Pawsitive Vibes: Final Report

UC Davis MSBA – Big Data Analytics
Team Fifteen: Kyle Ayisi, Jovoney Morton, Rohan Saxena, Jeremy Twumasi Ampofo, Jerry Xia

---

Relevant Project Materials

- Github Repository | Presentation Deck | Streamlit App

---

1. Business Objective

Pet adoption in the U.S. is riddled with inefficiencies. In 2024 alone, shelters took in over 5.8 million pets, with more than 334,000 euthanized. Many of these outcomes stem from mismatches with adopters selecting pets that don't align with their lifestyles. This leads to returns, frustrations, and prolonged shelter stays.

Our goal was to reimagine pet adoption as a data-driven process. *Pawsitive Vibes* uses machine learning to recommend compatible pet-adopter matches, increasing the likelihood of successful, long-term placements while reducing stress on shelter systems.

2. Key Actionable Initiative

To address this problem, our team built a recommendation engine and front-end app that simulates real-world adoption matching. This solution combines synthetic data generation, scalable processing, model training, and interactive deployment. Our most impactful initiative was delivering a functioning MVP with both predictive capability and interpretability.

Key components included:

- **Synthetic adopter profiles** generated using faker and SDV libraries
- **Real pet data** sourced from the Austin Animal Center (120K+ records)
- **Rule-based matching logic** to label training data
- **Wide & Deep neural network** trained to predict match quality
- **Streamlit app interface** with two views: AdopterView and ShelterView
- **Local LLM integration** for natural-language explanations of recommendations

3. Metrics of Success

We focused on both model performance and system usability as our success indicators.

Model evaluation:

- Test Accuracy: **96.86%**
- Test AUC: **0.9962**
- Loss: **0.0824**

System performance:

- Efficient data pairing for over 2 million examples
- Smooth user experience in the web app interface
- Real-time recommendation generation

Our working hypothesis was that an ML-driven matching system would:

- Increase compatibility-based match rates by **30%+** compared to baseline methods
- Improve adopter decision efficiency by **40% or more**

4. Role of Analytics

Analytics was integral to the design, execution, and testing of the solution.

- **Enablement:** Spark and pandas were used to generate, process, and store synthetic datasets.
- **Ideation:** Exploratory analysis shaped the design of match-scoring heuristics and feature engineering.
- **Evaluation:** Predictive metrics and user-facing LLM explanations helped assess match quality and transparency.

In short, analytics powered everything—from how we built data to how the end-user interacts with our product.

5. Thinking Through the Analytics

Data Design:
We combined real and synthetic data to simulate a functioning shelter system.

- Pet data (125,000+ profiles) included age, breed, color, and sex
- Adopter profiles (8,000 generated) included:
    - Age, housing type, household size
    - Activity level, pet preferences, prior ownership

Target variable: Match likelihood (binary: good fit = 1, poor fit = 0)
Features: A mix of categorical and numerical attributes for both adopter and pet

Analytics types used:

- **Exploratory:** Understand feature distributions and imbalances
- **Predictive:** Train Wide & Deep model on labeled pairs
- **Narrative (causal-inspired):** LLM-generated explanations simulate human-like reasoning for match logic

Challenges encountered:

- Missing features (e.g., pet weight) were neutralized in the scoring function
- GCP quota issues were resolved by using pandas with 'gcsfs' instead of Dataproc
- TensorFlow shape/dtype mismatches were resolved by explicitly formatting model inputs

6. Executing the Analytics

Each team member contributed to different parts of the pipeline:

- **Data creation and processing:** Kyle & Jovoney
- **Model training and tuning:** Jeremy
- **Streamlit deployment and UI:** Jerry & Rohan
- **Cross-team collaboration:** Metric definition, test plans, LLM tuning

Spark was used early for synthetic data generation and rule-based scoring. TensorFlow powered the predictive modeling. The app itself was built using Python and Streamlit, with llama-cpp-python powering the local LLM for match rationales.

7. Implementation Strategy

The app simulates a two-sided experience:

- **AdopterView:** Users enter attributes (age, housing, lifestyle) and receive ranked pet matches with scores and LLM explanations.

- **ShelterView:** Staff can input a pet's profile and view top adopter candidates, useful for prioritizing outreach or preparing for meet-and-greets.

Features were selected for simplicity and flexibility to make the interface usable with minimal technical background. Real-time scoring is fast, and explanations help increase trust in the AI system.

8. Scale and Future Plans

Scaling the project to production involves several challenges:

Barriers:

- Limited access to real adopter behavioral data

- Varying shelter digital readiness and openness to automation

- Underfunded operations with minimal tech staff

Solutions:

- Provide a plug-and-play app requiring no engineering to deploy

- Include LLM-powered explanations to reduce "black box" fears

- Plan for retraining and feedback loops as more user data becomes available

Our long-term vision is to continuously improve the system by retraining on real adopter behavior and incorporating feedback signals (e.g., thumbs up/down on matches). Pawsitive Vibes is built to scale ethically, responsibly, and empathetically.