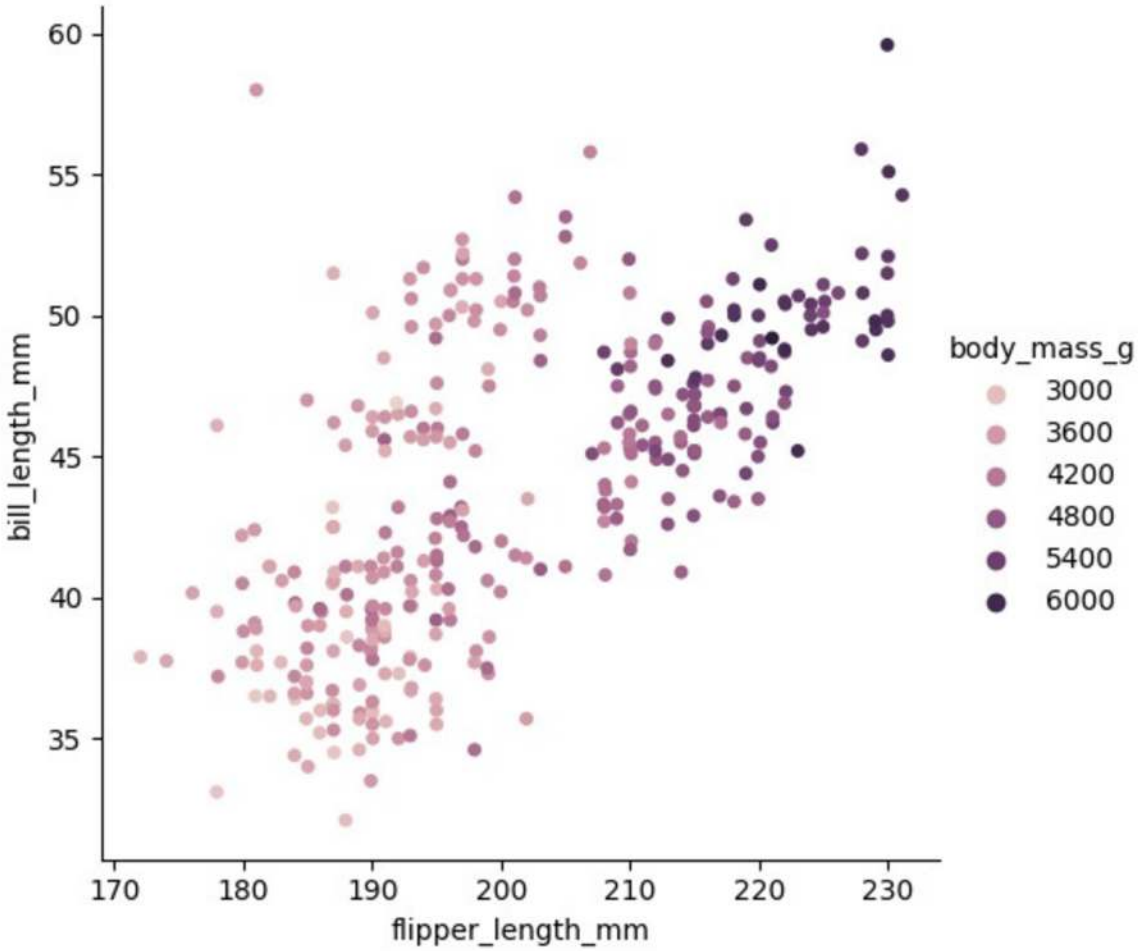
Data visualization functionalities for data exploration and inspection were provided by using the seaborn library within the PyCharm IDE. Seaborn is a Python data visualization library based on matplotlib. It is very useful for building highly capable data driven software tools.



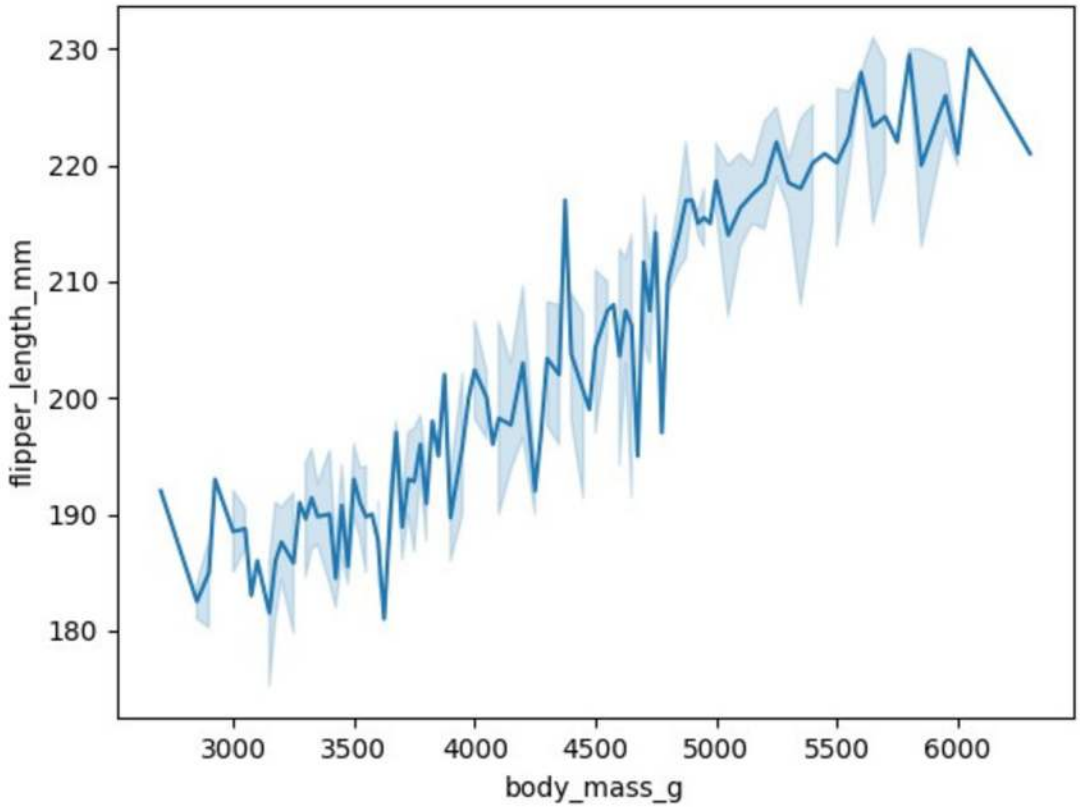
seaborn heatmap including pearson correlation values



seaborn pairplot



seaborn catplot with a color based body mass result



seaborn lineplot

Interactive Queries

The data product supplied by Pandas and Penguins LLC will have a fully functional interface for the user designed to run inside of the PyCharm IDE. The user will run the filename “Main.py” and be prompted for a USER ID for security purposes.

```
To run interface, please enter USER ID
```

The user will enter the USER ID “TEST” to complete the login.

The user will be shown a valid USER ID login message

```
To run interface, please enter USER ID TEST
USER ID VALID
```

The user will be prompted to enter the penguin bill length

```
To run interface, please enter USER ID TEST
USER ID VALID

Please enter bill_length_mm
```

Following input of bill length, the user is prompted to enter penguin flipper length

```
Please enter bill_length_mm 40
Please enter flipper_length_mm
```

```
Please enter bill_length_mm 40
Please enter flipper_length_mm 190
```

Following penguin flipper length input, the predicted weight is given to the user.

```
Please enter bill_length_mm 40
Please enter flipper_length_mm 190

Predicted Weight = 4091.5 Grams
------------------------------
```

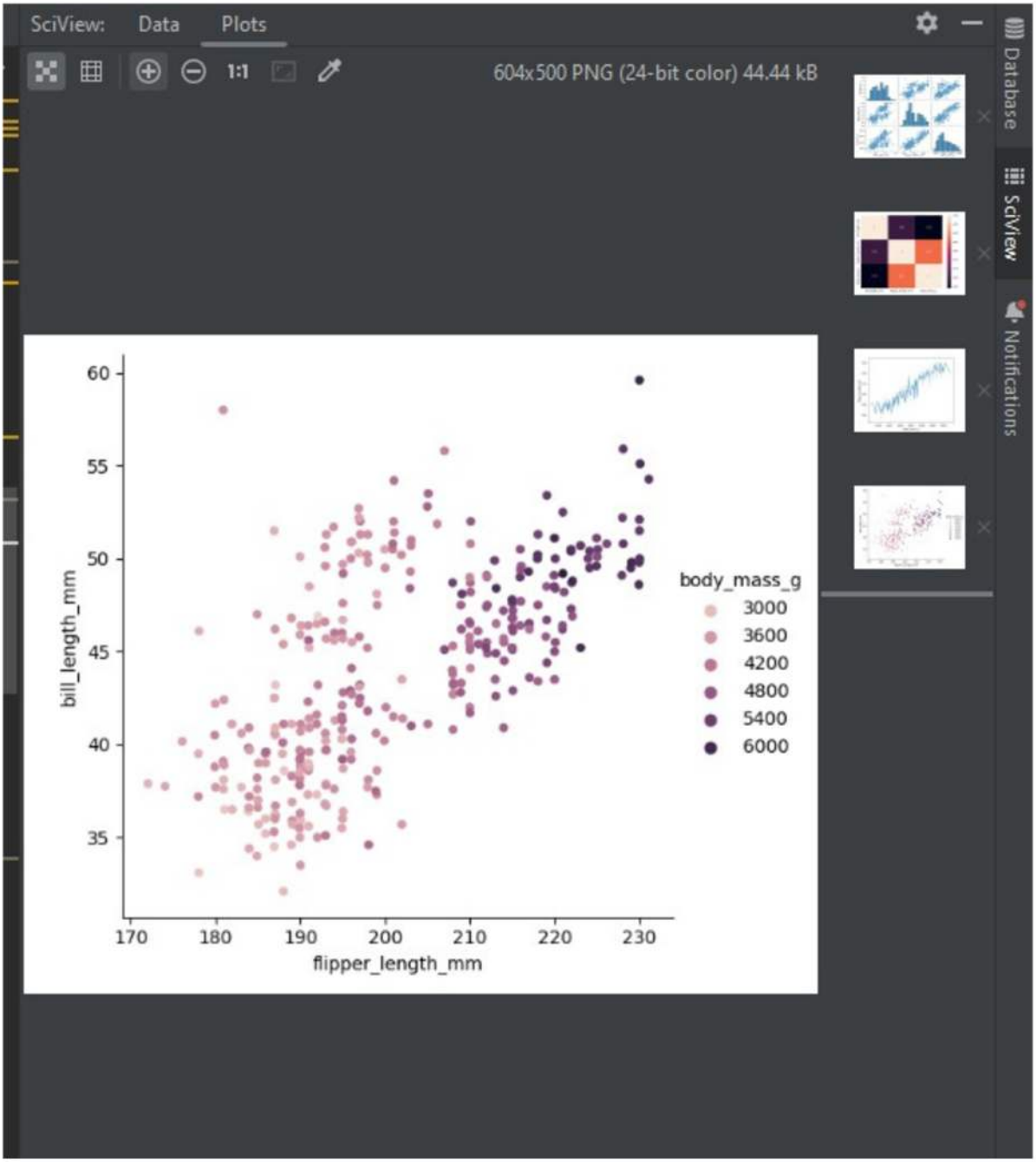
The user is then given the prompt to enter “Y” for the required user-friendly functional dashboard, or “N” to exit.

```
Please enter bill_length_mm: (Example 40) 40
Please enter flipper_length_mm: (Example 180) 180

Predicted Weight = 4130.5 Grams
------------------------------


To load user-friendly functional dashboard with 3 visualization types, Enter 'Y', otherwise enter 'N' to exit program
```

If user enters “Y”, the interface replies with the message below.

```
y

r2 Score =  0.971484511737117

Please refer to PyCharm SciView for user-friendly functional dashboard

------------------------------
Program finished, Thank you!
```

Also, In the PyCharm IDE, the SciView will appear with 4 different data visualizations



Adaptive Components

The following section will be a discussion on the implementation of machine-learning methods and algorithms. The implementation of the random forest algorithm was made simple thanks to the use of the sklearn library within the Python language.

```
data = Descriptive.values
X, y = data[:, :-1], data[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

regr = RandomForestRegressor(random_state=0)
regr.fit(X, y)
y_pred = regr.predict(X_test)
```

Shown above is the implementation of the components and machine learning methods used in the random forest algorithm driven predictive output design. This section of the codebase really is the “brain” of the entire data product. X, in this case, is assigned to the input variables, and the y value is set to the predictive output of penguin weight.

Data Product Accuracy

Many statistical equations were considered for use as functionalities to evaluate the accuracy of the data product. The final decision was to implement the statistical analysis of R^2, also known as the coefficient of determination.

```
print("r2 Score = ", r2_score(y_test, y_pred))
```

Shown above is the code used to print the generated R^2 score. Please refer to the image in the section above for the code used to generate the values y-test and y_pred. It is important to note that the random forest algorithm has to be a component in part of these values as we are verifying the accuracy of our algorithm to predict an accurate dependent variable, penguin weight.

```
r2 Score =  0.971484511737117
```

Shown above is the output, which tells us that the design choices and implementation of algorithm testing and training has given us a predictive tool that is at least 97% accurate. We believe that given the simplicity of the inputs, and also the functional and practical design of the data product, that we have obtained a successful model.

Security Features

It was important to consider functionality in the codebase to include industry-appropriate security features. It was decided to include a user login as a form of “key” to enter the full functionality of the program. It was not completely necessary to include many layers of security, given that we are collecting weights of penguins, however it is good practice to consider some basic layer of security. The intentions of malicious actors may not be known in advance, therefore, Pandas and Penguins believe the login using only a user ID is a smart defensive move for future threats on security.

```
run = input("\n\nTo run interface, please enter USER ID ")

if run == "TEST":

        print("USER ID VALID\n")

elif run != "TEST":


        print("INVALID USER ID")
        exit()
```

Shown above is the Python code which offers a layer of industry appropriate security

Product Monitoring Tools

When designing a modern machine learning predictive software service it is important to include tools to monitor and maintain the completed product. The first tool used will be the Python print() function of the accuracy of the predictive model. This printout will be included with every output result, in which the user can confirm in real time how accurate their predictive model is at that particular moment. This is a very important design choice as we must give the customer, Arctic Solutions, a real-time scientific measurement to monitor and maintain the product for their intended use case.

```
n

r2 Score =  0.971484511737117

------------------------------
Program finished, Thank you!
```

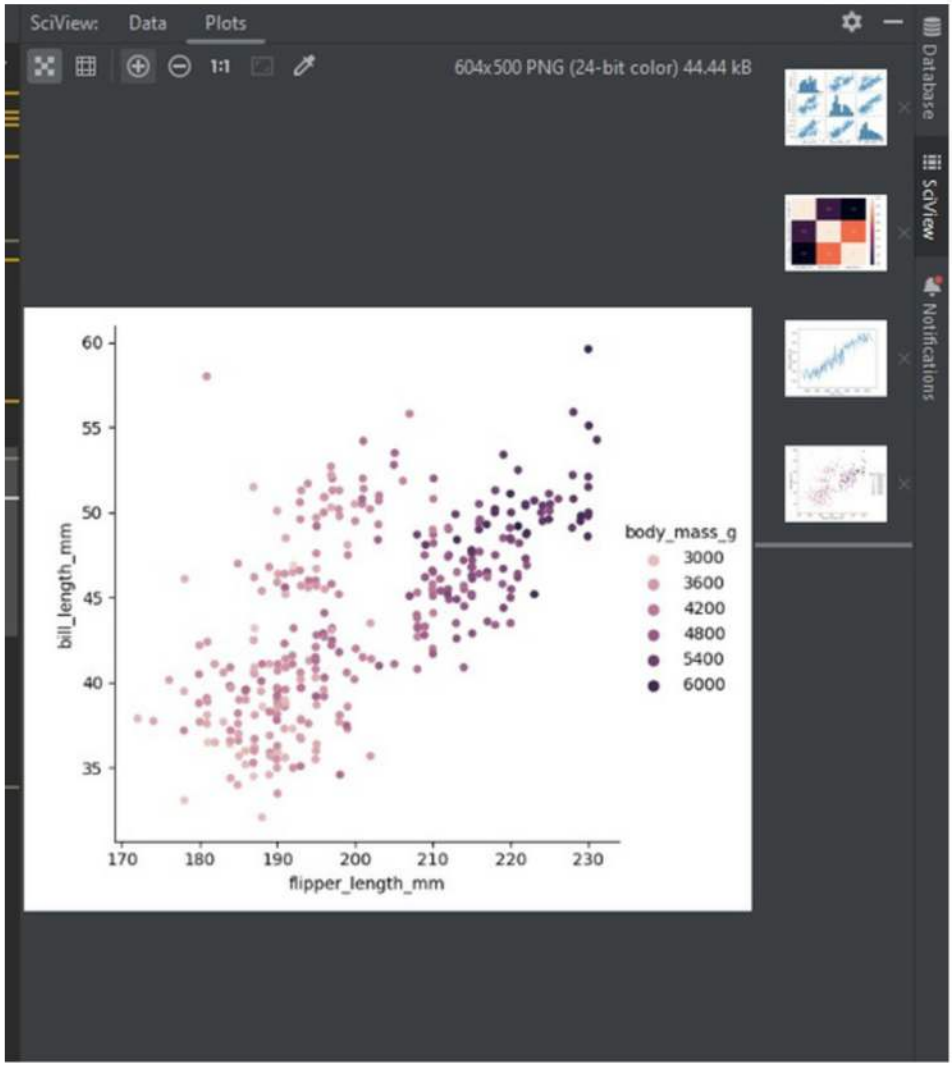
Even if the customer decides to not enter the dashboard function, the product monitoring tool is still displayed as shown above.

Functional Dashboard

A user-friendly and functional dashboard that includes at least three visualization types is included in the product with further details below.

```
Please enter bill_length_mm: (Example 40) 40
Please enter flipper_length_mm: (Example 180) 180

Predicted Weight = 4130.5 Grams
------------------------------


To load user-friendly functional dashboard with 3 visualization types, Enter 'Y', otherwise enter 'N' to exit program
```

Shown above, the user is given the option to enter the functional dashboard by entering 'Y'.



Shown above is a reference of the user-friendly functional dashboard with four visualization types.

Section D – Product Review Analysis

Business Vision and Requirements

The project purpose will be to create a software tool that is trained using machine learning to predict the weight of penguins without the need of expensive and unreliable scales. The tool will be trained with data provided on the animal species from a data set obtained from kaggle.com.

This tool has various goals which include.

- Proven statistical accuracy
- Simple to use user interface and easy to maintain code base
- Provides rapid accurate results on the weight of penguins without the need of expensive and potentially unreliable scales

The human resource requirements for the proposed project are as follows. The project will require two separate installments of $15,000 for the design and finished product provided by pandas and penguins to arctic solutions. The first installment will be due following the SRS document creation. The second installment will be due on a successful round of acceptance testing on behalf of arctic solutions. There will also be a yearly maintenance fee of $5,000, which will include up to 50 hours of customer support for any issues that may arise, or potential modifications requested on behalf of arctic solutions.

Pandas and Penguins LLC has a strategic four point plan that embodies the business vision of the team and leadership.

1. Commitment to quality

    Provide finished products that are thoroughly tested and proven

2. Provide solutions using modern tools

    Always stay up to date with industry standards and practices

3. Collaboration is key

    Cross platform and dedicated to involvement in the industry

4. Customer support for long term relationships

    Current customers are always the next best customer

Data Used

Raw and cleaned data sets with the code and executable files used to scrape and clean data is discussed below. The entire data product is built off of the penguins.csv file.

Main.py | penguins.csv

```
species,island,bill_length_mm,bill_depth_mm,flipper_length_mm,body_mass_g,sex,year
Adelie,Torgersen,39.1,18.7,181.0,3750.0,male,2007
Adelie,Torgersen,39.5,17.4,186.0,3800.0,female,2007
Adelie,Torgersen,40.3,18.0,195.0,3250.0,female,2007
Adelie,Torgersen,,,,,2007
Adelie,Torgersen,36.7,19.3,193.0,3450.0,female,2007
Adelie,Torgersen,39.3,20.6,190.0,3650.0,male,2007
Adelie,Torgersen,38.9,17.8,181.0,3625.0,female,2007
Adelie,Torgersen,39.2,19.6,195.0,4675.0,male,2007
Adelie,Torgersen,34.1,18.1,193.0,3475.0,,2007
Adelie,Torgersen,42.0,20.2,190.0,4250.0,,2007
Adelie,Torgersen,37.8,17.1,186.0,3300.0,,2007
Adelie,Torgersen,37.8,17.3,180.0,3700.0,,2007
Adelie,Torgersen,41.1,17.6,182.0,3200.0,female,2007
Adelie,Torgersen,38.6,21.2,191.0,3800.0,male,2007
Adelie,Torgersen,34.6,21.1,198.0,4400.0,male,2007
Adelie,Torgersen,36.6,17.8,185.0,3700.0,female,2007
Adelie,Torgersen,38.7,19.0,195.0,3450.0,female,2007
Adelie,Torgersen,42.5,20.7,197.0,4500.0,male,2007
Adelie,Torgersen,34.4,18.4,184.0,3325.0,female,2007
Adelie,Torgersen,46.0,21.5,194.0,4200.0,male,2007
Adelie,Biscoe,37.8,18.3,174.0,3400.0,female,2007
Adelie,Biscoe,37.7,18.7,180.0,3600.0,male,2007
Adelie,Biscoe,35.9,19.2,189.0,3800.0,female,2007
Adelie,Biscoe,38.2,18.1,185.0,3950.0,male,2007
Adelie,Biscoe,38.8,17.2,180.0,3800.0,male,2007
Adelie,Biscoe,35.3,18.9,187.0,3800.0,female,2007
Adelie,Biscoe,40.6,18.6,183.0,3550.0,male,2007
Adelie,Biscoe,40.5,17.9,187.0,3200.0,female,2007
Adelie,Biscoe,37.9,18.6,172.0,3150.0,female,2007
Adelie,Biscoe,40.5,18.9,180.0,3950.0,male,2007
Adelie,Dream,39.5,16.7,178.0,3250.0,female,2007
Adelie,Dream,37.2,18.1,178.0,3900.0,male,2007
Adelie,Dream,39.5,17.8,188.0,3300.0,female,2007
Adelie,Dream,40.9,18.9,184.0,3900.0,male,2007
```

raw penguins.csv file shown above from within PyCharm IDE

Using the pandas library within Python, there have been many steps that support cleaning, parsing and wrangling of datasets.

```
df = pd.read_csv('CSV/penguins.csv', encoding='unicode_escape')
```

Shown above is the python code which used the pandas library to read the data file.

```
df.drop('year', inplace=True, axis=1)

df.drop('island', inplace=True, axis=1)
```

Shown above is the python function to drop columns from the penguins dataset.

```
if df.isnull().values.any() == True:
    newdf = df.dropna()
```

The main cleaning method used is the isnull() function. This python code is used for the purpose of, if any null values are found in a row, for all rows, then we must delete the entire row.

```
le = preprocessing.LabelEncoder()

newdf['species'] = le.fit_transform(newdf['species'])
newdf['sex'] = le.fit_transform(newdf['sex'])
```

Listed above is the LabelEncoder() method used from the sklearn preprocessing library. This is helpful for data cleaning and organization as it takes any column value that is in text format and changes to an integer value. This must be done for our testing and training of the machine learning algorithm, in this case, the random forest.

```
     species  bill_length_mm  ...  body_mass_g  sex
0          0            39.1  ...       3750.0    1
1          0            39.5  ...       3800.0    0
2          0            40.3  ...       3250.0    0
4          0            36.7  ...       3450.0    0
5          0            39.3  ...       3650.0    1
..       ...             ...  ...          ...  ...
339        1            55.8  ...       4000.0    1
```

Listed above is the output after using the label encoder. As you can see the ‘species’ and ‘sex’ have been changed from text based data points, to integer values.

Data Focused Code

The first section will discuss the code used for the analysis of the data.

```
corr = descriptive.corr(method='pearson')
```

This python code was used to create the correlation values on the “descriptive” dataframe.

	bill_length_mm	flipper_length_mm	bill_depth_mm	sex	species	body_mass_g
bill_length_mm	1.000000	0.653096	-0.228626	0.344078	0.730548	0.589451
flipper_length_mm	0.653096	1.000000	-0.577792	0.255169	0.850737	0.872979
bill_depth_mm	-0.228626	-0.577792	1.000000	0.372673	-0.740346	-0.472016
sex	0.344078	0.255169	0.372673	1.000000	0.010964	0.424987
species	0.730548	0.850737	-0.740346	0.010964	1.000000	0.750434
body_mass_g	0.589451	0.872979	-0.472016	0.424987	0.750434	1.000000

Printed values of Pearson’s Correlation Coefficient

	bill_length_mm	flipper_length_mm	bill_depth_mm	sex	species	body_mass_g
bill_length_mm	1.000000	0.653096	-0.228626	0.344078	0.730548	0.589451
flipper_length_mm	0.653096	1.000000	-0.577792	0.255169	0.850737	0.872979
bill_depth_mm	-0.228626	-0.577792	1.000000	0.372673	-0.740346	-0.472016
sex	0.344078	0.255169	0.372673	1.000000	0.010964	0.424987
species	0.730548	0.850737	-0.740346	0.010964	1.000000	0.750434
body_mass_g	0.589451	0.872979	-0.472016	0.424987	0.750434	1.000000

The area that is notated in the above image is the data that will help us decrease our user input requirements as required by the customer.

```
corr = descriptive.corr(method='pearson')
sb.pairplot(descriptive)
plt.show()

sb.heatmap(corr, annot=True)
plt.tight_layout()
plt.show()

sb.lineplot(x='body_mass_g', y='flipper_length_mm', data=linechart)
plt.show()

dot = descriptive[['body_mass_g', 'flipper_length_mm', 'bill_length_mm']]
sb.catplot(data=dot, x='flipper_length_mm', y='bill_length_mm', hue='body_mass_g', native_scale=True)
plt.show()
```

Shown above is the code created to use the seaborn library which helped us analyze the dataset

This section will discuss the code used to construct the predictive data product.

```
data = Descriptive.values
X, y = data[:, :-1], data[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

regr = RandomForestRegressor(random_state=0)
regr.fit(X, y)
y_pred = regr.predict(X_test)
```

Shown above is a block of code using Python which facilitates the construction of the data product. Using the sklearn library, the algorithm known as random forest was used for this particular data product. In the image above the data is split into independent and dependent variables. X being the input values of penguin dimensions, and y being the output value of penguin weight. The data is then put into a train/test split in order to format the information for the RandomForestRegressor(). The dataset is then applied to the algorithm using the regr.fit() function. Then finally the random forest prediction output is assigned to the variable y_pred.

Hypothesis Assessment

We will use the original hypothesis to answer the following question. How did the model perform? Did it succeed? Did it fail? For review, the hypothesis from section A is listed below.

"Using the expertise of Pandas and Penguins data science and software development team, Arctic Solutions will have a proven accurate and fixed cost method of obtaining the weight of penguins for their expeditions. The interface will be simple to use, quick responding, and provide statistically proven accurate results. This will be accomplished by using modern techniques in data organization which will simplify the input process. Also, using modern machine learning techniques, an algorithmic model will be trained behind the scenes, to provide the customer a highly accurate output data point."

The model and project were a success and performed up to the standard of the original hypothesis.

1. Simple to use interface

   As shown by the interactive query section, the interface is very simple to use.
2. Quick responding

   As seen in the source code and use of product, the user experience is very fast.
3. Statistically proven accurate results

   Using the r^2 analysis, the output has been proven accurate.
4. Use modern techniques to simplify input requirements

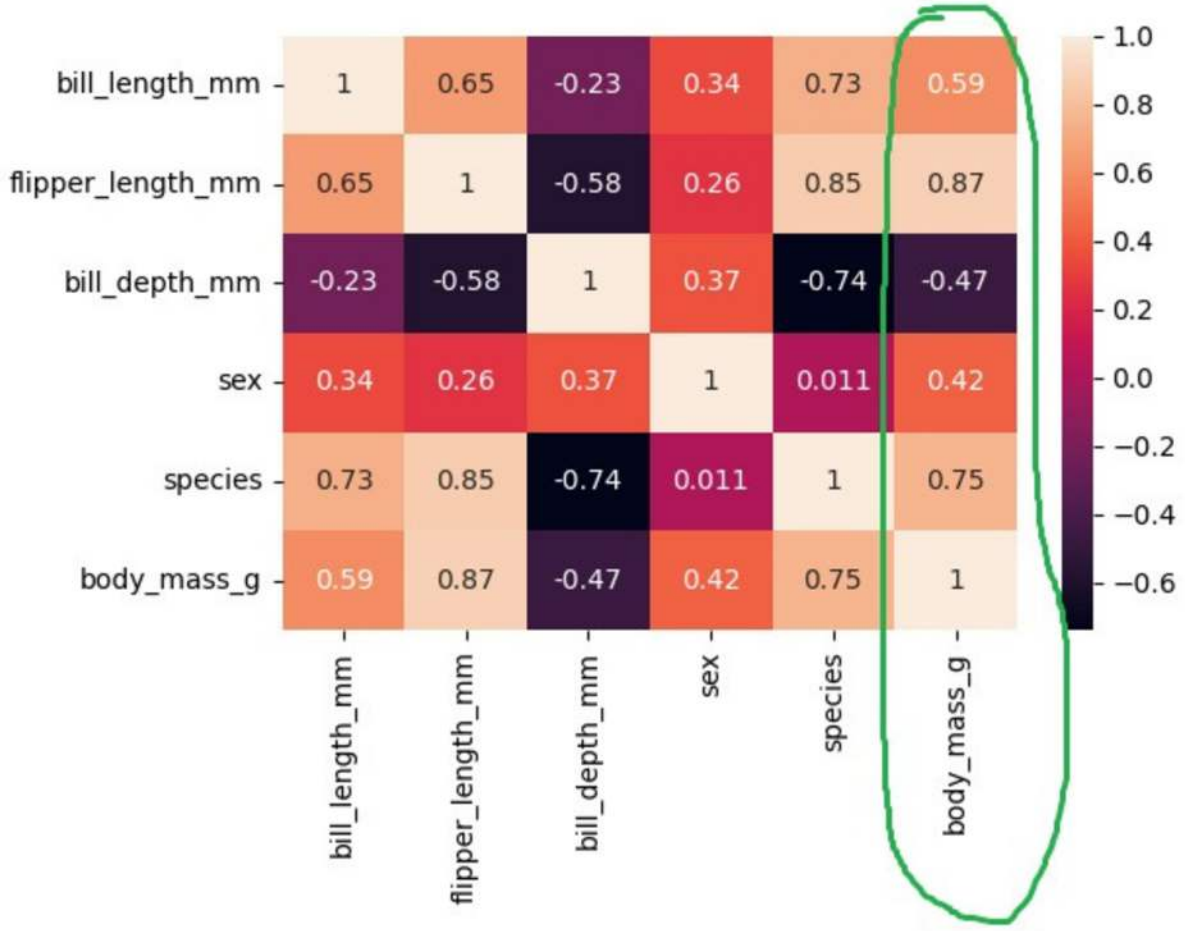
   Using seaborn charts and Pearson's correlation, the program only requires the most vital inputs.
5. Modern machine learning techniques for accurate output data

   Random Forest algorithm, as suggest by cited journal articles, has proven to be very accurate.

Effective Storytelling using Visualizations

Discussed below will be various visualizations and elements of effective storytelling supporting the data exploration and preparation, data analysis, and data summary, including the phenomenon and its detection

```
sb.heatmap(corr, annot=True)
plt.tight_layout()
plt.show()
```



The above visualization is a good summary of the elements used to explore and prepare the data for later use. It was important to use only highly correlated values in order to build a reliable predictive method. One data phenomenon that was discovered using this chart was the correlation of the bill depth to weight of the penguins. The customer, Arctic Solutions, needed a predictive model built on the least amount of input variables, while still producing an accurate output. Flipper length and bill length were decided as the two variables as the customer thought adding 'sex' and 'species' could lead to inputting incorrect data. These two categories had the potential to be based on visual estimates only, and not a measurable input. Arctic Solutions asked for only two inputs, and it was decided that they both should be unbiased, measurable input variables.

Product Accuracy

Pandas and Penguins believes that the model performed well and was a success. A critical unbiased metric to prove that was the use of the R^2 score, also known as the coefficient of determination.

```
data = Descriptive.values
X, y = data[:, :-1], data[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

regr = RandomForestRegressor(random_state=0)
regr.fit(X, y)
y_pred = regr.predict(X_test)
```

The image above shows the code block which builds the variables used in the R^2 score calculation.

```
print("\nr2 Score = ", r2_score(y_test, y_pred), "
```

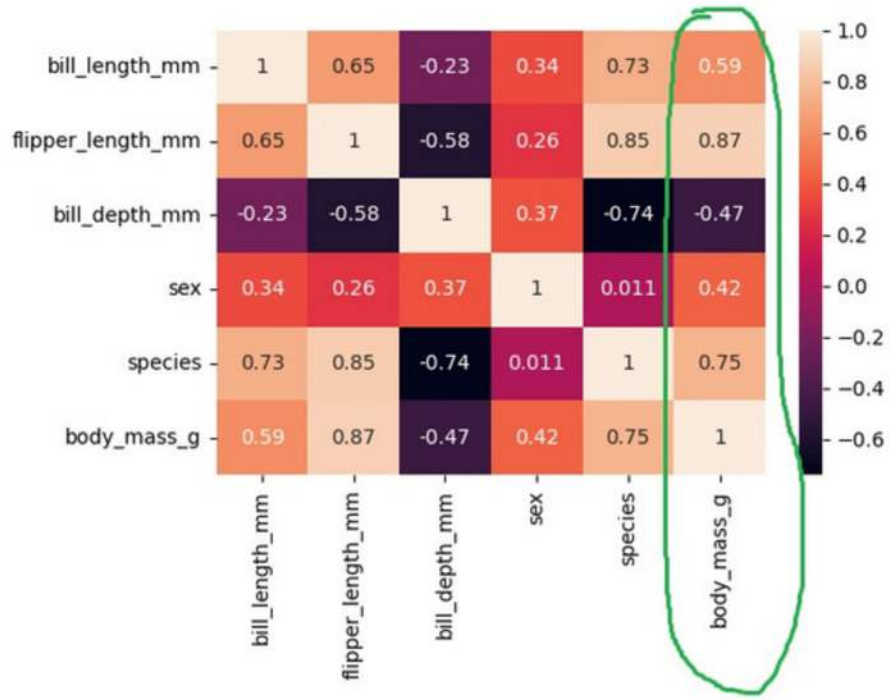
```
Y

r2 Score =  0.971484511737117

Please refer to PyCharm SciView for visualizations

------------------------------
Program finished, Thank you!
```

As the data product user interface is near its exit point, the R^2 score is printed. This tool, which was provided by the sklearn library, allows for a verification of the models performance in regards to accuracy of output. Per discussions between Pandas and Penguins and Arctic Solutions, it was agreed that a 97% or greater accuracy rate for the product is acceptable for use.

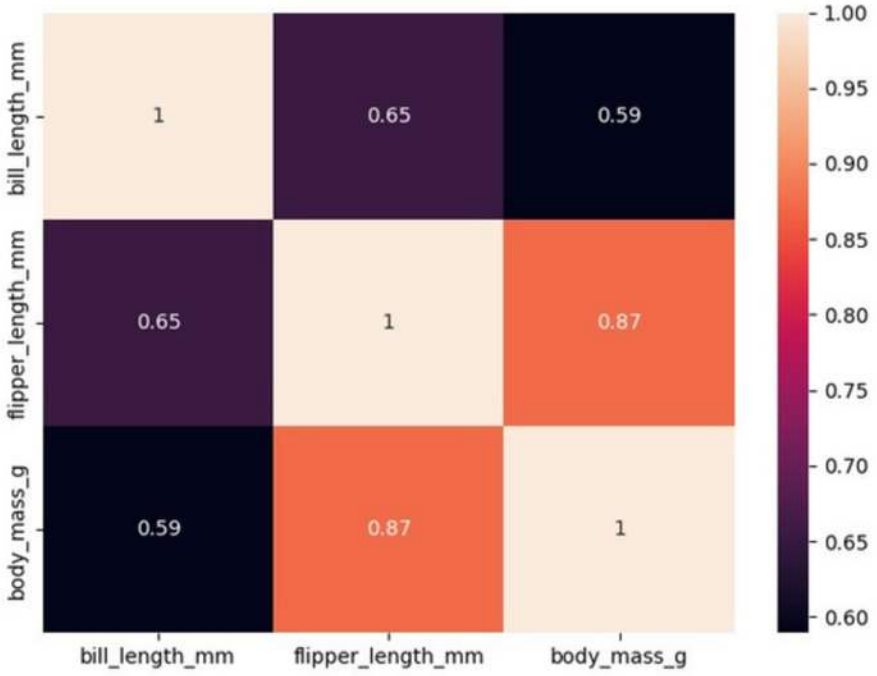
Product Testing

As per the provided plans in section B, the results of the data product testing, the revisions, and optimizations and their screenshots will be discussed below.

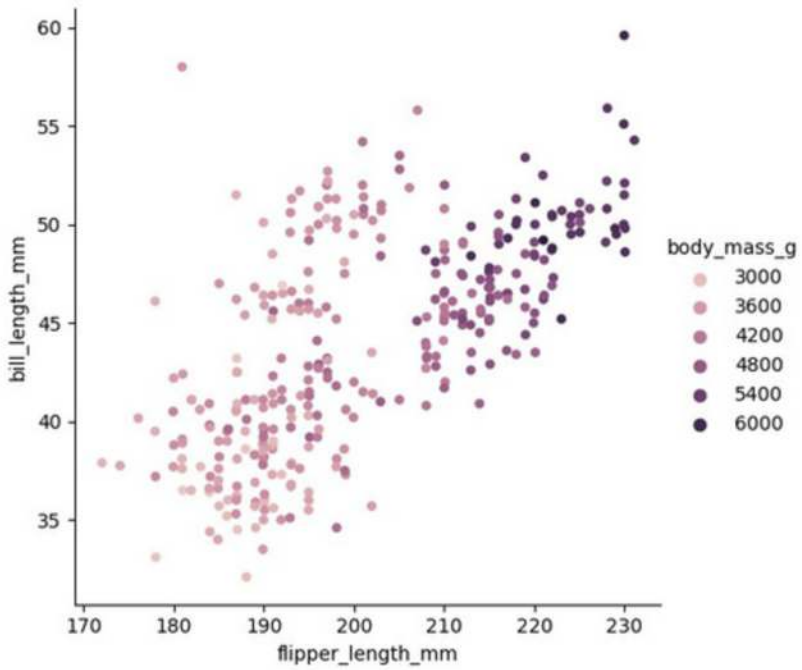
Using Pearson's correlation coefficient on the input variables will validate the decision of providing a tool set that only includes input variables that lead to great accuracy. This will provide a good analysis of a balance between potential user inputs and their ability to provide an accurate penguin weight output as required by the customer



Above image is generated using the raw dataset.



Above image is generated after data product testing, revisions and optimizations. A great improvement.



The above image shows the final variable and output relationships. Thanks to various optimizations in the dataset, the finished data product is able to provide a very accurate result.

The algorithm output will be analyzed using an R^2 score, known as the coefficient of determination. This will validate and verify the requirement of the customer for accurate output data given a simplified minimal input process.

```
data = Descriptive.values
X, y = data[:, :-1], data[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

regr = RandomForestRegressor(random_state=0)
regr.fit(X, y)
y_pred = regr.predict(X_test)
```

```
print("\nr2 Score = ", r2_score(y_test, y_pred), "
```

```
Y

r2 Score =  0.971484511737117

Please refer to PyCharm SciView for visualizations

------------------------------
Program finished, Thank you!
```

The above images show the code for the random forest algorithm and the R^2 score. This high accuracy score was obtained using data testing, revisions, and optimization.

Source Code and Executable File(s)

The entire program can be run from the included file c964.

The folder c964 will contain 2 files.

-”Main.py” Includes data cleaning, algorithm prediction, and user interface.

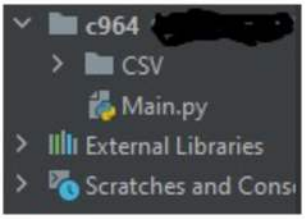
-”penguins.csv” Data file in original format, will be called in Main.py file.

User Guide

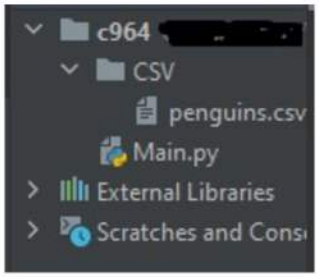
Visual Reference Keys

Development and user environment

Only two files needed



Folder structure



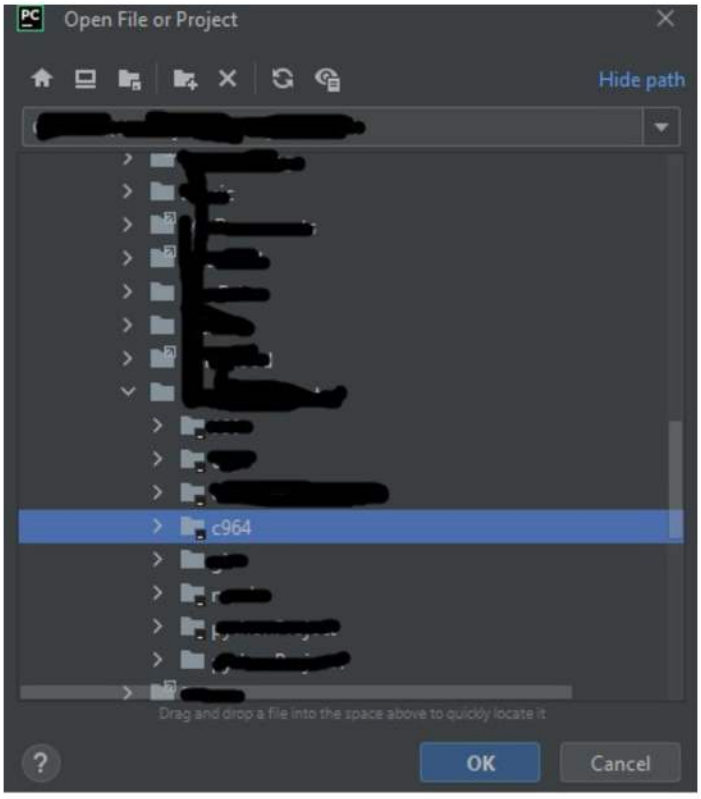
Folder expanded view

```
import pandas as pd
import seaborn as sb
from sklearn import preprocessing
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
```

Reference image of packages

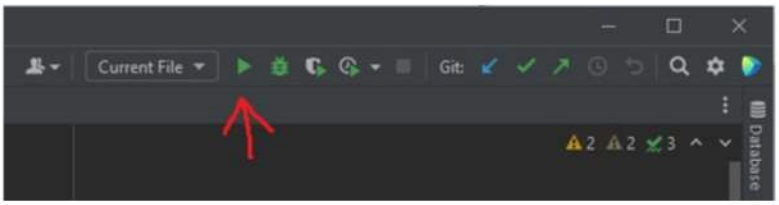
1. Download and install PyCharm from JetBrains official website.
2. Open PyCharm IDE
3. Using the menu at top of PyCharm, select “File”, then “Open”

4. Open file name “c964”

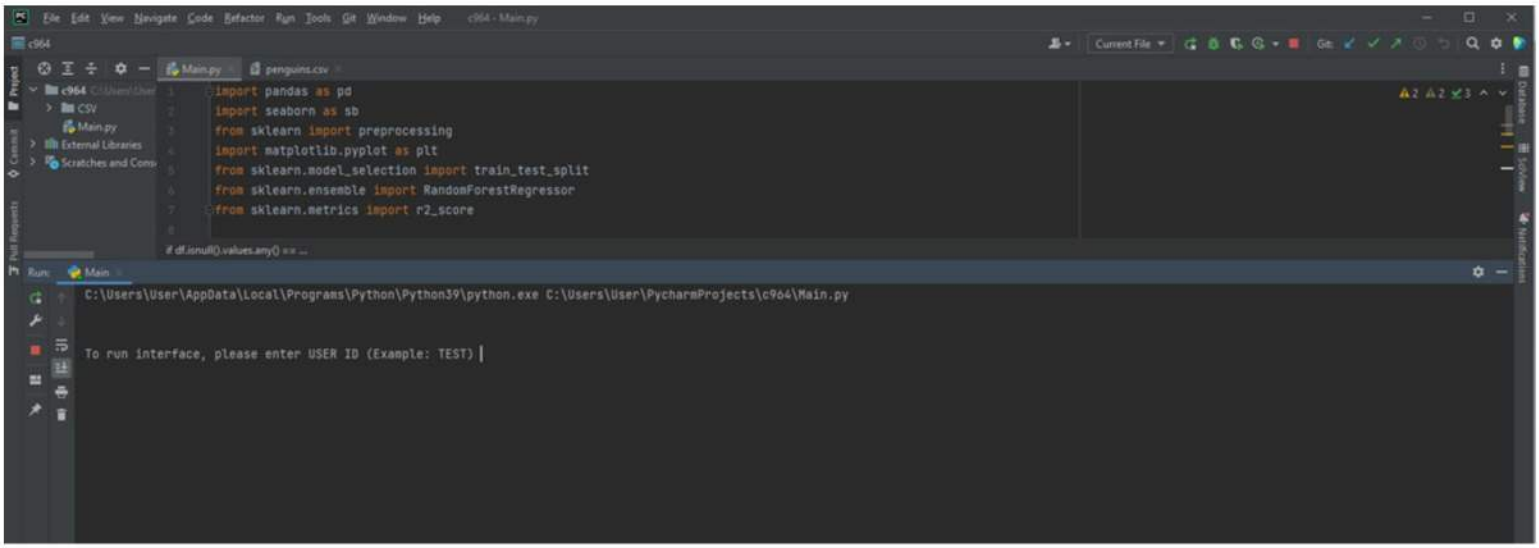


5. PyCharm will assist in the installation and initialization of the various libraries in the “Main.py” file.

6. With the “Main.py” file selected, press the green start button on the upper right hand toolbar.



7. Here is a photo reference of a fully functional running environment for the software



8. Enjoy and have fun! Thank you for trusting Pandas and Penguins LLC.

References

Gray, K. R., Aljabar, P., Heckemann, R. A., Hammers, A., & Rueckert, D. (2013). Random forest-based similarity measures for multi-modal classification of Alzheimer's disease. NeuroImage, 65, 167–175. https://doi.org/10.1016/j.neuroimage.2012.09.065

Lingjun, He; Levine, Richard A.; Fan, Juanjuan; Beemer, Joshua; and Stronach, Jeanne (2018) "Random Forest as a Predictive Analytics Alternative to Regression in Institutional Research," Practical Assessment, Research, and Evaluation: Vol. 23 , Article 1. DOI: https://doi.org/10.7275/1wpr-m024

Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., & Feuston, B. P. (2003). Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. Journal of Chemical Information and Computer Sciences, 43(6), 1947–1958. https://doi.org/10.1021/ci034160g