**Part A – Letter of Transmittal**

Trevor Widler

Irvine, CA 92614

[Twidle1@wgu.edu](mailto:Twidle1@wgu.edu)

April 20, 2024

Director Charles Sams II

900 Ohio Dr SW, Washington, DC 20024

[CharlesSams@nps.gov](mailto:CharlesSams@nps.gov)

Dear Mr. Sams,

I am delighted to present the proposal for the implementation of our mushroom identification program. This proposal outlines the benefits, objectives, methodology, and ethical considerations of the project, with the aim of securing your approval and support for its continuation.

Our mushroom identification project addresses the critical issue of ensuring visitor safety within national parks by accurately distinguishing between toxic and non-toxic mushrooms. This initiative will empower park rangers and visitors alike to make informed decisions, mitigating the risks associated with accidental mushroom ingestion.

The proposal provides a comprehensive overview of the project, including its objectives, methodology, and anticipated impact on stakeholders. We have also outlined the ethical and legal precautions that will be implemented to safeguard sensitive data and ensure compliance with relevant regulations.

I am confident that this proposal will demonstrate the value of our project and the positive impact it can have on the safety and well-being of national park visitors. Should you require any further information or clarification, please do not hesitate to contact me.

Thank you for considering our proposal.

Regards,

Trevor Widler

Project Proposal:

Mushroom Identification Project for National Park Safety

Summary of the Problem: The identification of toxic mushrooms poses a significant challenge within national parks, where visitors may inadvertently consume poisonous varieties, leading to serious health complications or fatalities. Current methods of mushroom identification are often unreliable and time-consuming, putting visitors at risk.

Description of How the Data Product Benefits the Customer and Supports Decision-Making:

Our mushroom identification project utilizes machine learning algorithms to accurately differentiate between toxic and non-toxic mushroom species. This data product will provide park rangers and visitors with real-time information on mushroom safety, enabling them to make informed decisions when foraging or exploring the park.

Outline of the Data Product: The data product consists of a machine learning model trained on a dataset containing features such as mushroom characteristics (e.g., cap shape, color, odor) and toxicity labels. Users can input images or descriptions of mushrooms, and the model will classify them as toxic or non-toxic with a high degree of accuracy.

Description of the Data:

We will utilize publicly available datasets containing information on various mushroom species, including their physical attributes and toxicity levels. Additionally, we may collect data from park visitors through surveys or observations to enhance the model's accuracy and relevance to the local ecosystem.

Objectives and Hypotheses of the Project:

Objective:

Develop a machine learning model capable of accurately identifying toxic mushrooms.

Hypothesis:

By leveraging advanced machine learning techniques, we can create a model that outperforms traditional methods of mushroom identification and significantly reduces the risk of mushroom-related incidents in national parks.

Outline of the Project Methodology:

Data Collection:

Gather datasets on mushroom characteristics and toxicity levels.

Data Preprocessing:

Clean and preprocess the data to prepare it for model training.

Model Selection:

Experiment with various machine learning algorithms to identify the most suitable model for mushroom identification.

Model Training:

Train the selected model on the prepared dataset to learn patterns and correlations between mushroom features and toxicity.

Model Evaluation:

Assess the performance of the trained model using validation datasets and performance metrics.

Deployment:

Integrate the trained model into a user-friendly application for park rangers and visitors to access.

Funding Requirements:

We estimate the project's total cost to be $450,000, covering expenses such as data acquisition, infrastructure, personnel, and ongoing maintenance and support.

Impact of the Solution on Stakeholders:

Park Rangers:

Enhanced ability to identify toxic mushrooms and mitigate risks to visitor safety.

Visitors:

Access to reliable information on mushroom safety, reducing the likelihood of accidental ingestion.

National Park Service:

Improved safety protocols and risk management strategies, fostering a safer park environment for all.

Ethical and Legal Considerations:

Data Privacy:

Strict protocols will be implemented to ensure the confidentiality and anonymity of any data collected from park visitors.

Compliance:

The project will adhere to all relevant legal and regulatory requirements, including data protection laws and ethical guidelines for research involving human subjects.

Expertise Relevant to the Solution Proposed:

Our team possesses extensive experience in machine learning, data analysis, and software development, with a proven track record of delivering innovative solutions to complex problems. We are committed to leveraging our expertise to ensure the success of this project and the safety of national park visitors.

Thank you for considering our proposal. We look forward to the opportunity to collaborate with the National Park Service on this important initiative.

Sincerely,

Trevor Widler

**B – Executive Summary**

Decision Support Problem or Opportunity:

We are addressing the need for efficient and accurate decision support in identifying toxic mushrooms within national parks, aiming to enhance visitor safety and mitigate the risks associated with accidental ingestion.

Description of Customers and Fulfillment of Needs:

Our primary customers are park rangers and visitors, who require reliable information on mushroom safety to make informed decisions while exploring the park. This product will fulfill their needs by providing real-time identification of toxic mushrooms, empowering them to avoid potential health hazards.

Existing Gaps in Data Products (if applicable):

Current methods of mushroom identification rely on manual inspection and are often unreliable and time-consuming. Our data product bridges this gap by leveraging machine learning algorithms to provide accurate and efficient mushroom classification.

Data Available or Needed for Data Product Lifecycle:

We will utilize publicly available datasets on mushroom characteristics and toxicity levels, supplemented by data collected from park visitors through surveys or observations. This data will support the entire lifecycle of the data product, from model training to ongoing validation and improvement.

Methodology for Data Product Design and Development:

Our methodology involves rigorous data preprocessing, model selection, and evaluation, guided by best practices in machine learning and data analysis. We will iteratively refine the model based on feedback and validation results to ensure its effectiveness and reliability.

Deliverables Associated with Design and Development:

Deliverables include a trained machine learning model for mushroom identification, an intuitive user interface for accessing and utilizing the model, and comprehensive documentation to support implementation and maintenance.

Plan for Implementation and Anticipated Outcomes:

Implementation will involve integrating the data product into existing park management systems and providing training for park personnel on its use. Anticipated outcomes include improved visitor safety, reduced incidents of mushroom-related illnesses, and enhanced decision-making capabilities for park staff.

Validation and Verification Methods:

We will validate the developed data product through rigorous testing against known datasets and real-world scenarios, ensuring its accuracy and reliability. Feedback from users and ongoing monitoring of performance metrics will further validate its effectiveness in meeting customer requirements.

Programming Environments, Costs, and Human Resources:

We will utilize Python programming language and libraries such as scikit-learn, pandas, and seaborn for model development. Costs will primarily involve data acquisition and infrastructure maintenance. Human resources required include data scientists, software developers, and domain experts in mushroom identification.

Projected Timeline:

Start Date: April 1, 2024

End Date: October 31, 2024

Milestones:

Data Collection (April 1, 2024 - April 15, 2024)

Model Development (April 16, 2024 - May 15, 2024)

User Interface Design (May 16, 2024 - June 15, 2024)

Testing & Validation (June 16, 2024 - August 15, 2024)

Deployment (August 16, 2024 - October 31, 2024)

Duration for Each Milestone:

Approximately 1 month for Data Collection and Model Development, 1 month for User Interface Design, 2 months for Testing & Validation, and 2.5 months for Deployment.

Dependencies:

Data Collection precedes Model Development; Testing & Validation precedes Deployment.

Resources Assigned:

Data Scientists (3), Software Developers (2), Domain Experts (1)

Our data product represents a transformative approach to addressing the critical issue of mushroom identification within national parks. By leveraging advanced machine learning algorithms and comprehensive data analysis techniques, our solution offers a comprehensive and reliable method for distinguishing between toxic and non-toxic mushroom species. This innovation has the potential to revolutionize visitor safety protocols and decision-making processes within park environments.

One of the most pressing concerns in national parks is the risk of accidental mushroom ingestion, which can lead to severe health complications or even fatalities. Traditional methods of mushroom identification, relying on manual inspection or limited field guides, are often time-consuming, error-prone, and insufficient to meet the dynamic challenges of park ecosystems. Our data product fills this crucial gap by providing park rangers and visitors with instant access to accurate information on mushroom toxicity levels.

By enabling rapid identification of toxic mushrooms, our data product empowers park visitors to make informed decisions about foraging activities, hiking routes, and camping locations. Moreover, it equips park rangers with a powerful tool to enhance their ability to respond swiftly to potential hazards and emergencies, thereby safeguarding the well-being of visitors and preserving the natural integrity of the park environment.

The significance of our data product extends beyond immediate safety concerns to encompass broader implications for ecological conservation and sustainability. By reducing the incidence of mushroom-related illnesses, we contribute to the overall enjoyment and appreciation of national park resources, encouraging responsible exploration and stewardship practices among visitors. Additionally, our solution supports park management efforts by providing valuable insights into the distribution and prevalence of toxic mushroom species, facilitating targeted conservation strategies and habitat management initiatives.

Furthermore, the implementation of our data product represents a paradigm shift in decision-making processes within national parks, fostering a culture of data-driven management and proactive risk mitigation. By harnessing the power of cutting-edge technology and interdisciplinary collaboration, we elevate the standards of safety, accessibility, and environmental stewardship in park management practices.

In conclusion, our data product represents more than just a technical solution to a specific problem—it embodies a commitment to innovation, sustainability, and the well-being of park visitors and ecosystems. Through its implementation, we aspire to create safer, more resilient, and more inclusive park environments that enrich the experiences of all who cherish and explore these natural treasures.

**C. Project Description:**

Project Purpose:

The goal of this application was to showcase to NPS the practical application of machine learning in enhancing visitor safety during mushroom foraging activities. This prototype sought to establish that NPS could effectively ensure the safety of their guests by developing a final application version utilizing a proprietary dataset crafted through collaboration with mycologists. Through successful demonstration, the application illustrated that employing a machine learning algorithm focused on the five most crucial characteristics for classification could accurately predict whether a mushroom is non-toxic or toxic with a 100% accuracy rate.

Datasets:

The dataset utilized in this application comprises over eight thousand mushrooms, each with various attributes. Initially, I visualized the data in a Jupyter notebook, creating a pandas dataframe from the CSV data and displaying the first five rows. To prepare the data for predictive algorithms, I converted the letters representing each attribute into numeric values using the Labelencoder function from Scikit-Learn.

Subsequently, the dataset was divided into two sets to facilitate training and testing of the algorithm. One set contained all attributes except for the class label, while the other contained only the class label. These sets were then split in half to create training and testing subsets, each containing an equal number of specimens. With the data thus prepared, it was ready for input into a random forest classifier.

The random forest classifier was initialized with the split data, allowing me to utilize the feature importances method. By analyzing the feature importances, I gained insights into the significance of each characteristic for classification. Visualizing these characteristics ordered by their classification significance using Matplotlib was particularly insightful. Notably, this analysis revealed that the 'veil-type' attribute had zero influence on classification, as every specimen in the dataset had a 'partial' veil-type. Consequently, this characteristic was removed from further consideration.

Data Product Code

With the dataset now streamlined and irrelevant characteristics removed, the subsequent step was to assess the accuracy of the random forest classifier in predicting class. Remarkably, it achieved 100% accuracy in predicting class when tested with all characteristics, albeit only after the removal of the 'veil-type' attribute. This represented the highest achievable accuracy before the exclusion of 'veil-type'.

Subsequently, I examined the performance of the random forest classifier when provided with individual columns of data. The algorithm's predictive accuracies for each characteristic are detailed below:

Characteristic Algorithm Predictive Accuracy

Odor 98.25%

Gill-size 75.43%

Gill-color 80.99%

Spore-print-color 86.8%

Stalk-root 64.35%

Bruises 74.59%

Ring-type 78.31%

Population 72.62%

Cap-color 59.5%

Achieving over 98% accuracy with just one attribute was promising, indicating that a combination of a few characteristics could potentially yield 100% prediction accuracy. Subsequently, I evaluated various combinations of characteristics, recording their prediction accuracies. Ultimately, I selected the combination of characteristics that yielded 100% predictive accuracy for inclusion in the user interface.

Furthermore, I delved into the attributes within these selected characteristics. Using Matplotlib, I visualized the quantity of specimens for each attribute, along with their classification as either non-toxic or toxic. Notably, certain attributes, such as 'buff' gill-color, exhibited high classification significance due to their association with toxicity.

With the characteristics and their attribute values inferred for inclusion in the application, I proceeded to develop the front end. This involved creating a simple Flask application with an HTML template featuring dropdown menus for the selected characteristics. Upon form submission, the selected attribute values were transferred to the main Python file for further processing. Within the main Python file, a function handled GET and POST requests, capturing the form data and processing it to generate predictions using a random forest classifier. Error handling was incorporated to address submissions that did not match any specimens within the data, ensuring informative feedback for users.

Hypothesis Verification

Initially, my hypothesis posited the existence of a concise combination of characteristics capable of classifying mushrooms with 100% accuracy. While this hypothesis was confirmed, I had initially hoped to achieve this with a model trained using fewer characteristics. My findings demonstrated that the characteristic of odor alone could accurately classify all mushrooms with over 98% accuracy. However, to achieve full accuracy, this characteristic needed to be combined with five others. Although it remains possible that a combination of fewer characteristics could yield 100% accuracy, the five characteristics tested validated my hypothesis. Consequently, it is evident that pursuing a final version of the prototype is both feasible and worthwhile.

Effective Visualizations & Reporting

The visuals integrated into the interface serve the dual purpose of highlighting the significance of each attribute for classification and providing insights into the distribution of specimens across different attributes. Upon accessing the interface, users are presented with a bar graph illustrating the quantity of edible and poisonous mushrooms.

Subsequently, users encounter a heatmap displaying the significance of all characteristics, except for veil-type, for classification purposes. This heatmap offers a comprehensive overview of each characteristic's relevance in classifying mushrooms.

The subsequent visual representation takes the form of a violin plot, which offers a detailed breakdown of each key characteristic and its attributes by class. This visualization is particularly insightful as it elucidates the attributes that are especially indicative of class distinction, both for poisonous and edible specimens.

Following this, users are presented with visuals that dissect each characteristic by attribute, demonstrating the prevalence of each attribute across different classes. These visuals provide valuable insights into attribute distribution and aid in the selection of characteristic combinations for testing against the random forest classifier.

Overall, these visualization methods play a crucial role in guiding the selection of characteristic combinations for testing and facilitate the removal of extraneous data that could potentially compromise classification accuracy or represent attribute options lacking matching specimens.

Accuracy Analysis

This prototype achieved a remarkable milestone by accurately classifying mushrooms with 100% precision. This exceptional accuracy is attributed to the careful selection of characteristics, which consistently led to unambiguous classifications of either edible or poisonous mushrooms, with no instances of misclassification observed.

To provide a visual representation of the classification performance, a confusion matrix was constructed, demonstrating the flawless classification achieved by the selected characteristics. Notably, every mushroom was accurately classified without any discrepancies.

Moreover, to validate the accuracy further, the scikit-learn classification report was utilized. This comprehensive report reaffirmed the flawless performance of the model, highlighting that not a single mushroom was misclassified. The report serves as a robust verification of the prototype's accuracy, underscoring its reliability in accurately distinguishing between edible and poisonous mushrooms.

Application Testing

Unit testing was meticulously conducted throughout the process, focusing on adjusting the combination of characteristics tested by the algorithm. Aligned with my initial hypothesis, the objective was to ensure the algorithm's near 100% accuracy with the fewest number of characteristics possible. This involved an exhaustive exploration of various combinations to determine the optimal features for inclusion.

Subsequently, usability testing was undertaken to evaluate the application's handling of submissions that did not align with any specimens in the dataset. Instances of null results encountered during this testing phase triggered internal server errors. To address this, an exception handler was implemented to present informative messages to users, ensuring a smooth user experience.

These tests were iterated upon post-deployment to verify the application's seamless functionality and alignment with intended objectives. This rigorous testing regimen underscored the commitment to delivering a robust and reliable application.

Application Files

To run the fungi-identification program simply follow these steps:

Ensure that Python is installed on your system. If not, you can download and install it from the official Python website. Make sure to select Python 3.x for compatibility with the program.

Once Python is installed, open a terminal or command prompt and install the required libraries using pip, Python's package manager. Typically, the IDE you are using will recommend the package to install.

Simply copy and paste the following command: `pip install pandas scikit-learn Flask seaborn` Next, download the Python program from its source, which might be hosted on platforms like GitHub. After downloading, navigate to the directory where the program is located using the terminal or command prompt. You can use the `cd` command followed by the path to the program's directory. With the terminal or command prompt in the program's directory, you can run the Flask application by executing the `app.py` file. To do so, type: `python app.py` Once the Flask app is running, it will start the development server, and you'll see output indicating that the server is running, along with the localhost URL and port number where the app is hosted.

To interact with the Flask application, open a web browser and navigate to the specified localhost URL and port number. Usually, this URL is `http://127.0.0.1:5000/` or `http://localhost:5000/`, where `5000` is the default port number used by Flask.

Upon accessing the web interface in your browser, you'll see the program to utilize.

Users can access the descriptive methods by scrolling down to the section where characteristics and their various attributes are visualized in relation to class. Following a prediction, users can easily return to the previous screen to initiate the algorithm with another combination of attributes. This seamless navigation ensures a user-friendly experience, allowing users to explore different attribute combinations and predictions with ease.

User Guide

Simply navigate to the web hosted site and enter the attributes you observe in the specimen (or imagine if working from a simulation). Click the ‘Classification’ button to get an estimation of toxicity. Below is an example attribute combination:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gill-Color** | **Population** | **Spore-Print-Color** | **Bruises** | **Stalk-Root** |
| black | abundant | black | false | equal |
| purple | solitary | black | true | bulbous |
| yellow | clustered | yellow | false | missing |
| chocolate | abundant | black | false | equal |
| pink | several | black | true | bulbous |
| white | scattered | brown | true | club |

Learning Experience

Embarking on this capstone project provided me with an invaluable opportunity to further immerse myself in the realm of machine learning, a field I am deeply passionate about. While I had some prior experience with Python, primarily in introductory programming tasks, delving into the intricacies of data analysis within Python was a significant step forward for me. Throughout the project, I encountered various technologies and tools essential for machine learning endeavors, each offering its unique set of challenges and learning curves.

Despite facing unfamiliar territory, I approached these challenges with determination and a willingness to learn. Consulting documentation for the different technologies became a routine part of my workflow, helping me navigate through complex concepts and functionalities. Additionally, I found myself turning to online resources such as Stack Overflow to troubleshoot common issues and refine my understanding of Python's capabilities in the context of machine learning.

As I navigated through the project, I realized the importance of breaking down complex problems into manageable sub-tasks. This approach not only helped me tackle daunting challenges effectively but also reinforced my confidence in addressing large-scale projects in the future. Moreover, the experience highlighted the vast potential of programming, particularly in the context of machine learning, where Python serves as a powerful tool for exploring and solving a myriad of real-world problems.

In essence, this capstone project served as a transformative journey, deepening my understanding of Python's role in machine learning and solidifying my passion for the field. It underscored the significance of continuous learning and problem-solving in overcoming obstacles and achieving success in complex endeavors.

Sources Cited

Mushroom. (1987). UCI Machine Learning Repository. https://doi.org/10.24432/C5959T.