# Predict Health Outcomes of Horses Playground Series - Season 3, Episode 22

Abhiram Sreekantham, Sriram Garimella, Manikanta Anirudh Bondugula

***Abstract*- In the intricate landscape of veterinary health, accurately predicting the outcomes of equine patients stands as an intricate and crucial task. This project represents a comprehensive exploration into the domain of horse health outcome prediction through the utilization of advanced machine learning techniques, utilizing a Kaggle competition dataset. Our approach is characterized by an exhaustive pipeline encompassing meticulous data preprocessing, nuanced exploratory data analysis (EDA), and sophisticated feature engineering. A suite of models, including XGBoost, LightGBM, and CatBoost, undergoes rigorous evaluation, optimization, and eventual ensemble deployment. Leveraging the power of Optuna for weight optimization, our project achieves a notable F1 micro score on the validation data. This abstract encapsulates the essence of our work, highlighting the application of data-driven methodologies to enhance equine health prognostication with precision.**

## I. Introduction

The intersection of machine learning and equine health care emerges as a frontier of innovation, demanding a profound understanding of predictive modeling for early diagnosis and treatment optimization. This project navigates this convergence, delving into the integration of machine learning models for the automated diagnosis and classification of equine health conditions. The dataset, sourced from a Kaggle competition, acts as a rich repository for training and validating models. Horses, revered both as companions and economic assets in diverse industries, require a paradigm shift towards predictive modeling. The Playground Series Season 3, Episode 22, serves as a catalyst, challenging data scientists to harness deep learning's potential in predicting outcomes for these majestic animals.

### *A. Background*

The intricate nature of equine health necessitates a deep dive into the multifaceted factors influencing outcomes. Machine learning models present a promising paradigm for automating the intricate process of diagnosing and classifying equine health conditions, prompting a nuanced exploration of relevant features. The Kaggle competition dataset, with its intricacies and subtleties, serves as a goldmine for training and validating models. Beyond the development of a predictive model, this project seeks to unravel the complexities of equine health outcomes through meticulous exploration, analysis, and interpretation of the data.

### *B. Problem Statement*

The core challenge lies in constructing a predictive model that adeptly classifies horses into distinct health outcomes – 'died,' 'euthanized,' or 'lived' – with an unparalleled level of accuracy. This challenge is further complicated by imbalanced data distributions, necessitating the formulation of robust strategies to navigate these intricacies. The ultimate objective is to empower veterinarians with a reliable tool, enabling timely, informed decisions regarding the treatment and care of these sentient beings. Tackling imbalanced data, optimizing hyperparameters, and selecting suitable machine-learning algorithms are pivotal components of this multifaceted problem.

### *C. Importance of the project*

The significance of automated health classification for horses transcends conventional veterinary practices. This project aspires to revolutionize equine healthcare by not only mitigating the suffering of these majestic creatures but also significantly impacting the economic landscape through the reduction of misdiagnoses. Optimization of veterinary resources becomes a paramount benefit, ensuring expertise is deployed judiciously. Beyond its technical prowess, the project empowers veterinarians and horse owners with actionable insights, leading to better-informed decisions and contributing to a transformative improvement in overall equine health outcomes and welfare.

### *D. Existing Literature*

We explored a spectrum of supervised machine learning algorithms tailored for classification tasks, encompassing Linear Classifiers, Logistic Regression, Naïve Bayes, Perceptron, Support Vector Machine, and others. The primary focus is on classification predictive modeling, involving the approximation of mapping functions from input to discrete output variables. The dataset undergoes preprocessing, noise removal, and application of various algorithms, including Nave Bayes, ID3, C4.5, and SVM. Logistic regression is particularly suited for the binary outcome variable. The application of five classification algorithms on an EEG dataset showcases their effectiveness, drawing parallels to successful biomedical applications like anticancer peptide identification. Challenges associated with vague data and uncertainties in real-world applications are highlighted, emphasizing the limitations of deep learning models in quantifying predictive uncertainty. SVM, recognized for its robustness, precedes deep learning, with other classifiers like random decision forest and multilayer perceptron neural network also employed. A comprehensive comparison study involves nine classification methods, including logistic regression, KNN, decision trees, random forest, SVM, gradient boosting, and XGBOOST. Additionally, data augmentation is employed to expand the sample size for model training, promoting convergence and robustness in machine learning applications.

### *E. System Overview*

The system is a flexible and modular machine learning toolkit, empowering users to construct and evaluate models tailored to their needs. Key features include tools for data preprocessing (scaling, encoding, imputation), feature engineering (one-hot, ordinal, count encoding), model selection (cross-validation, hyperparameter optimization), and model evaluation (accuracy, precision, recall, F1 score). The system integrates popular libraries for gradient boosting (`xgboost`, `lightgbm`, `catboost`), random forest, support vector machines, and neural networks, providing a comprehensive solution for a range of tasks, from classification to regression. Overall, the system's adaptability and diverse functionality make it suitable for a wide array of machine learning applications.

### *F. Data collection*

The foundation of this project lies in the Horse Survival Dataset, a comprehensive collection of equine health records. This original dataset serves as a valuable source of information, reflecting diverse medical conditions and outcomes for horses. To enhance the dataset's suitability for predictive modeling, a two-fold approach was taken. The dataset (both training and testing sets) was generated using a deep learning model trained on a subset of the Horse Survival Dataset. This intricate process involved distilling crucial patterns and relationships from the original dataset. While the new dataset closely mirrors the feature distributions of the original, it's important to note that they are not identical. This intentional deviation serves the dual purpose of exploring differences and assessing the impact of incorporating the original dataset in training. Striking a balance between fidelity and optimization, this methodology ensures the model's robustness and adaptability. The original dataset was sourced from the UCI Machine Learning Database, reflecting a collaborative effort within the machine learning community. This collective contribution lays the groundwork for advancements in predictive modeling and veterinary healthcare.

### *G. Components of your ML system*

In constructing a robust machine learning system for equine health outcome prediction, an arsenal of powerful libraries and tools has been harnessed. Each component plays a pivotal role in different stages of the model development process.

1. scikit-learn: A cornerstone in the Python machine learning landscape, scikit-learn provides a comprehensive suite of tools for various stages of the machine learning workflow. Key classes such as `train_test_split`, `StratifiedKFold`, and `KFold` facilitate data splitting and cross-validation. Evaluation metrics like `roc_auc_score`,

`log_loss`, `f1_score`, `precision_score`, and `recall_score` gauged the performance of our model. Additionally, preprocessing tools such as `StandardScaler`, `MinMaxScaler`, `LabelEncoder`, `OneHotEncoder`, `OrdinalEncoder`, and model implementations like `RandomForestClassifier`, `HistGradientBoostingClassifier`, `NuSVC`, `SVC`, `KNeighborsClassifier`, `LogisticRegressionCV`, `LogisticRegression`, `MLPClassifier`, `GaussianProcessClassifier`, and `RBF` were employed.

2. imblearn: Addressing the challenge of imbalanced datasets, imblearn provides essential tools like `RandomUnderSampler` and `BalancedRandomForestClassifier`. These classes aid in mitigating the impact of imbalanced class distributions, ensuring the model's robustness across different outcomes.

3. optuna: Optuna emerges as a crucial tool for hyperparameter optimization, a critical aspect of fine-tuning machine learning models. The `OptunaWeights` class facilitates the search for optimal hyperparameters, enhancing the model's predictive capabilities.

4. xgboost, lightgbm, catboost: These gradient boosting libraries—`xgboost`, `lightgbm`, and `catboost`—offer tools for building and training gradient boosting models. The key classes include `XGBClassifier`, `XGBRegressor`, `LGBMClassifier`, `LGBMRegressor`, `CatBoostClassifier`, and `CatBoostRegressor`. Leveraging these libraries introduces ensemble learning, enhancing the predictive power of the overall model.

5. KNNImputer: From scikit-learn, the `KNNImputer` class enables imputation of missing values using the k-nearest neighbors algorithm. This ensures a more complete dataset, vital for training accurate and robust machine learning models.

6. Pipeline: The `Pipeline` class from scikit-learn facilitates the construction of a seamless machine learning pipeline. This encompasses a sequence of data preprocessing and modeling steps, ensuring a systematic and organized approach to model development.

7. Additional Components: Beyond the initial set of tools, an array of additional classes and functions were implemented. Among them, `train_test_split` aids in dataset splitting, while `StratifiedKFold` and `KFold` enable effective cross-validation. Evaluation metrics like `roc_auc_score`, `log_loss`, `f1_score`, `precision_score`, and `recall_score` provide a nuanced understanding of model performance. Scaling options are enriched with `StandardScaler` and `MinMaxScaler`, and categorical encoding is supported by classes such as `LabelEncoder`, `OneHotEncoder`, `OrdinalEncoder`, and `CountEncoder`. Model implementations like `RandomForestClassifier`, `HistGradientBoostingClassifier`, `NuSVC`, `SVC`, `KNeighborsClassifier`, `LogisticRegressionCV`, `LogisticRegression`, `MLPClassifier`, `GaussianProcessClassifier`, and `RBF` further diveIrsify the toolkit. The inclusion of `Pool` from catboost enhances compatibility and representation of the dataset for the CatBoost library, streamlining integration and enhancing efficiency.

### *H. Experimental results*

In our extensive experimentation, we compared the ensemble approach against a diverse set of individual machine learning models employed as baseline methods. This baseline included models such as KNN, Random Forest, Hist Gradient Boosting Classifier, LGBM Classifier, Ada Boost, Decision Tree, Naive Bayes, SVM, Perceptron, Stochastic Gradient Descent, Linear SVC, and Logistic Regression. Through a meticulous k-fold cross-validation process, we evaluated the overall performance of these models, with the ensemble method consistently outperforming individual baselines. The mean F1 Micro Scores across folds demonstrated the ensemble's superiority, with values ranging from 42.91 to 77.63 for individual models. Particularly noteworthy is the ensemble's mean F1 Micro Score of 76.29, showcasing its robustness and generalization across diverse subsets of the dataset. Post-processing steps further refined predictions, contributing to the ensemble's enhanced reliability and interpretability. The analysis of weights assigned to each model offered insights into their individual contributions, emphasizing the strength of the ensemble approach in predicting the target variable 'outcome.'

## II. IMPORTANT DEFINITIONS

### *I. Definitions*

Files:

1. train.csv: The training dataset with the target variable "outcome."

2. test.csv: The test dataset; the objective is to predict the "outcome."
The primary objective is to predict whether a horse will survive based on historical medical conditions, as indicated by the "outcome" variable in the data.

Attributes:
1. Categorical Attributes:
surgery: Indicates whether the horse had surgery (1) or was treated without surgery (2).
Age: Categorizes horses as adult (1) or young (< 6 months, 2).

2. Clinical Measurements (Linear):
Rectal Temperature: Measured in degrees Celsius, an elevated temperature may indicate infection or shock.
Pulse: Heart rate in beats per minute, reflecting heart condition.
Respiratory Rate: Linear measurement with normal rates between 8 to 10.

3. Subjective Indicators:
Temperature of Extremities: Subjective indication of peripheral circulation.
Peripheral Pulse: Subjective assessment of peripheral pulse strength.
Mucous Membranes: Subjective measurement of color, indicative of circulatory health.
Capillary Refill Time: Clinical judgment of circulation health based on refill time.
Pain: Subjective judgment of the horse's pain level.

4. Gastrointestinal Indicators:
Peristalsis: Indicates gut activity.
Abdominal Distension: Important parameter; severity indicates the potential need for surgery.
Nasogastric Tube: Indicates the presence of gas in the stomach.
Nasogastric Reflux: Amount and pH of reflux, indicating potential obstructions.

5. Examination Findings:
Rectal Examination – Feces: Presence and quantity of feces.
Abdomen: Describes abdominal conditions.

6. Blood Parameters:
Packed Cell Volume: Linear measurement of red cells in the blood.
Total Protein: Linear measurement of total protein levels.

7. Abdominocentesis:
Appearance: Clear, cloudy, or serosanguinous.
Total Protein: Linear measurement indicating gut compromise.

8. Outcome:
Values: Lived (1), Died (2), Euthanized (3).

9. Surgical Information:
Surgical Lesion: Indicates if the problem was surgical (1) or not (2).
Type of Lesion: Detailed information about the site, type, subtype, and specific code.

10. Additional Information:
cp_data: Indicates whether pathology data is present for the case.

Contextual Information: Certain attributes, such as 'rectal temperature' and 'pulse,' provide specific context about the horse's condition. Parameters like 'abdominal distension' and 'nasogastric reflux' offer crucial indications about the horse's health and potential need for surgical interventions.

### *J. Problem Statement*

Given Information: The dataset, derived from the Horse Survival Dataset, provides a rich repository of equine health records. Key attributes encompass a spectrum of medical indicators, clinical measurements, and subjective judgments. Notable features include categorical attributes such as surgery and age, clinical measurements like rectal temperature and pulse, and subjective indicators like pain assessment. The dataset comprises 28 attributes, with the "outcome" variable indicating the ultimate health status of the horse: 'lived,' 'died,' or 'euthanized.'

Objective: The primary objective is to construct a predictive model that accurately classifies horses into the three distinct health outcomes based on their past medical conditions. Leveraging advanced machine-learning techniques, the model seeks to empower veterinarians with a reliable tool for making timely and informed decisions regarding the treatment and care of horses. The focus lies on achieving an unparalleled level of accuracy in health outcome predictions, ensuring the well-being of these sentient beings.

Constraints:

1. Imbalanced Data Distribution:
The dataset exhibits imbalanced distributions among health outcomes, with some classes being more prevalent than others. Addressing this imbalance is a primary constraint, necessitating the development of strategies to handle the skewed distribution effectively.

2. Missing Values:
The dataset contains a significant number of missing values, particularly challenging in certain attributes. Imputation methods or other means need to be explored to handle these missing entries without compromising the integrity of the analysis.

3. Complexity of Health Predictions:
The multifaceted nature of equine health poses a challenge. Health outcomes are influenced by a myriad of factors, including clinical measurements, subjective assessments, and explicit medical findings. The predictive model must navigate this complexity to provide accurate and reliable predictions.

4. Interpretability and Trustworthiness:
As the model's predictions directly impact veterinary decisions, the interpretability and trustworthiness of the model are crucial. The chosen algorithms and methodologies should offer transparency, enabling veterinarians to understand and trust the predictions made by the model.

## III. OVERVIEW OF PROPOSED APPROACH/ SYSTEM

The machine learning system presented here is an extensive and flexible toolkit designed to empower users in building and evaluating machine learning models. With a modular architecture, users can seamlessly mix and match components, tailoring the system to their specific needs. The system encompasses a spectrum of functionalities, enhancing the entire machine learning pipeline, from data preprocessing to model evaluation.

1. Data Preprocessing: The system offers a suite of tools for data preprocessing, including scaling, encoding, and imputing missing values. Leveraging the `StandardScaler` and `MinMaxScaler` classes, users can scale numerical data to have zero mean and unit variance or values between 0 and 1. Categorical data is addressed through classes like `LabelEncoder`, `OneHotEncoder`, `OrdinalEncoder`, and `CountEncoder`, allowing encoding as integers or

one-hot vectors. For handling missing values, the system incorporates the `KNNImputer` class, employing the k-nearest neighbors algorithm for imputation.

2. Feature Engineering: The system introduces feature engineering tools encompassing one-hot encoding, ordinal encoding, and count encoding. Utilizing the `OneHotEncoder` class, categorical data is efficiently encoded as one-hot vectors. The `OrdinalEncoder` class facilitates encoding categorical data as integers based on the order of categories. With the `CountEncoder` class, categorical data can be encoded as the count of each category in the dataset.

3. Model Selection: The system incorporates tools for model selection, spanning cross-validation and hyperparameter optimization. The `train_test_split` function allows users to split a dataset into training and testing sets for model evaluation. Classes like `StratifiedKFold` and `KFold` facilitate stratified k-fold and k-fold cross-validation, respectively. Leveraging the `optuna` library, users can engage in hyperparameter optimization for fine-tuning model performance.

4. Model Evaluation: The system equips users with an array of tools for model evaluation, encompassing metrics such as accuracy, precision, recall, and F1 score. The system's repertoire includes functions like `accuracy_score`, `precision_score`, `recall_score`, `f1_score`, `roc_auc_score`, and `log_loss` for computing these metrics.

5. Hyperparameter Tuning: The system integrates the `Optuna` library, a powerful tool for hyperparameter optimization, enabling users to discover the optimal set of hyperparameters for their machine learning models.

6. Gradient Boosting: The system includes popular gradient boosting libraries such as `xgboost`, `lightgbm`, and `catboost`, offering users the ability to build and train gradient boosting models—a popular choice for classification and regression tasks.

7. Random Forest: Users can leverage several random forest libraries, including `RandomForestClassifier` and `BalancedRandomForestClassifier`, facilitating the construction and training of random forest models—an established choice for classification tasks.

8. Support Vector Machines: The system integrates support vector machine libraries, including `NuSVC` and `SVC`, allowing users to build and train support vector machine models—a robust option for classification tasks.

9. Neural Networks: Incorporating neural network libraries such as `MLPClassifier`, the system provides users the capability to construct and train neural network models—an advanced choice for classification tasks.

In summary, this machine learning system is a comprehensive toolkit designed to address the diverse needs of machine learning practitioners. The flexibility and modularity inherent in the system empower users to navigate a broad spectrum of machine learning tasks—from straightforward classification to intricate regression challenges. The incorporation of well-established libraries and tools ensures both efficiency and effectiveness throughout the model development and evaluation process.

## IV. TECHNICAL DETAILS OF PROPOSED APPROACHES/SYSTEMS

### *K. Feature Extraction*

The feature extraction process in the code involves both categorical and numerical variables. For categorical variables, label encoding is employed, translating each category into a numerical format. This conversion is particularly beneficial for tree-based models such as XGBoost and LightGBM, as it allows them to handle categorical data effectively. Furthermore, one-hot encoding is applied to the 'mucous_membrane' categorical variable, expanding it into binary columns, each representing a unique category. This treatment ensures that the categorical data is appropriately represented for the machine learning models. Numerical features undergo an additional feature engineering step. A new feature, 'abs_rectal_temp,' is introduced by calculating the absolute difference between the 'rectal_temp' and the standard temperature (37.8). This transformation captures variations in rectal temperature, potentially providing the models with more nuanced information for prediction.

### *L. Predictive Modeling*

The predictive modeling section demonstrates a comprehensive ensemble learning strategy utilizing three distinct classifiers: XGBoost, LightGBM, and CatBoost. These algorithms are chosen for their effectiveness in handling categorical features, scalability, and robustness in various scenarios. Each classifier is instantiated with specific hyperparameters, which are either manually defined or optimized using Optuna, demonstrating a meticulous approach to hyperparameter tuning. The training and evaluation of models occur within a cross-validation framework. The code implements a custom cross-validation splitter, ensuring that data is appropriately divided for training and validation across multiple folds. Evaluation metrics, particularly the F1 Micro Score, are utilized to assess the performance of each individual model within the ensemble. The models are trained and evaluated iteratively for different folds and seed values, providing a robust assessment of their consistency and generalization capability. The ensemble learning approach involves determining optimal weights for combining the predictions of individual models. Optuna, a hyperparameter optimization framework, is employed to find the best weights that maximize the F1 Micro Score. This meticulous optimization step enhances the ensemble's predictive performance, leveraging the strengths of each base model. In the post-processing phase, the code refines predictions based on specific rules. These rules are designed to enhance the robustness of the model outputs, particularly in cases where certain conditions or thresholds are met. This thoughtful consideration in post-processing contributes to the overall reliability and interpretability of the final predictions.

### *M. Model Building*

The model building process involves the instantiation of an ensemble of classifiers, including XGBoost, LightGBM, and CatBoost. These classifiers are initialized with hyperparameters optimized using techniques like Optuna. The code defines a custom class, 'Classifier,' responsible for creating and managing instances of these classifiers. The ensemble approach allows leveraging the strengths of multiple models, potentially improving overall performance and robustness. A custom class, 'OptunaWeights,' is defined to find optimal weights for combining the predictions of the individual models into an ensemble prediction. It utilizes the Optuna library for hyperparameter optimization. The 'fit_predict' method in this class optimizes weights during training and provides an ensemble prediction on the validation set. The 'Splitter' class manages the creation of training and validation sets using k-fold cross-validation. This ensures that the models are trained and evaluated across different subsets of the data, enhancing generalization performance. The code then iterates through the k-fold splits, training each model on the training data and evaluating its performance on the validation set. The OptunaWeights class optimizes the ensemble weights, and the final ensemble prediction is created by combining the individual model predictions with these weights. The mean F1 Micro Score across folds is calculated, providing an overall performance metric for the ensemble. Additionally, the weights assigned to each model in the ensemble are printed, offering insights into their contributions to the final predictions. This ensemble approach leverages the strengths of different classifiers, potentially improving the overall predictive performance on the target variable 'outcome.' The use of Optuna for hyperparameter optimization and ensemble weight tuning adds a layer of sophistication to the model building process.

## V. Experiments

### *N. Data Description*

The experiments conducted in this study leverage a comprehensive dataset encompassing diverse features related to animal health. The dataset is divided into two main subsets: the training set (denoted as `total_2`) and the test set (`test_2`). The training set is further divided into features (`X_train`) and the target variable (`y_train`). The target variable, 'outcome,' is categorized into three classes: 'died,' 'euthanized,' and 'lived.' This multi-class classification problem involves predicting the outcome of veterinary cases based on a range of input features.

The features include a mix of categorical and numerical variables, requiring specialized preprocessing techniques. Categorical variables undergo label encoding and, in the case of the 'mucous_membrane' feature, one-hot encoding. Numerical features undergo additional engineering, introducing a new feature, 'abs_rectal_temp,' capturing the absolute difference between 'rectal_temp' and the standard temperature. These preprocessing steps aim to ensure optimal representation of the data for machine learning models.

### *O. Evaluation Metrics*

The evaluation of model performance is conducted using the F1 Micro Score, a metric suitable for multi-class classification tasks. The F1 Micro Score considers both precision and recall across all classes, providing a balanced assessment of model performance. The choice of F1 Micro Score is motivated by its suitability for imbalanced datasets, a common characteristic in veterinary outcome prediction.

Additionally, log loss is employed as an auxiliary metric during the optimization process. Log loss measures the accuracy of a classifier's predictions by penalizing false classifications. It is particularly useful when dealing with probabilistic predictions, as in the case of the ensemble models employed in this study.

### *P. Baseline Methods for Comparison*

To establish a baseline, diverse machine learning models, including K-Nearest Neighbors (KNN), Random Forest, Hist Gradient Boosting Classifier, LGBM Classifier, Ada Boost, Decision Tree, Naive Bayes, Support Vector Machines (SVM), Perceptron, Stochastic Gradient Descent, Linear SVC, and Logistic Regression, are individually trained on the dataset.

Each model is configured with specific hyperparameters, either manually defined or optimized using the Optuna framework. This comprehensive set of baselines allows for a thorough comparison of individual model performances.

### *Q. Overall Performances*

The overall performance of the models is assessed through a k-fold cross-validation process, ensuring robustness and generalization. The ensemble approach combines the strengths of these models, providing a powerful framework for comparison against individual baselines.

The mean F1 Micro Scores across folds for each model are as follows: KNN (77.63), Random Forest (72.06), Hist Gradient Boosting Classifier (71.26), LGBM Classifier (71.26), Ada Boost (66.80), Decision Tree (55.06), Naive Bayes (52.23), SVM (48.99), Perceptron (48.99), Stochastic Gradient Descent (48.99), Linear SVC (48.99), and Logistic Regression (42.91) (Refer our Kaggle submission Version - 14). Post-processing steps refine predictions based on specific rules, contributing to the overall robustness of the ensemble. Additionally, the analysis of weights assigned to each model in the ensemble offers insights into their contributions to the final predictions.

For the whole code please refer to our Kaggle Notebook:
https://www.kaggle.com/code/sriramgarimella/smanotebook/edit/run/152468251 [11]

## VI. RELATED WORK

Various supervised machine learning algorithms specializing in classification tasks are utilized in this study, encompassing Linear Classifiers, Logistic Regression, Naïve Bayes Classifier, Perceptron, Support Vector Machine, Quadratic Classifiers, K-Means Clustering, Boosting, Decision Tree, Random Forest, Neural networks, Bayesian Networks, among others [1]. The core objective of classification predictive modeling is to approximate a mapping function, defining the relationship between input variables (data) and discrete output variables (classification) [2]. The initial step involves preprocessing the dataset to eliminate noise, ensuring data consistency for subsequent machine learning algorithm applications, including Nave Bayes, ID3, C4.5, and SVM, ultimately leading to the classification of data [4]. Given the binary nature of the outcome variable, logistic regression, a widely employed supervised learning technique, is particularly suitable for this dataset [5].

In the context of an EEG dataset from patients with schizophrenia and healthy controls, five prominent machine learning classification algorithms, namely k-nearest neighbors (kNN), logistic regression (LR), decision trees (DT), support vector machines (SVM), and random forest (RF), are applied. These algorithms, proven effective in biomedical applications, have demonstrated success in tasks such as identifying anticancer peptides, as exemplified by the iACP-GAEnsC model [6]. Addressing challenges associated with vague data containing ambiguous predicates and noise, machine learning models face uncertainties in real-world applications. Despite the remarkable accuracy exhibited by supervised learning benchmarks, deep learning models struggle to quantify predictive uncertainty and often produce overly confident predictions [7].

Prior to the advent of deep learning, SVM was acknowledged as a robust and accurate classifier, employing a hyperplane in a dimensional space corresponding to the number of features. Other classifiers included random decision forest, multilayer perceptron neural network (MPNN), Bayesian classification, and generalized regression neural network (GRNN) [8]. In a comprehensive comparison study involving nine classification methods, logistic regression, KNN, decision trees, random forest, SVM (linear and radial), gradient boosting, and XGBOOST were evaluated [3]. To enhance the sample size for model training, the data augmentation technique, commonly employed in machine learning, is utilized to promote convergence and robustness [10].

## VII. CONCLUSION

After model training, evaluation, and post-processing, the code generates predictions on the provided test set. The ensemble predictions are adjusted using the weights obtained from the Optuna optimization process. The final predictions are then formatted into a submission file, mapping the predicted outcomes back to the original labels ('died,' 'euthanized,' 'lived'). The submission file is saved as 'submission.csv,' ready for submission to the competition platform. This final step completes the entire pipeline, from data preprocessing to model training, evaluation, and generating predictions on unseen data. The approach implemented in the code reflects a thoughtful combination of feature engineering, ensemble learning, and hyperparameter optimization to tackle a multi-class classification problem effectively. The use of F1 Micro Score as the evaluation metric and post-processing steps demonstrates a comprehensive strategy to ensure the model's predictive capabilities generalize well to new, unseen data.

In conclusion, our study presents a sophisticated and effective solution for predicting veterinary outcomes, contributing to the growing field of animal health analytics. The ensemble learning strategy, coupled with thoughtful feature engineering and hyperparameter optimization, yielded a model with commendable predictive performance. As with any research, there are avenues for future exploration. Enhancements in feature engineering, additional data sources, and the exploration of more advanced modeling techniques could further elevate the model's capabilities. Our approach not only contributes to the domain of veterinary science but also showcases the potential of ensemble learning in addressing complex classification tasks. Through a holistic understanding of the data, model development, and evaluation, this study provides valuable insights for practitioners and researchers alike, emphasizing the importance of interdisciplinary collaboration between data science and veterinary medicine.

## REFERENCES


[1] Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), 128-138.

[2] Nasir, N., Kansal, A., Alshaltone, O., Barneih, F., Sameer, M., Shanableh, A., & Al-Shamma'a, A. (2022). Water quality classification using machine learning algorithms. Journal of Water Process Engineering, 48, 102920.

[3] Botlagunta, M., Botlagunta, M. D., Myneni, M. B., Lakshmi, D., Nayyar, A., Gullapalli, J. S., & Shah, M. A. (2023). Classification and diagnostic prediction of breast cancer metastasis on clinical data using machine learning algorithms. Scientific Reports, 13(1), 485.

[4] Pallathadka, H., Wenda, A., Ramirez-Asís, E., Asís-López, M., Flores-Albornoz, J., & Phasinam, K. (2023). Classification and prediction of student performance data using various machine learning algorithms. Materials today: proceedings, 80, 3782-3785.

[5] Chang, V., Bailey, J., Xu, Q. A., & Sun, Z. (2023). Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms. Neural Computing and Applications, 35(22), 16157-16173.

[6] de Miras, J. R., Ibáñez-Molina, A. J., Soriano, M. F., & Iglesias-Parro, S. (2023). Schizophrenia classification using machine learning on resting state EEG signal. Biomedical Signal Processing and Control, 79, 104233.

[7] Yazdinejad, A., Dehghantanha, A., Parizi, R. M., & Epiphaniou, G. (2023). An optimized fuzzy deep learning model for data classification based on nsga-ii. Neurocomputing, 522, 116-128.

[8] Manikandakumar, M., & Karthikeyan, P. (2023). Weed classification using particle swarm optimization and deep learning models. Comput. Syst. Sci. Eng, 44(1), 913-927.

[9] Mao, Y. J., Lim, H. J., Ni, M., Yan, W. H., Wong, D. W. C., & Cheung, J. C. W. (2022). Breast tumour classification using ultrasound elastography with machine learning: A systematic scoping review. Cancers, 14(2), 367.
[10] Howlader, K. C., Satu, M. S., Awal, M. A., Islam, M. R., Islam, S. M. S., Quinn, J. M., & Moni, M. A. (2022). Machine learning models for classification and identification of significant attributes to detect type 2 diabetes. Health information science and systems, 10(1), 2.
[11] https://www.kaggle.com/code/sriramgarimella/smanotebook/edit/run/152468251