Project Documentation: Beyond the Veil of Wellness

Title:

Beyond the Veil of Wellness: Machine Learning's Unique Journey in Animal Health Classification

Category:

Machine Learning

Skills Required:

- Machine Learning
- Data Analysis
- Python (Flask)
- HTML/CSS (Web Frontend)
- Model Deployment

Project Description:

“Harmonizing Health and Machine” is an AI-driven initiative that utilizes machine learning to assess the health of animals using behavioral, physiological, and environmental data. It allows for early detection of illnesses and objective diagnosis across farms, zoos, and homes.

Use Case Scenarios:

Scenario 1: Early Illness Detection in Farm Animals

ML model detects early signs of mastitis from sensor data in cows, improving animal welfare and reducing costs.

Scenario 2: Automated Health Monitoring in Zoos

Deep learning flags abnormal behavior in an elephant, indicating a joint issue, prompting a vet check-up.

Scenario 3: Pet Wellness App

An ML-enabled mobile app tracks pet data and alerts deviations, aiding in early diagnosis like diabetes or arthritis.

Project Flow:

1. User inputs values via web UI.
2. Model predicts health status based on input.
3. Prediction is displayed on the UI.

Milestone 1: Define Problem / Problem Understanding

Activity 1: Specify Problem Statement

Develop a machine learning model to classify animal health status as either *Critical* or *Normal* based on health indicators.

Activity 2: Business Requirements

- **Risk Assessment:** Analyze multiple factors to determine disease likelihood.
- **Financial Planning:** Budget for preventive and emergency veterinary care.
- **Early Warning System:** ML-based alerts for timely interventions.

Milestone 2: Data Collection and Preparation

Dataset Source

- Format: CSV
- Source: Kaggle

Attributes Description:

Feature	Description
AnimalName	Name of species
BloodBrainDisease	Diseases of blood/brain
AppearenceDisease	Physical appearance related diseases
GeneralDisease	Common diseases
LungDisease	Lungs related diseases
AbdominalDisease	Abdomen related diseases
HealthStatus	Target variable: Critical or Normal

Activity 1.1: Import Libraries

Import necessary libraries: `pandas`, `numpy`, `matplotlib`, `seaborn`, `sklearn`, etc.

Activity 1.2: Load Dataset

```
import pandas as pd
df = pd.read_csv("data.csv")
```

Activity 2.1: Preliminary Info

Use `df.shape`, `df.info()`, and `df.describe()` for initial inspection.

Activity 2.2: Handle Missing Values

- Check missing: `df.isnull().sum()`
- Replace nulls in `HealthStatus` with mode.

Activity 2.3: Drop Unwanted Columns

- None dropped, all features retained.

Milestone 3: Exploratory Data Analysis

Descriptive Statistics

Use `df.describe()` and `df.value_counts()` to understand distribution.

Visualization

- Bar plots and histograms for class balance
- Decision: Balance the dataset if skewed

Encode Categorical Data

Use label encoding for all columns since they are categorical.

Milestone 4: Model Building

Train-Test Split

```
from sklearn.model_selection import train_test_split
X = df.drop('HealthStatus', axis=1)
y = df['HealthStatus']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Models Trained:

- Logistic Regression
- K-Nearest Neighbors
- Decision Tree Classifier
- Random Forest Classifier

Each model:

```
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Model Evaluation:

Use `accuracy_score`, `confusion_matrix`, and visualization.

Best Model:

Random Forest (Accuracy: 95%)

Milestone 5: Model Deployment

Save Model

```
import pickle
pickle.dump(rfc, open("rfc.pkl", "wb"))
```

Load Model

```
model = pickle.load(open("rfc.pkl", "rb"))
```

Milestone 6: Web App Integration

Project Structure:

```
ML_Project/
|| app.py
|| rfc.pkl
|| data.csv
|| Project_Report.pdf
|| video_link.txt
|├── templates/
|     ├── index.html
|     └── result.html
|└── static/ (if needed)
```

Flask Backend (app.py)

- Use Flask to render templates and handle form input
- Predict and display result

HTML Pages

- **index.html**: Input form for all 6 features
- **result.html**: Display prediction (Critical or Normal)

GUI Description

Inputs Collected:

- AnimalName
- BloodBrainDisease
- AppearenceDisease
- GeneralDisease
- LungDisease
- AbdominalDisease

Output:

- Predicted Health Status (Normal / Critical)

Final Deliverables

1. Codebase with model and web app
2. Dataset (CSV)
3. Pickle file of trained model (rfc.pkl)
4. Project Report (`Project_Report.pdf`)
5. Video Demonstration Link (`video_link.txt` or in README)
6. README.md with instructions

Project Demonstration

Activity 1: Record End-to-End Explanation Video

Explain problem statement, data preparation, modeling, evaluation, and demo.

Activity 2: Upload to GitHub

Include all files: code, dataset, documentation, and video link.

Conclusion

This project successfully applies machine learning to classify animal health conditions using a well-structured pipeline: data processing, model training, evaluation, and deployment through a user-friendly web application.