STELLA'S CHOICE

Dog Breed Recommendation Website

Issam Ahmed

Computer Science Capstone
Western Governors University
July 2020

Table of Contents

Letter of Transmittal

June 22, 2020

Chrissy Brando
Executive Director, Stella's Shelter
4223 Farm Street
Philadelphia, PA 19133

Dear Ms. Brando,

As you're aware, roughly 30% of our intake last year was either abandoned by their owner or returned due to behavior incompatibility. Approximately 25% of the dogs residing in our shelter have been in our system for longer than 40 days, prompting financial constraints on our limited resources. The proposed solution should help increase both our adoption and retention rates.

The proposed solution will introduce a supplementary tool on our existing website where potential adopters will match their desires and lifestyle factors to different breeds. Depending on these factors, the site will generate a list of adoptable dogs matching these breeds. The website will also promote dogs predicted to be long-term residents. We hope that potential adopters will build emotional attachments and lead to an increase in adoption.

The strategic benefit includes an increase in adoption rate and retention rate while providing more space in our shelters. Additionally, we will expect to see an increase in our operations budget. Budget change is due to added incoming adoption fees, and a reduction in upkeep and maintenance costs related to long-term residents.

The projected expense to develop this solution is $25,000. The bulk of the cost is the required human resources to build the website. We are in talks with our corporate sponsor, Pet Supply Inc., in covering most of the cost. We also have a reliable team that built our main website to take on this development project. The team will be able to produce a working solution before our major adoption events in the late summer.

Our mission is to find every dog a loving home, and this solution will be a significant benefit to the shelter's objectives and goals. Please let me know if you have any further questions about the proposal. I hope to hear your response soon.

Sincerely,

Issam Ahmed
Development/Outreach Manager

Section A: Project Recommendation

Problem Summary

Animal control departments across the country face the same issue: too many stray or abandoned dogs, many of whom are long term residents with a low chance of being adopted. The city of Philadelphia alone had approximately 5,200 intake dogs in 2019 alone. Unfortunately, while adoption rates have improved over the last decade, around 7% of these dogs were euthanized in the previous year.

Stella's Shelter is a no-kill shelter that strives to find homes for all dogs in Philadelphia. We also have the problem of too many long-term dogs residing in our shelters and foster system. Potential adopters are often seeking younger and pure-breed dogs, leaving older dogs and "mutts" in the system indefinitely. Additionally, we have numerous dogs, about 30% of intake, who either surrendered by the owner or returned due to behavioral and incompatible issues. Returned dogs are a substantial financial constraint on our limited budget. It also limits how many dogs we can take in from kill shelters and the city animal control. Stella's Choice, a web-based personality matching service, can help increase the adoption rate for these dogs by increasing the visibility and value of these lesser-desired dogs as well as help match dogs that fit with the adopter's lifestyle, increasing the retention rate.

Application Benefits

Currently, Stella's Shelter website has a search functionality but no recommendation system. Creating a recommendation system based on user input will help increase the adoption rate across the board, and we plan on giving increased visibility to less adoptable dogs.

The website will help to increase the likelihood that mixed breeds will be adopted. The recommendation will also be able to help match breed behaviors to the adopter lifestyle. A better behavioral match will help increase the retention rate of adopters and decrease returns. We will also predict how likely a dog will remain in the shelter for more than 30 days and promote them.

As stated before, the strategic benefit is an increase in our adoption rate and retention rate. Both rates increasing has many additional benefits. Firstly, the maintenance costs of keeping a dog, such as food and medical expenses, will transfer to the adopters. Most of the dogs that remain in our shelter are usually older and mixed breed dogs. There are substantial veterinary

costs associated with older dogs. Finding a home will help decrease veterinary costs. There are also administrative costs with using our foster system and the discounted financial support we provide fosters—the less time a dog spends in our system, the lower the maintenance cost and budgetary constraints.

Increasing our adoption rate draws in more money into our operations budget. As a non-profit organization, we rely on donations and adoption fees. An increase in adoptions will bring in more money. Also, we will have more room in our shelters and foster system. An increase in the operational budget, a decrease in cost, and additional space mean we can increase the number of dogs we can take in from the city and other kill shelters in the tristate, reducing euthanasia. These align with our shelter's business goals and mission statement.

Application Description

Our development team will produce a web application solution written in Python and using the Django framework. We will use several HTML, CSS, and JavaScript frameworks to facilitate production. At the front end, we will provide a clear interface for the user to select specific behavior attributes that align with their lifestyle. Top breed recommendations will be displayed, and a list of adoptable dogs will be in tabular format.

When the shelter gets a new dog, they will update the website through an admin page and fill in attributes associated with the dog. The program will then predict, using these values, whether it will remain in the shelter for less than or more than 30 days. We will use this determination to promote dogs that are at risk.

With the user-selected attributes, the algorithm will then recommend several breeds that match. Then utilizing the breed recommendation, the website will present adoptable dogs that match the top-recommended breeds. We will promote and highlight dogs that we predict will stay more than 30 days on this list. An interface will give our shelter the capability to update available dogs. The interface will have a login feature for security reasons.

On the completion of development, the application will launch on our Amazon Web Services (AWS) instance.

Data Description

We are using two datasets to form the basis of our application. The first is a set of breed attributes that we will use to match an adopter with a list of compatible breeds. The dataset used for breed recommendation will be a JSON file converted to a CSV file from Kaggle.com. The data lists dog breeds and specific behavioral attributes with a scale from one to five. As the data set incorporates cat breed information and other unnecessary information, it will go through a cleaning and normalizing process.

The second dataset consists of a list of adoptable dogs and aids our application in calculating the predicted adoption speed of our dogs. The available adoptable dog data set will also originate from Kaggle. This information is currently on a CSV file.

The admin can further manipulate the adoptable dog data by an interface, where they can add or remove dogs up for adoption.

Objectives and Hypotheses

The objective of the proposed solution is twofold. First, increase the adoption rate of dogs at high risk of not being adopted in the current system by identifying these animals and promoting them more aggressively. Secondly, improve the retention rate of all adoptions by gathering information about the adopter's preferences and lifestyle and suggesting breeds from that information. These two objectives consolidated will decrease our maintenance cost and increase our operational budget. A sub-objective is to generate an environment to encourage more adoptions and form emotional fondness for potential adopters to one of our adoptable dogs.

We hypothesize that incorporating this adoption matching tool will provide useful and accurate recommendations and predict how long a dog will stay in the shelter. Additionally, we speculate that age, and being a mixed breed are main factors driving adoption speed. The algorithm and data will go through several iteration development and training to produce precise results to increase our retention rate.

Methodology

This project development will go through the Agile method. We envision this project as an iterative process, developing on previous builds, particularly in regards to our data-handling algorithms. Using Agile, we can use feedback from stakeholders to improve the recommendation system. Finally, we can implement our website and integrate it into our main site in a shorter time frame. The time constraint is an essential factor as historical data indicate a higher intake in our shelter in late summer.

The following are the project's phases in Agile implementation:

1. **Requirement** - Gather and determine requirements for the current iteration
2. **Development** - Design and develop sites features based on the accumulated requirements
3. **Testing** - Quality assurance testing covering internal and external training, interface and usability, and black-box testing. Includes documentation development.
4. **Delivery** - Validate, integrate and deliver a working product
5. **Feedback** - Gather customer and or stakeholder feedback for revision and refinement in the next iteration cycle

Funding Requirement

The overall estimate of the project is $25,000. However, we are negotiating with our development firm for a diminished rate, which they have accommodated us before since we are a non-profit. We also have a corporate sponsorship with Pet Supply Inc. We can negotiate with them to take most of the financial burden. There will be no licensing cost since we will be using free open source frameworks. No additional maintenance cost required since our previous website maintenance budget includes the hosting fees required.

Stakeholders Impact

Other than Stella’s Shelter, the key stakeholder is the potential adopters. The website's success depends on attracting more adopters to our website and give them exceptional recommendations on the breed and current dogs available for adoption. The adopter's satisfaction will drive more people to adopt, through word of mouth or social media.

Other stakeholders include our shelters, sister rescue programs, and the city's animal control. The completion of this program will help the group's mission statements and business goal of saving as many dogs as possible.

Finally, a potential stakeholder with significant influence over the project is Pet Supply Inc. if they decide to go ahead with the sponsorship. They will have some terms and requirements in the design process. This project's success will help them drive their sales and potentially create a positive outlook on their corporation.

Data Precautions

There are no concerns about potential protected data. The only data we are applying for the algorithm breed behavior information and adoptable dogs' records. Future information accumulated from users includes their feedback, interactions, and approval of the breed recommended. No sensitive or private data collected during user interaction and the lifespan of this website. Once a user selects a dog, the site redirects to our main website, which manages private data required for the adoption process as directed by standards and regulations.

Developer's Expertise

The developer firm we are preparing to work with is Ciro-Luna Agency. They built our current adoption site, which was successful and met all our previous requirements with a significant discount. The two developers on staff who were our principal contact developers have several years of experience developing and possess relevant degrees. One developer covers the frontend requirements, and the other the backend development. Additionally, these two developers and many of their staff have adopted directly from our shelter prior and have a similar passion for helping dogs.

Section B: Technical Proposal

Problem Statement

The project's goal is to create a recommendation tool to integrate into our existing website to increase the adoption and retention rate of dogs adopted out of Stella's Shelter. Additionally, we wish to minimize the number of dogs who reside in the shelter for longer than 30 days. The website will take the user's desired characteristics and predict five of the closest breed classifications, query the database and present a list of adoptable dogs matching the breed. The focus is to promote dogs predicted to remain in the shelter longer than 30 days in this list.

Customer Summary

The customer base is anyone willing to adopt a dog, principally located in the Greater Philadelphia area. They can range from first time adopters to seasoned trainers. Ages range from college students to retirees. In general, customers are families and couples from ages 25 to 40. This tool will be especially valuable for people with breed and lifestyle requirements.

Our existing customers are usually highly enthusiastic and motivated to procure a dog and find us through social media or word of mouth. The goal is to reach people who have a slight interest and motivate them to adopt. The intention is to promote this website to attract this customer base. Therefore, the website must be intuitive and easy to use.

Existing System Analysis

There is currently no recommendation method on the main website. We do store and present fundamental data such as name, breed, age, and sex of dogs presently available on the website's database. The customer simply picks the dog which seems suitable or the "cutest." Adopters usually come predetermined in adopting popular breeds. There is a significant return rate due to these methods of selecting a dog, producing incompatibility between the owner and dogs. As stated, the goal is to minimize this by creating a predicting algorithm that pairs users to breeds based on desired attributes and lifestyles.

Advertising of dogs is a one-time social media post about recently procured dogs followed by photos and quick descriptions. Currently, there is no method to predict which dogs will remain in the shelter for a longer duration; therefore, we cannot give them extra promotional resources. The new system will fulfill this gap by utilizing historical accumulated data to predict

how long the dog will linger in the shelter. With this information, the staff can start promoting them immediately and more frequently.

The health records and a dog's duration in the shelter currently resides in spreadsheets, across multiple shelters. Incorporating this into one database will further enhance the prediction software once the website is live. This will assist in promoting dogs across our shelters and not merely the specific region.

Data

The first set of data utilized is the breed attributes and ratings for the individual breed, which is freely available through Kaggle. The file comes in a JSON format, which we convert into a CSV for readability and convenience. It will be applied to train the machine learning model and create data visuals. The data contains cat breed attributes, which are outside the scope and discarded. Some breeds with missing data will similarly be modified until we can get further information. The reduction of numerous attributes into five dominant attributes, and the breed's adult size will generate a better user experience. These attributes will not be updated or modified by the user or the admin.

The next data set used will be historical data of adopted pets. This data is also in a CSV format and will be used to train and test the prediction model. The data has four distinct adoption speeds, such as one week, or 90 days, but our aim is less than a month or more than a month. The data will be adjusted to reflect this. The data also holds cat breeds, removed as in the previous set. It includes several fields that are not required, such as ID or fee, and will not be in the model. Incorrect or incomplete data, such as inaccurate breed names, will be removed.

Considering our website will incorporate two datasets, they must be normalized and cleaned to reflect each other accurately. Mismatch breed labels will be manually corrected. Breed labels that exist in the historical data set but not in the data set of breed attributes are dismissed. This is to make sure no anomalies occur when attempting to predict a breed classification.

Project Methodology

The project will be developed through the Agile development method. Historical data shows that more adoptions occur during late summer, and we desire a working product promptly. The website will be updated and refined over several iterations by applying the Agile method.

1. **Requirement**
   - The development team will meet the product owner who represents the stakeholders and define requirements for the current iteration and scrum master will adjust the backlog.
   - Selected backlog items converted into tasks

2. **Development**
   - Design and develop the website according to the tasks
   - Daily standups to check progress and evaluate roadblocks
   - Unit test conducted by the developer for quality check and satisfy tasks expectations

3. **Testing**
   - Final unit testing for each task by developers
   - QA testing commences with black box testing covering Presentation, Business and Data layer to verify all layers operate precisely
   - System and regression testing to verify current iteration works with prior builds and layers work together
   - Usability and acceptance testing are performed with the product owner to fit stakeholder's satisfaction

4. **Delivery**
   - Validate sprint development and verify all requirements met
   - Make current iteration's website live

5. **Feedback**
   - Feedback from stakeholders and product owners are employed along with sprint retrospective to evaluate for next iteration

Project Outcomes

The following comprise the project and product deliverables.

Project Deliverables

- **Schedule** – Principal milestones with dates for principal stakeholders to track progress.
- **Mockups/Wireframes –** The frontend design proposals for the website, including the dashboard and admin page.
- **Test Plans** – It includes the testing process and procedure outlined for quality assurance and validation for the project.

- **Data Plan** – The data requirements from principal stakeholder, cleaning, and normalization plan to verify and validate standards and regulations.
- **Data Analyzes** – The analyzes of data applied to train the machine-learning algorithm. Clarify the descriptive and non-descriptive techniques used.

Product Deliverables

- **Recommendation Website** – An accessible online tool for potential adopters to receive breed recommendations and matching adoptable dogs. The website will highlight dogs predicted to remain longer than 30 days in the shelter to be promoted.
- **Prediction Algorithms –** The breed recommendation algorithm and adoption speed prediction algorithm using machine learning, trainable for future improvements, and fine-tuning.
- **Dashboard –** A dashboard for data visualization and analytics, ability test adoption speed, and see all adoptable dogs.
- **Admin Page and Security –** Secure admin access to create supplementary users and add, remove, update available dogs from the database.
- **Code and Related Files –** Transfer all Django files and database associated with the product for maintenance.

Implementation Plan

The following aspects of the implementation plan:

- **Strategy for Implementation –** After Quality Assurance testing and stakeholder validation, the product is to be deployed and hosted through AWS.
- **Phases of Rollout** – The initial iteration will produce a working website to satisfy the time constraint and essential requirements. Subsequent iterations will further enhance the product, advance prediction, and add new functionalities. These will rely on feedback from adopters and stakeholders.
- **Testing** – Testing will be administered according to plan, through several methods (Ex: unit testing, black box) during the developmental phases. Quality assurance and validation will happen in the testing phase of the iteration: final acceptance test and validation conducted before launch and final distribution.
- **Milestones** – The essential milestones are identified with stakeholders and implemented through the schedule plan. Milestones will track the budget and progress of the project.
- **Dependencies** – Stella Shelter's AWS service must be available and account information required for successful deployment.

- **Deliverables –** A functional website that adheres to major requirements after the first iteration. The site comprises the breed recommendation, adoption speed prediction, dashboard, and an admin interface. Deliverables for subsequent iterations based upon stakeholder feedback.
- **User Testing –** Conducted during Stella Shelter's late summer adoption event. The staff will use handheld devices to interact with potential adopters and accumulate input on the recommendation process. Feedback will be vital to enhance the website, make it more user friendly and enjoyable. Future iterations will use feedback left on the site and by shelter staff to improve the product.

Evaluation Plan

With the Agile methodology, there will be substantial input during the phases with the product owner's help. The developers and QA engineers will conduct verifications outlined in the testing plan. The validation process in each development phase conducted by the project manager and product owner. Project manager to communicate to key stakeholders on any significant changes or roadblocks. This method of evaluation will cover the website's functionality, usability, and any applicable standards.

Currently, there are no test data for breed attributes. To validate accuracy, the developers will require additional data, either through feedback from customers, or a new data source. Key stakeholder supports this method since the recommendation currently nudges adopters towards a group of breeds. Adoption speed prediction will utilize a confusion matrix to engage the accuracy of the model. The requirements aim for more than 50 percent accuracy. There is sufficient valid data to conduct this evaluation.

Resources and Costs

The following is the itemization of costs, programming, and human resources required for project fulfillment.

- **Programing Resources**

  Design and development will use our local dev environments operating on Windows 10 computers. Initial design wireframes will require MockFlow, a free online tool. Website and GUI mockups designed through Adobe XD and used for design validation. Frontend developers will employ a Bootstrap framework, HTML, and JavaScript to produce a functional and responsive website. The following are supplementary resources required for frontend development to build the presentation layer.

    - JQuery

- Material Kit JS
- Chart.JS

The foundation of the development process, the business and data layer, will require Django, a Python web framework. It is ideal for rapid development, which is a critical project requirement. It applies a model-template-view architecture. The backend developer will utilize this to create necessary machine-learning algorithms and the database. The following are additional resources required for backend development.

- Python 3.7
- SQLite
- Scikit-Learn
- Pip
- Numpy
- Panda
- Matplotlib
- Seaborn

The GitKraken software will handle version control for both the frontend and the backend to maintain cohesion. The team will be primarily using PyCharm IDE to develop the project as the bulk of the code will utilize Python.

- **Environmental Cost**

  We require updating to PyCharm IDE to take advantage of its built-in tools supporting Django. We additionally need to subscribe to Adobe XD software. There are no other environmental costs as the rest of the programing technology is free or already covered earlier.

Resource	Cost
PyCharm IDE Professional	$16.60 per month
Adobe XD Single App	$9.99 per month
Total:	**$26.59 per month**

- **Human Resource Cost**

  The cumulative human resource cost of the project will be $20,000. We do not see the project progressing past two iterations. As this is a discounted fixed rate project, any subsequent iteration cost will not be charged to Stella's Shelter. The following is the cost break down.

Initial Iteration – 2 Weeks Duration

Service	Cost Per Hour	Total Hours	Total Cost
Planning & Design	$150	24	$3,600
Development	$200	40	$8,000
Testing and Review	$100	16	$1,600
		Total:	**$13,200**

Subsequent Iteration – 1 Week Duration

Service	Cost Per Hour	Total Hours	Total Cost
Planning & Design	$150	8	$1,200
Development	$200	24	$4,800
Testing and Review	$100	8	$800
		Total:	**$6,800**

Timeline and Milestones

A working product is expected by the end of July 2020 before the adoption season starts. The proposed second iteration for improvement will take place after. The following are the significant milestones and timeline.

Milestone	Resources	Timeline/Duration
Budget Approval	Product Owner	7/1
Project Approval/Kick-off	Project Manager/Product Owner	7/2
Requirement Gathering	Team/Product Owner	7/3 – 7/6
Wire Frame	Frontend Developer	7/7
Website & GUI Mockups	Frontend Developer	7/8 – 7/9
Architecture Design Meeting	Backend Developer	7/7
Final Design Meeting	Frontend Developer/Product Owner	7/10
Data Analyzes & Prediction Models	Backend Developer	7/9
Templates & Frontend Complete	Frontend Developer	7/13-7/17
Backend & Database Complete	Backend Developer	7/13-7/17
Demo Meeting	Team/Product Owner	7/20
First QA Testing	QA Engineer	7/20-7/21

Milestone	Resources	Timeline/Duration
Product Validation & Approval	Project Manager/Product Owner	7/22
Initial Deployment	Backend Developer	7/22
User Testing (Adoption Event)	Shelter Staff/Team Volunteers	7/23-7/26
Feedback/Requirement Gathering	Team/Product Owner	7/27
Design/Planning Meeting	Team/Product Owner	7/27
Front-end Improvements Complete	Frontend Developer	7/28-7/30
Prediction Models Improved	Backend Developer	7/28-7/30
Website Improvement Complete	Team	7/30
Second QA Testing	QA Engineer	7/31
Final Product Validation & Approval	Project Manager/Product Owner	7/31-8/1
Redeployment	Backend Developer	8/1

Section C: Product Attributes

Data Methods

The recommendation tool allows potential adopters to select a rating from a small number of breed attributes. A descriptive method takes this user input and uses it to choose the most fitting breeds and presents it to the user. The original dataset used a large variety of attributes and was not optimal for the user interface or for generating a list of recommended breeds. We grouped similar attributes and used attribute ratings to generate a better model.

For the adoption speed model, descriptive methods were used to find correlation in data and factor out unnecessary and redundant data. The data dashboard on the website displays the different adoption speeds and the number of dogs in each, both for historical and current data.

The non-descriptive method Random Forest Classification was the primary machine-language method used to predict the top breeds matching the user's inputs. Naïve Bayes was the main component in predicting a dog's duration in the shelter (adoption speed).

Datasets

The following datasets were used:

Dataset	Purpose	Formatting
Beed attribute data	Used in the descriptive algorithm that takes in user input and returns the top 5 breeds	Obtained from Kaggle as a JSON file and converted to CSV for use training the algorithm.
Historical adoption speed data	Used to generate the predicted dog's duration in the shelter (adoption speed)	Obtained from Kaggle as a CSV file. Contains the adoption speed as well as other dog attributes.
Adoptable dog data	Used during testing as the initial list of adoptable dogs. Served as training and testing during the algorithm development phase.	Obtained from the same source of the adoption speed data, this CSV comes with the same parameters of the historical data, but without the adoption speed.

Analytics

The following analytic methods were used to manage data:

- **Data visuals:** displayed on the website's data dashboard, describe the data with bar graphs.
- **Data description:** helped factor similar attributes and narrow focus, for modeling and user experience.
- **Correlation Analysis:** implemented helped further optimize the prediction model.
- **Logistic Regression:** highlight important features affecting outcome
- **Confusion matrix:** created helped to gauge the adoption speed model's accuracy. This matrix verifies and validates the model's ability to predict.

Data Cleaning

Both datasets contained terrible, missing, or incomplete data. The datasets also included cat breeds, which was out of our scope for this project. Therefore, it was necessary to clean the data, remove undesirable data such as null values. The breed attributes and adoption speed datasets were from different sources and hence had to go through another process, so that breed names match and correlate. A manual process was applied to slightly altered or misspelled breed labels. Some of the data deemed unnecessary for prediction was retained for frontend and user experience but not utilized in training the model, while other data not used for prediction or display was simply discarded.

Data Visualization

The data analysis process utilized several visualizations, such as bar graphs, correlation maps, and a confusion matrix. Some of these visuals are displayed on the dashboard for convenience and described further in the post-implementation documentation. More bar graphs exist on the dashboard to give users the ability to explore the data. Charts include breed attributes and where the breed falls in the ranking. Additionally, the dashboard shows several factors affecting adoption speeds.

Real-Time Queries

The Django framework operates on a model-template-view architecture. Each model represents an object, in our case, either an adoptable dog or a specific breed. These models are stored and queried from an SQL database. The prediction models use CSV files for processing speed, but the resulting predicted adoption speed is stored in the database. Matched adoptable dogs in the database displayed in tabular format for users to explore. When an admin creates edits or deletes adoptable dogs, the adoptable speed re-calculates, and the

framework conducts relevant queries. The dashboard presents current data stored in the database with queries.

Adaptive Element

The predictive models use two machine-learning methods, Naïve Bayes and Random Forest. The website uses Python libraries Sklearn and Panda to process the data and return required predictions. These models can take new data and retrain prediction for any future improvements. They are also flexible if future implementations require new fields or data points. Flexibility is essential, as feedback generated in the future requires additional change and enhancement to the models.

Outcome Accuracy

Unfortunately, we currently do not have enough data to test the recommendation system and determine accuracy. It will rely on newly acquired data or feedback from adopters and the shelter staff. It is highly subjective and, therefore, difficult to measure without substantial error margins.

Training and development of the adoption speed prediction rely on historical data. There is a sufficient amount of data to test the model and gather insight on its accuracy. The program splits the data, one for training and the other for testing. Accuracy analysis is determined using the previously mentioned libraries. A confusion matrix will visualize the model behavior and prediction accuracy.

Security Measures

The benefit of using the Django framework is that it comes with standard security layers to protect the website. An SSL certificate will provide encrypted communications between browser and server. AWS has several appropriate security features that come with its hosting services. The admin interface is password-protected, with the ability to add, delete, or edit other users. This measure helps protect the database and gives several shelter locations the ability to update adoptable dogs.

Product Health Monitoring

The Django framework comes with several debugging tools and maintenance logs to track product performance. Critical maintenance logs placed on vital points of the product displayed on the console. Additionally, a file will hold records for maintenance. The log will also track users' attribute inputs for further enhancement to the prediction models. Logs automatically registered when updating the database.

Dashboard

The final product is a website that has a friendly user interface that takes in user inputs of desired attributes. The site displays the top breed recommendation with a visual display of its attribute's ratings. Four other breed recommendations will follow this with a list of adoptable dogs. Dogs predicted to be long-term residents in the list come with a simple checkmark and visually highlighted. A bar on top will provide navigation for the user to explore more of the website. A data dashboard will help the user explore the data and see the process of the prediction models. It also includes an interface for the user to test the adoption prediction model. The admin page provides an easy interface to interact with the adoptable dog database. When updating data for a particular dog the interface, it recalculates the adoption speed.

Section D: Post-Implementation Report

Project Purpose

This project has two primary goals:

- Develop a breed recommendation system based on the adopter's desires and lifestyle.
- Predict new dogs' adoption speed and promote ones predicted to remain longer than 30 days.

The end result is a web-based tool that allows potential adopters to see a list of recommended dogs based on the dog breed and user input. The tool also highlights dogs that are predicted to be long-term residents. The project's purpose was to increase the adoption and retention rate of Stella's Shelter dogs and reduce the number of long-term residents. The website also provides a data dashboard and an admin interface to add new dogs.

Datasets

Breed Attributes

The first data set consists of breed attributes and is used in the breed recommendation model. The file retrieved was a JSON file and converted into a CSV file. The file contained cat breeds and dog breeds, which required a manual process of deleting the cat columns. Below is an extract of the raw data.

breeds	Adaptability	All Around Friendliness	Exercise Needs	Health Grooming	Trainability	Adapts Well to Apartment Living	Affectionate with Family	Amount Of Shedding	Dog Friendly	Drooling Potential	Easy To Groom	Easy To Train	Energy Level	Exercise Needs	Friendly Toward Strangers
Affenpinscher	3	3	4	2	3	5	5	1	4	1	3	2	4	3	3
Afghan Hound	4	4	4	2	3	5	5	4	4	1	1	1	5	4	2
Airedale Terrier	2	4	5	3	5	1	4	2	4	1	2	4	5	5	3
Akita	3	2	4	4	4	2	5	5	1	5	1	2	4	4	2
Alaskan Klee Kai	3	3	4	3	4	3	4	4	2	2	4	4	4	4	2
Alaskan Malamute	2	4	5	3	4	1	4	5	2	1	1	4	5	5	5
American Bulldog	2	3	4	3	3	1	5	1	2	4	4	4	4	4	2
American English Cc	3	5	5	3	5	1	5	3	4	1	5	5	5	5	5
American Eskimo Dc	4	5	4	3	4	5	5	5	5	1	3	5	5	4	4
American Foxhound	2	5	5	4	4	1	4	3	5	1	5	2	5	5	4

General Health	Good For Novice Owner	Incredibly Kid Friendly Dogs	Intelligence	Intensity	Potential For Mouthiness	Potential For Playfulness	Potential For Weight Gain	Prey Drive	Sensitivity Level	Size	Tendency To Bark Or Howl	Tolerates Being Alone	Tolerates Cold Weather	Tolerates Hot Weather	Wanderlust Potential
4	4	1	4	3	4	4	3	3	3	1	2	1	3	3	2
3	3	5	4	2	3	4	1	5	5	4	2	2	5	5	5
3	2	4	5	3	5	5	4	5	3	3	4	2	3	3	4
4	2	1	3	3	3	5	4	4	5	4	5	1	5	2	4
4	2	3	4	2	2	4	2	5	4	2	3	2	5	2	4
4	1	3	4	5	3	5	3	4	4	4	5	1	5	2	5
3	1	4	3	4	2	4	2	4	4	4	2	1	2	2	3
4	2	5	4	5	3	5	3	5	5	3	5	2	4	4	5
5	4	5	4	3	3	5	4	3	4	2	4	2	5	3	3
5	2	5	3	4	3	5	5	4	2	3	5	1	4	4	5

There are 31 attributes for each breed, too numerous to be viable for modeling purposes and the user-experience. On further exploration of the data description provided from the source, five main grouped attributes have averages that cover all the other attributes. The following are the groupings and the attributes that fall under them.

- **Adaptability -** Adapts well to apartment living, Good for novice owners, Sensitivity level, Tolerates being alone, Tolerates cold weather, Tolerates hot weather
- **Exercise Needs -** Energy Level, Intensity, Exercise Needs, Potential for playfulness
- **Friendliness –** Affectionate with family, Kid friendly, Dog friendly, Friendly toward strangers
- **Health/Grooming –** Amount of shedding, Drooling potential, Easy to groom, General health, Potential for weight gain, Size
- **Trainability –** Easy to train, Intelligence, Potential for mouthiness, Prey drive, Wanderlust potential, Tendency to bark or howl

Dog size is an essential factor but included in the health and grooming average. Therefore, it was necessary to keep the size column separate, then re-average the health/grooming column without the size data. Excel provided a simple method to accomplish this.

Once the relevant attributes were determined, unnecessary columns were removed. Breed ID was added for computing purposes since the machine-learning algorithm applied needed numerical data. This Breed ID was taken from the adoption speed data below so that they would match. Two breeds had null values, which required removal. Below is an extract of the final version used for the recommendation model and the database.

Breeds Names	Breed	Adaptability	Physical Needs	All Around Friendliness	Health Grooming	Trainability	Size
Affenpinscher	0	3	4	3	2	3	1
Afghan Hound	1	4	4	4	2	3	4
Airedale Terrier	2	2	5	4	2	5	3
Akita	3	3	4	2	4	4	4
Alaskan Klee Kai	4	3	4	3	3	4	2
Alaskan Malamute	5	2	5	4	3	4	4
American Bulldog	6	2	4	3	3	3	4
American English Cc	7	3	5	5	3	5	3
American Eskimo D	8	4	4	5	4	4	2
American Foxhounc	9	2	5	5	4	4	3

Adoption Speed

Below is the raw data from the training data CSV file:

Type	Name	Age	Breed1	Breed2	Gender	Color1	Color2	Color3	MaturitySi	FurLength	Vaccinated	Dewormed	Sterilized	Health	Quantity	Fee	State	RescuerID	VideoAmt	Descriptio	PetID	PhotoAmt	AdoptionSpeed
2	Nibble	3	299	0	1	1	7	0	1	1	2	2	2	1	1	100	41326	8480853f5	0	Nibble is a	86e108	1	2
2	No Name	1	265	0	1	1	2	0	2	2	3	3	3	1	1	0	41401	3082c7125	0	I just foun	6296e5	2	0
1	Brisco	1	307	0	1	2	7	0	2	2	1	1	2	1	1	0	41326	fa90fa5b1	0	Their pregı	3422e4	7	3
1	Miko	4	307	0	2	1	2	0	2	1	1	1	2	1	1	150	41401	9238e4f44	0	Good guar	5842f1	8	2
1	Hunter	1	307	0	1	1	0	0	2	1	2	2	2	1	1	0	41326	95481e953	0	This hands	850a43	3	2
2		3	266	0	2	5	6	0	2	1	2	2	2	1	1	0	41326	22fe332bf	0	This is a st	d24c30	2	2
2	BULAT	12	264	264	1	1	0	0	2	3	2	2	3	1	1	300	41326	1e0b5a458	0	anyone wi	1caa6f	3	1
1	Siu Pak &	0	307	0	2	1	2	7	2	1	2	2	2	1	6	0	41326	1fba5f6e5	0	Siu Pak jus	97aa9e	9	3
2		2	265	0	2	6	0	0	2	2	2	2	2	1	1	0	41326	d8af7afec	0	healthy an	c06d16	6	1
2	Kitty	12	265	0	2	1	7	0	2	2	3	3	3	1	1	0	41326	1f3f36e4b	0	Very manj	7a0942	2	4

The type column represented either a dog or a cat. Using the Panda library, the data was cleaned by first removing cat data. The data included some instances of multiple dogs in a listing, and these values were removed. The shelter does not advertise multiple dogs together. Similarly, other columns deemed unnecessary were dropped, such as name, color, zip, rescuer ID, video amount, description, fee, pet ID, and photo amount. Below is the code snippet used to accomplish this:

```
# Remove Cat Data
data = data[data.Type != 2]
# Remove Dogs with quantity greater than 1
data = data[data.Quantity == 1]
# Drop unnecessary columns
data = data.drop(['Type', 'Quantity', 'Name', 'Color1', 'Color2', 'Color3', 'Fee', 'State',
                  'RescuerID', 'VideoAmt', 'Description', 'PetID', 'PhotoAmt'], axis=1)
```

Using the function, describe(), which shows descriptive statistics, we get the following.

	count	mean	std	min	25%	50%	75%	max
Age	6622.0	14.431139	22.523478	0.0	2.0	4.0	16.0	255.0
Breed1	6622.0	257.315162	83.016725	0.0	205.0	307.0	307.0	307.0
Breed2	6622.0	75.171247	124.434611	0.0	0.0	0.0	141.0	307.0
Gender	6622.0	1.576261	0.494187	1.0	1.0	2.0	2.0	2.0
MaturitySize	6622.0	1.950770	0.509085	1.0	2.0	2.0	2.0	4.0
FurLength	6622.0	1.488825	0.597365	1.0	1.0	1.0	2.0	3.0
Vaccinated	6622.0	1.628511	0.734106	1.0	1.0	1.0	2.0	3.0
Dewormed	6622.0	1.499396	0.734863	1.0	1.0	1.0	2.0	3.0
Sterilized	6622.0	1.875113	0.603784	1.0	1.0	2.0	2.0	3.0
Health	6622.0	1.041377	0.211675	1.0	1.0	1.0	1.0	3.0
AdoptionSpeed	6622.0	2.580489	1.143568	0.0	2.0	3.0	4.0	4.0

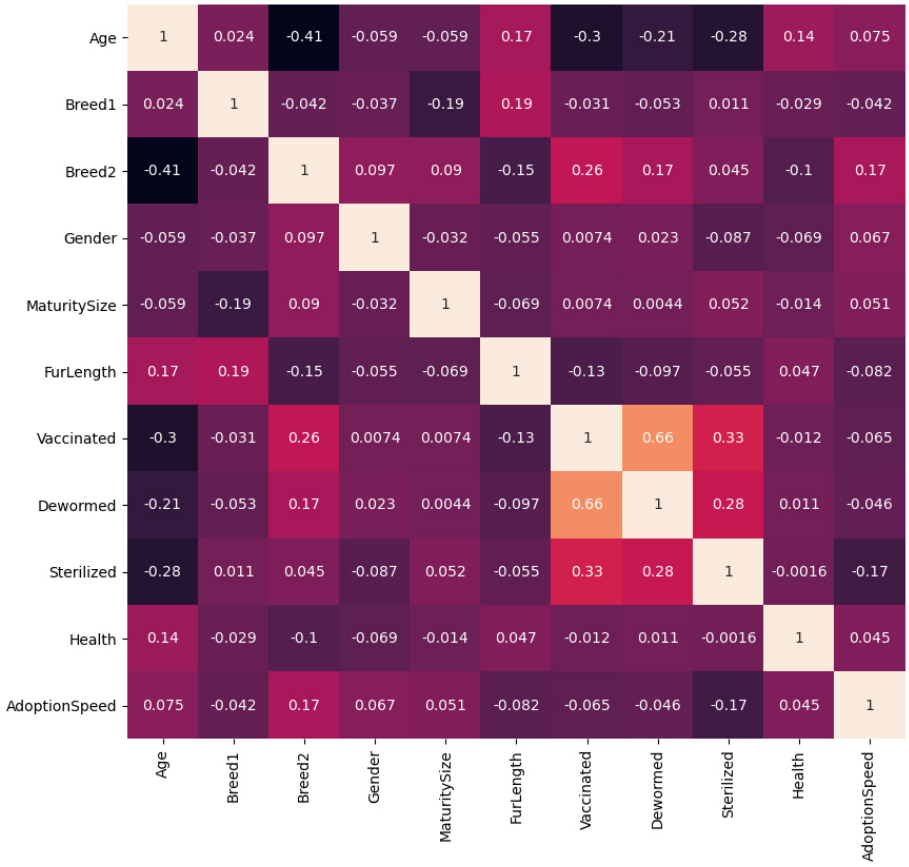
Values of "0" in column breed1 represent null values, while value 307 represents an unknown mixed breed. A similar situation arises with the vaccinated, dewormed, and sterilized columns, except the value was three, which meant "not sure." These required cleaning and removal as they might cause anomalies. Additionally, some data had matching breed values in breed1 and breed2 columns to represent pure breeds. In this case, the breed2 value was changed to zero.

Several other values required changes for standardization. For some columns, like "vaccinated" or "dewormed", the CSV uses 1 (yes) and 2 (no). These were changed to 1(yes) and 0 (no). Breed2 column represents the dog's secondary breed, values from 1-214 to map it to an actual breed, and a value of 0 if a pure breed. For the prediction algorithm, we instead desired a boolean of 0 for pure breed and 1 for mixed breed. Accordingly, any value >=1 was changed to 1.

Similarly, the adoption speed was changed from an integer value scaling from 0-4 to a boolean value of 0 (less than 30 days) and 1 (more than 30 days). The following snippet shows the code used to perform the adoption speed change and the other methods mentioned.

```
# Remove where breed1 equals 0 or 307
data = data[data.Breed1 != 0]
data = data[data.Breed1 != 307]
# if Breed1 matches Breed2 change Breed2 -> 0
data.loc[data['Breed1'] == data['Breed2'], ['Breed2']] = 0
# Remove where column equals 3 for health attributes
data = data[data.Vaccinated != 3]
data = data[data.Dewormed != 3]
data = data[data.Sterilized != 3]
# Convert 1 -> 0 and 2 -> 1
data.loc[data['Gender'] == 1, ['Gender']] = 0
data.loc[data['Gender'] == 2, ['Gender']] = 1
data.loc[data['Vaccinated'] == 1, ['Vaccinated']] = 0
data.loc[data['Vaccinated'] == 2, ['Vaccinated']] = 1
data.loc[data['Dewormed'] == 1, ['Dewormed']] = 0
data.loc[data['Dewormed'] == 2, ['Dewormed']] = 1
data.loc[data['Sterilized'] == 1, ['Sterilized']] = 0
data.loc[data['Sterilized'] == 2, ['Sterilized']] = 1
# Breed2 anything not 0 change to 1
data.loc[data['Breed2'] > 0, ['Breed2']] = 1
# Change adoption speed to reflect less than or greater than 30 days
data.loc[data['AdoptionSpeed'] == 1, ['AdoptionSpeed']] = 0
data.loc[data['AdoptionSpeed'] == 2, ['AdoptionSpeed']] = 0
data.loc[data['AdoptionSpeed'] == 3, ['AdoptionSpeed']] = 1
data.loc[data['AdoptionSpeed'] == 4, ['AdoptionSpeed']] = 1
```

The following is a correlation heat map to assess if there is any 1 to 1 relationship in the data.



Most of the correlation exists between the vaccinated, dewormed, and sterilized columns. The correlation makes sense because most dogs get all three done simultaneously and is standard in most shelters—however, because the highest correlation value is only 0.66, no changes were required.

Finally, using the Panda functions, ninique(), and isnull(), there was no single unique value or null value in any of the columns. The extract of the final cleaned data is below.

Age	Breed1	Breed2	Gender	MaturitySize	FurLength	Vaccinated	Dewormed	Sterilized	Health	AdoptionSpeed
14	189	0	0	3	1	0	0	1	1	1
5	128	1	0	2	1	0	0	0	1	1
48	205	0	1	2	2	0	0	1	1	0
60	195	0	0	2	2	0	0	0	1	1
36	109	0	1	3	2	0	0	1	1	0

Data Product Code

Breed Attributes

For the breed recommendation, the scope did not require an in-depth and accurate result, but just a group of breed recommendations that match the users' input the best. Feature importance methods were not applied as the desired goal was to utilize all six fields and for the user to consider. Graphs in a later section show the number of breeds that share similar trait ratings.

After experimenting with several different machine learning methods, the Random Forest Classifier gave the best-grouped results and hence implemented. The following code snippets are the preprocessing, normalizing, and Random Forest applied.

```
# Preprocessing
breeds['Breed'] = pd.cut(breeds['Breed'], bins=213, labels=breed_id)
label_breed = LabelEncoder()
breeds['Breed'] = label_breed.fit_transform(breeds['Breed'])
x = breeds.drop('Breed', 1)
y = breeds['Breed']
# Scale
sc = StandardScaler()
x = sc.fit_transform(x)
# Create RandomForest
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(x, y)
```

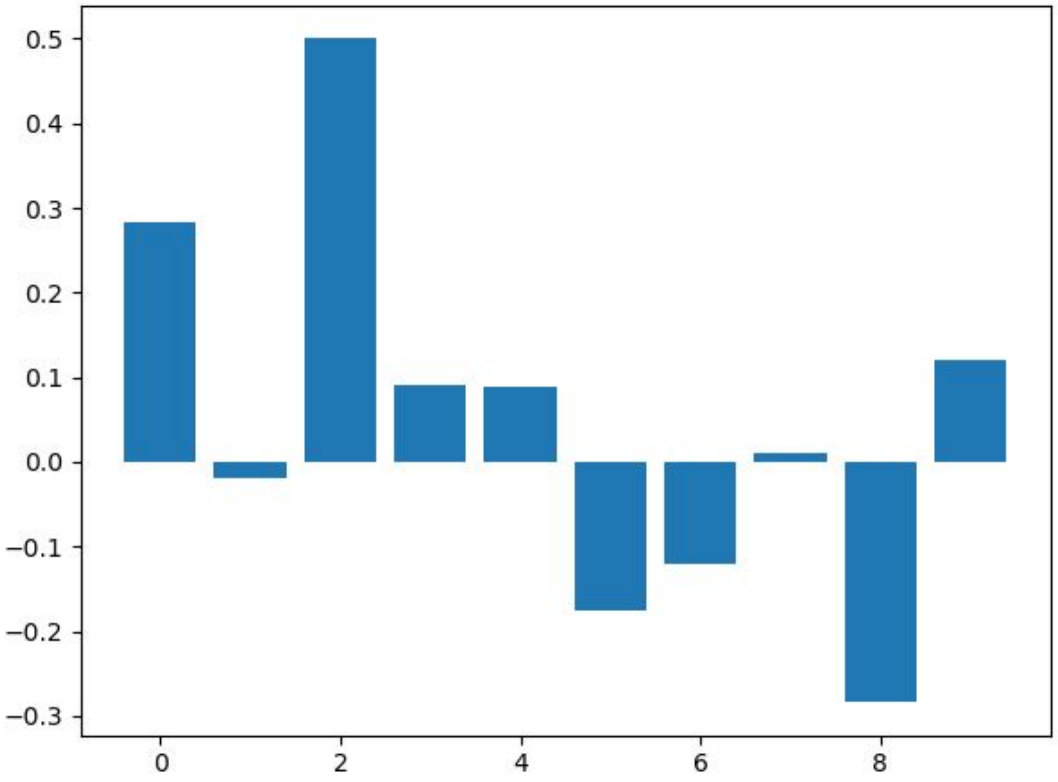
The recommendation method takes the user inputs, scales it, and returns an array of predicted breed IDs. Then a merge method and sort method returns the top five breed names that are then used by the GUI.

```
# Get values to predict
to_predict = [[adaptability, energy, friendliness, health_grooming, trainability, size]]
# Scale values to predict
to_predict = sc.transform(to_predict)
# Predict for all breeds
id_all_predicted = rfc.predict_proba(to_predict)[0]
# Get the top five closest match
predicted_array = first_five(merge(breed_names, id_all_predicted))
return predicted_array
```

Adoption Speed

The cleaned data first went through a preprocess and normalized. Next, a descriptive method, logistic regression model, is applied for further analysis of the data. This analysis is essential to highlight the features that are more relevant to the prediction outcome. It provided better insight and improved the prediction model. The following is the code and plotted graph.

```
x_lr = sc.fit_transform(x)
lr = LogisticRegression()
lr.fit(x_lr, y)
importance = lr.coef_[0]
plt.bar([x for x in range(len(importance))], importance)
plt.show()
```



The positive bars show greater relevance, while the negative show weaker relevance. The weaker factors, breed1, fur length, vaccinated, and sterilized, are discarded. The data set was then renormalized and split into a training and testing set.

```
# Drop low relevance columns
data = data.drop(['Breed1', 'FurLength', 'Vaccinated', 'Sterilized'], axis=1)
x = data.drop('AdoptionSpeed', 1)
y = data['AdoptionSpeed']
# Spilt Data -> train & test
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=13)
# Scale/Normalize
sc = StandardScaler()
sc.fit(x_train)
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

The predictive model uses Gaussian Naïve Bayes (a non-descriptive method) on the training data.

```
gnb = GaussianNB()
gnb.fit(x_train, y_train)
predict_gnb = gnb.predict(x_test)
accuracy = accuracy_score(y_test, predict_gnb)
print(accuracy*100)
```

The resulting accuracy was 64% by applying the testing set. The code used on the website is a simplified "summary" of the above code. That method returns either zero or one, representing less than or greater than 30 days.

Hypothesis Verification

The first hypothesis was that a machine-learning algorithm could give useful recommendations. Even though there was not enough data to subjectively assess the accuracy of the recommendation model, it was able to produce five breeds that adequately represented the user input.

The second hypothesis was that the model could predict how long a dog will stay in the shelter. As shown in the previous section, the prediction model had an accuracy of about 64%, above the required 50% level. Further data supports the fact that age and being a "mutt" are significant factors in determining the adoption speed. The graphs explaining these factors'

impact are available in a later section. The logistic regression graph also helped to highlight other factors, such as gender and general health.

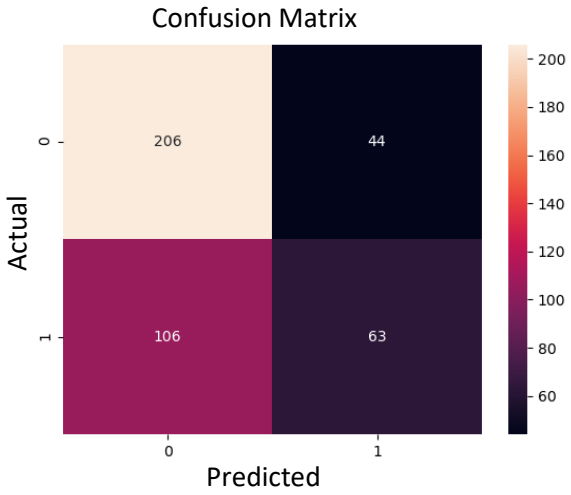
Therefore, the models can confirm the hypothesis to be accurate as they performed as desired by the clients.

Accuracy Analysis

As stated earlier, the adoption speed prediction accuracy was at about 64%. By splitting the training data for testing, the following classification report and confusion matrix highlight and validate results. The confusion matrix also shows that the model predicts speed 1 (more than 30 days) wrong more often than speed 0 (less than 30 days).

Classification Report

	precision	recall	f1-score	support
0	0.66	0.82	0.73	250
1	0.59	0.37	0.46	169
accuracy			0.64	419
macro avg	0.62	0.60	0.59	419
weighted avg	0.63	0.64	0.62	419

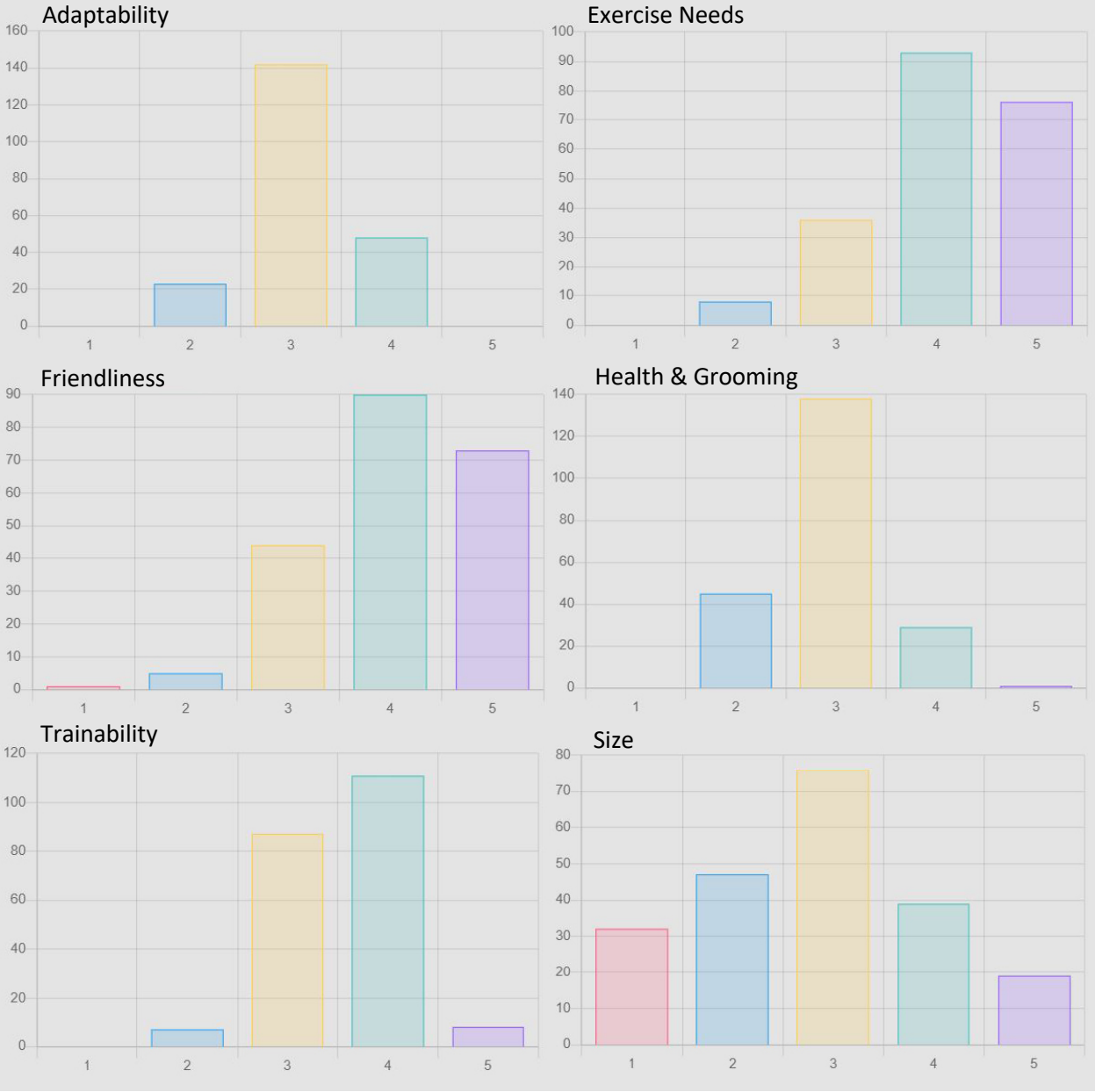


Effective Visualization and Reporting

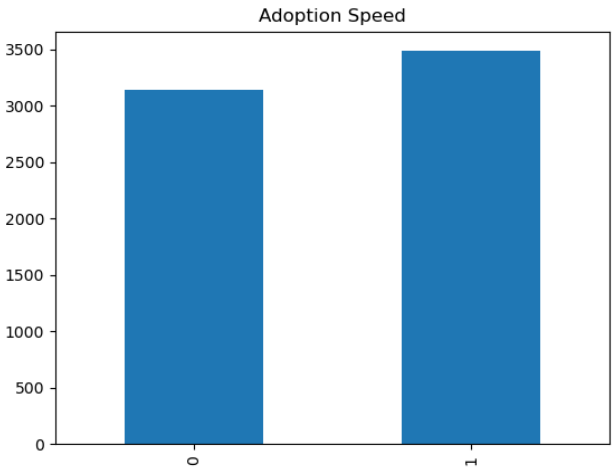
There are a couple of visualizations already used in the document to help analyze and understand the data. The following is a summary of three in the earlier sections.

- **Correlation Heat Map** used to determine if any 1 to 1 correlation existed in the data might skewer the prediction model.
- **Logistic Regression Bar Chart** – used to highlight important factors influencing the prediction model and help fine-tune it.
- **Confusion Matrix –** used to determine and assess model's accuracy.

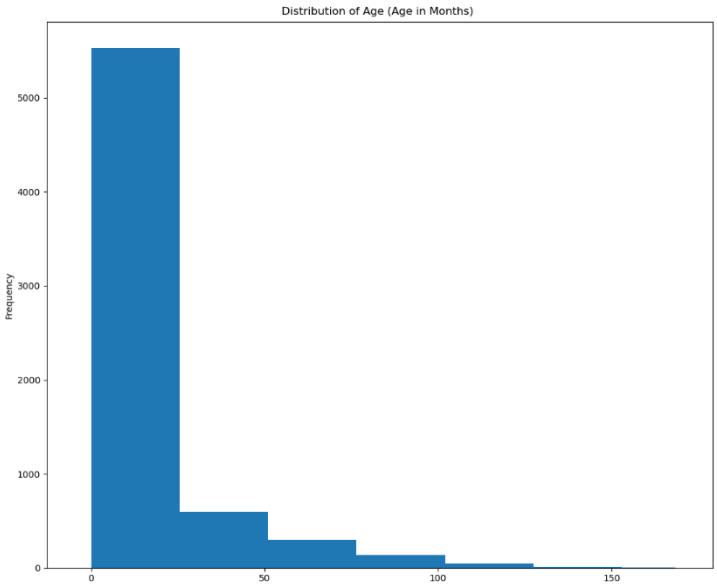
The program used SQL and a loop to aggregate breeds in similar ratings for each of the attributes found in the breed data. The returned information created data visuals to explore them further. Graphs created with JavaScript are below and displayed on the dashboard.

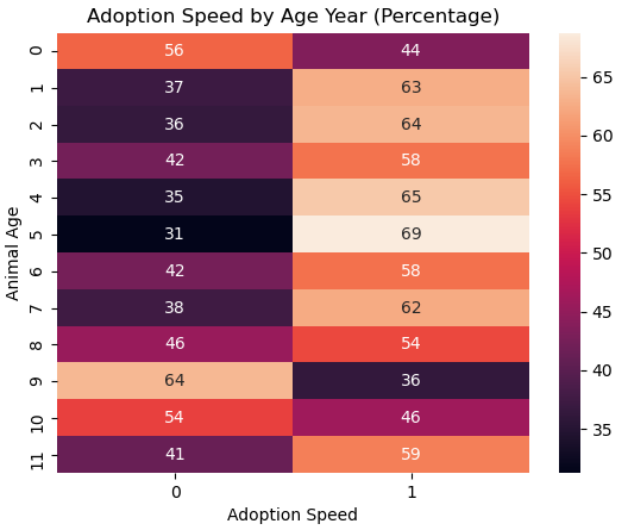


The graphs reveal that most breeds tend to be generally friendly and have high exercise needs. Most breeds also seem to lean on being highly trainable. The rest of the attributes display a semi-normal distribution.



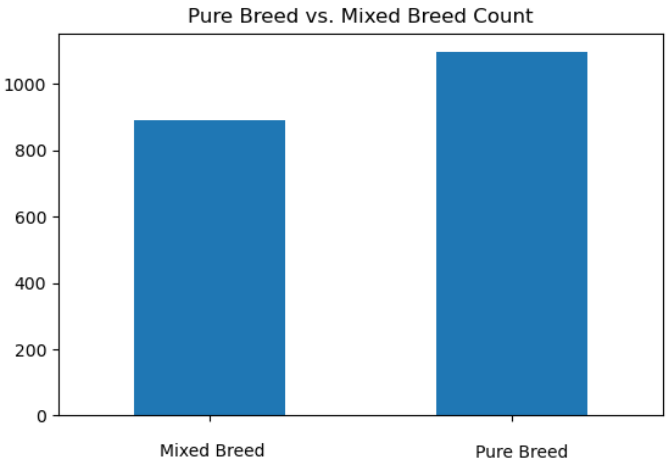
The above graph exhibits the distribution of adoptable speeds in the data. As mentioned before, zero represents less than 30 days, while one represents longer than 30 days. It is about even, but most dogs tend to stay longer in the shelter. The following graphs were created with python to explore the raw data on age and mixed breed for more helpful insight. These factors were included in the hypothesis, speculated to be essential in determining the adoption speed.

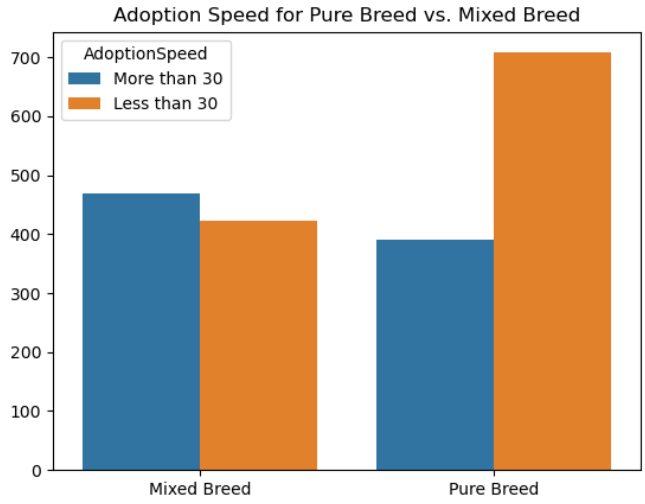




The age distribution shows a large portion of data is young dogs. Using the heat map, young dogs, especially under a year old, are more likely to be adopted under 30 days. Dogs from ages one to eight tend to stay in the shelter longer. This data confirms that adopters tend to go for puppies over adult dogs, as speculated in the hypothesis. Surprisingly, dog's aged nine and ten have an increased likelihood of getting adopted early. A possible explanation is that some

adopters have a soft spot for elderly dogs. Additionally, they require much less attention and effort, suitable for elderly adopters such as retirees.





The data has slightly more pure breeds than mixed breeds, as seen in the above distribution graphs. However, when comparing adoption speeds, pure breeds are more likely to be adopted under 30 days. The second graph strengthens our hypothesis that adopters tend to adopt pure breeds.

Visualization created was essential to explore the data and create better prediction models. Users can find these graphs on the website's data dashboard.

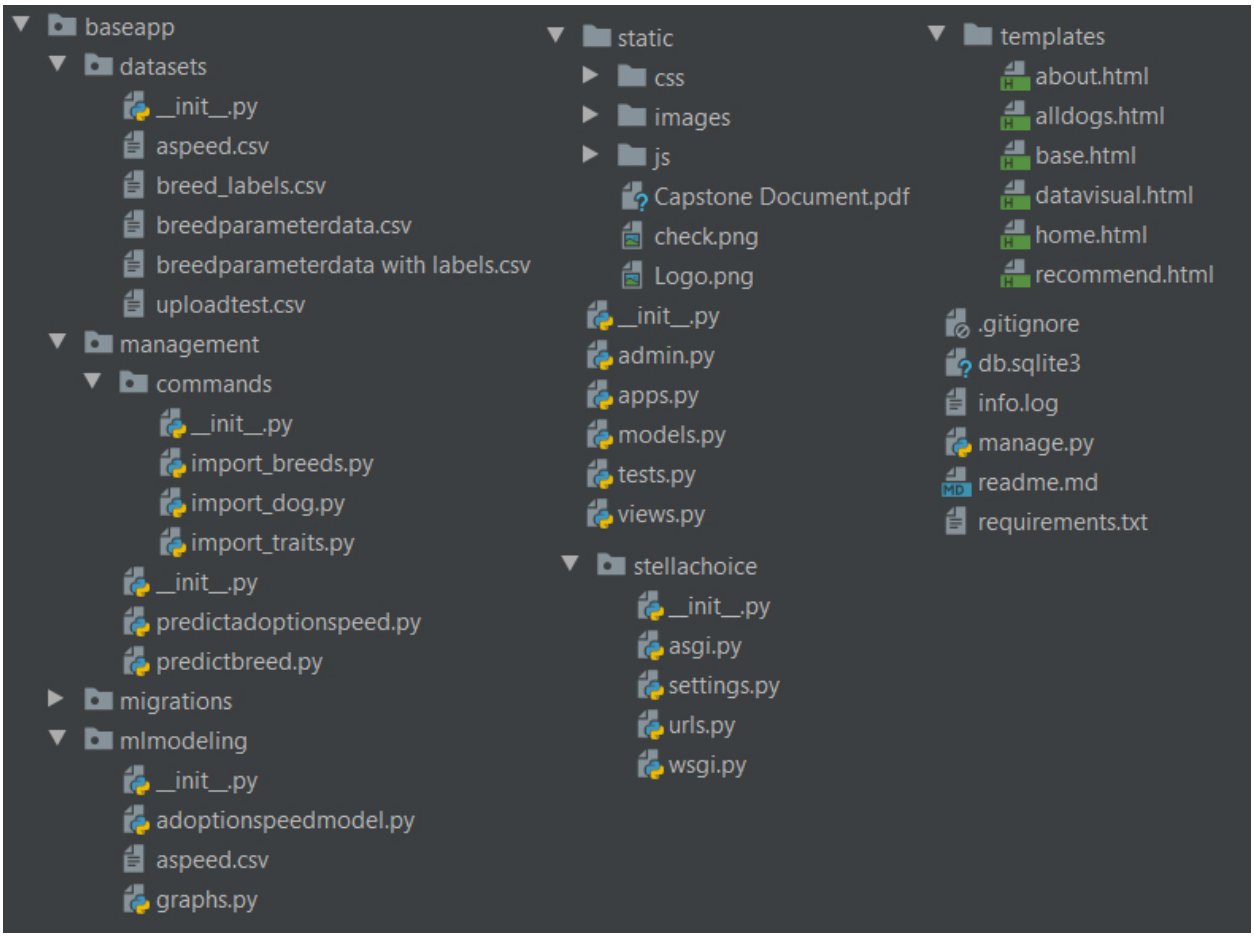
Application Testing

Several testing methods were applied while creating the website. The confusion matrix was used to test the prediction model's performance. The core functionalities of the prediction models were tested using white-box and unit tests.

Completing each website's module was followed by a unit test and black-box tests to ensure functionality. Integration and functional tests were then applied to confirm that all the separate modules worked together. A regression test followed any new changes or updates made to the website. Finally, a usability test, followed by an acceptance test, was applied by the product owner to confirm it was user-friendly and meet the project's requirements.

Application Files

The Django framework comes with several files to aid in constructing a website. Below is the file directory showcasing the files associated with the project.



The following are the summary and description of the folders and files pertinent to the project—some Django specific files related to the website settings are not described. These files are standard when installed and not edited for the project.

- **baseapp –** main application containing websites logic
  - **datasets** – Data used for prediction models and to populate database
    - aspeed.csv – training data for adoption speed prediction
    - breed_labels.csv – breed names
    - breedparameterdata.csv – breed parameters for recommendation model
    - breedparameterdata with labels.csv – above file with headers

        - uploadtest.csv – populate database with adoptable dogs
    - **management** – Folder containing predictions algorithms and scripts
        - **commands –** scripts for populating database
            - import_breed.py – import breed names
            - import_dog.py – import adoptable dogs and predict adoption speed
            - import_traits.py – import traits associated with breeds
        - predictioadoptionspeed.py – simplified method to return adoption speed
        - predictbreed.py – returns breed recommendation
    - **migrations –** SQLite and model related migrations
    - **mlmodeling –** prediction models build and analysis
        - adoptionspeedmodel.py – prediction model with data analysis
        - aspeed.csv – training data for adoption speed modeling
        - graphs.py – graphs for data visuals
    - **static –** required images, CSS, JS files for website to function
    - admin.py – admin rights on models and additional settings
    - apps.py – app settings
    - models.py – models to be used on the website (User, Dog, Breed)
    - views.py – logic used for frontend and GUI
- **stellachoice** – website and Django settings
    - settings.py – used to create logging and maintenance files
    - urls.py – site navigation settings
- **templates** – html templates
    - about.html – about page template
    - alldogs.html – outputs all dogs
    - base.html – base html settings (meta, link, nav, script)
    - datavisual.html – data dashboard template
    - home.html – home page template
    - recommend.html – recommendation and adoptable dogs' template
- **.**gitignore – version control ignored files
- db.sqlite3 – database
- info.log – maintenance log
- manage.py – Django management
- readme.md – installation notes
- requirements.txt – required libraries

User's Guide

Website Guide

Link: www.issamahmed.com/stellaschoice/

On the home page, the user inputs desired values and submit them. On submitting the form, we then direct the user to the recommendation page. This page displays breed recommendations and a list of adoptable dogs. Highlighted dogs represent dogs predicted to be long-term residents.

The about page is an explanation of the site's purpose and goals. The data dashboard allows the user to explore the data with visuals and examine the adoption speed prediction. The login button on the top right will lead the user to the admin page. The following is the required credentials.

> **Username:** admin@stella.com
>
> **Password:** admin

Here the user can add and edit other users or adoptable dogs. After filling out the necessary fields, the dog is saved or updated in the database with a new predicted adoption speed. Finally, the user can see all the adoptable dogs currently in the database using the All Dogs navigation link.

Installation Guide

This app requires Python 3.7 or higher and pip for dependency control. See requirements.txt for all Python dependencies for this project. Some libraries might have additional build dependencies.

1. Clone project into your working directory

2. Create your virtual environment:

- /path/to/python3.7 -m venv /path/to/project_root/venv
- (Optional) If using Pycharm, add the new venv interpreter to project settings (Preferences -> Project -> Project Interpreter -> select newly created interpreter to the list)

3. Ensure you're "sourced" into your virtual environment with source:

- /path/to/project_root/venv/bin/activate
- Your terminal should look something like:
    (venv) {machinename}:StellasChoice {user}
    - (venv) is prepending your typical console output

4. Install Python dependencies using:

- pip install -r /path/to/project_root/requirements.txt
- As stated above, some dependencies might have additional system requirements.

5. If using database file (db.sqlite3) skip this step and step 6. To create your Django migrations and SQLite database use the following Django commands:

- python manage.py makemigrations
- python manage.py migrate

6. Populate the database with the following:

- python manage.py import_breeds
- python manage.py import_traits
- python manage.py import_dog

7. App can now be run with Django command:

- python manage.py runserver

Additional deployment notes for AWS are included in readme.md in the project files.

Summation of Learning Experience

The completion of the capstone was a challenging and exciting journey. It was a final test of the accumulated knowledge throughout my time at WGU. I have had some previous experience with python and web technologies, which helped me create user-interface with ease. Also, programming related classes helped me apply data structures and object-oriented principles to the site's logic and ease the website's construction. However, I was not very adept with machine-language and its application. It was interesting to research and apply a complex subject. I hope to study it more and further utilize it in the future.

The Django framework was new to me and had its hurdles. I have heard its ability for rapid development and seemed ideal for my capstone. I had to follow several tutorials to grasp the concepts and apply them to the project. I am glad I learned its fundamentals as it's a popular framework.

In conclusion, the capstone project was a great learning experience. I thoroughly enjoyed taking a simple idea of helping dogs and producing a useful data product. I feel much better prepared for developing a full application from scratch.

Section E: Sources

ACCT Philly. (n.d.). Statistics & Reports. Retrieved July 3, 2020, from http://www.acctphilly.org/about/statistics/

Baynazarov, R. (2019, January 26). Cat and dog breeds parameters. Retrieved July 1, 2020, from https://www.kaggle.com/hocop1/cat-and-dog-breeds-parameters

Brownlee, J. (2020, June 30). How to Calculate Feature Importance With Python. Retrieved July 10, 2020, from https://machinelearningmastery.com/calculate-feature-importance-with-python/

Garg, R. (2020, June 07). 7 Types of Classification Algorithms. Retrieved July 10, 2020, from https://analyticsindiamag.com/7-types-classification-algorithms/

PetFinder.my Adoption Prediction. (2018, December 27). Retrieved July 1, 2020, from https://www.kaggle.com/c/petfinder-adoption-prediction/data

Sidana, M. (2020, April 09). Intro to types of classification algorithms in Machine Learning. Retrieved July 10, 2020, from https://medium.com/sifium/machine-learning-types-of-classification-9497bd4f2e14

Simplilearn. (2020, June 13). Understanding Naive Bayes Classifier. Retrieved July 10, 2020, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/naive-bayes-classifier