Healthy Pets
ML Prior Auth Model

Elena Moseyko – Senior Data Scientist Presentation

Agenda

- 1. Problem Statement & Objective
- 2. Data Exploration & Insights
- 3. Modeling Approach
- 4. Evaluation & Results
- 5. Recommendations & Next Steps

Problem Statement

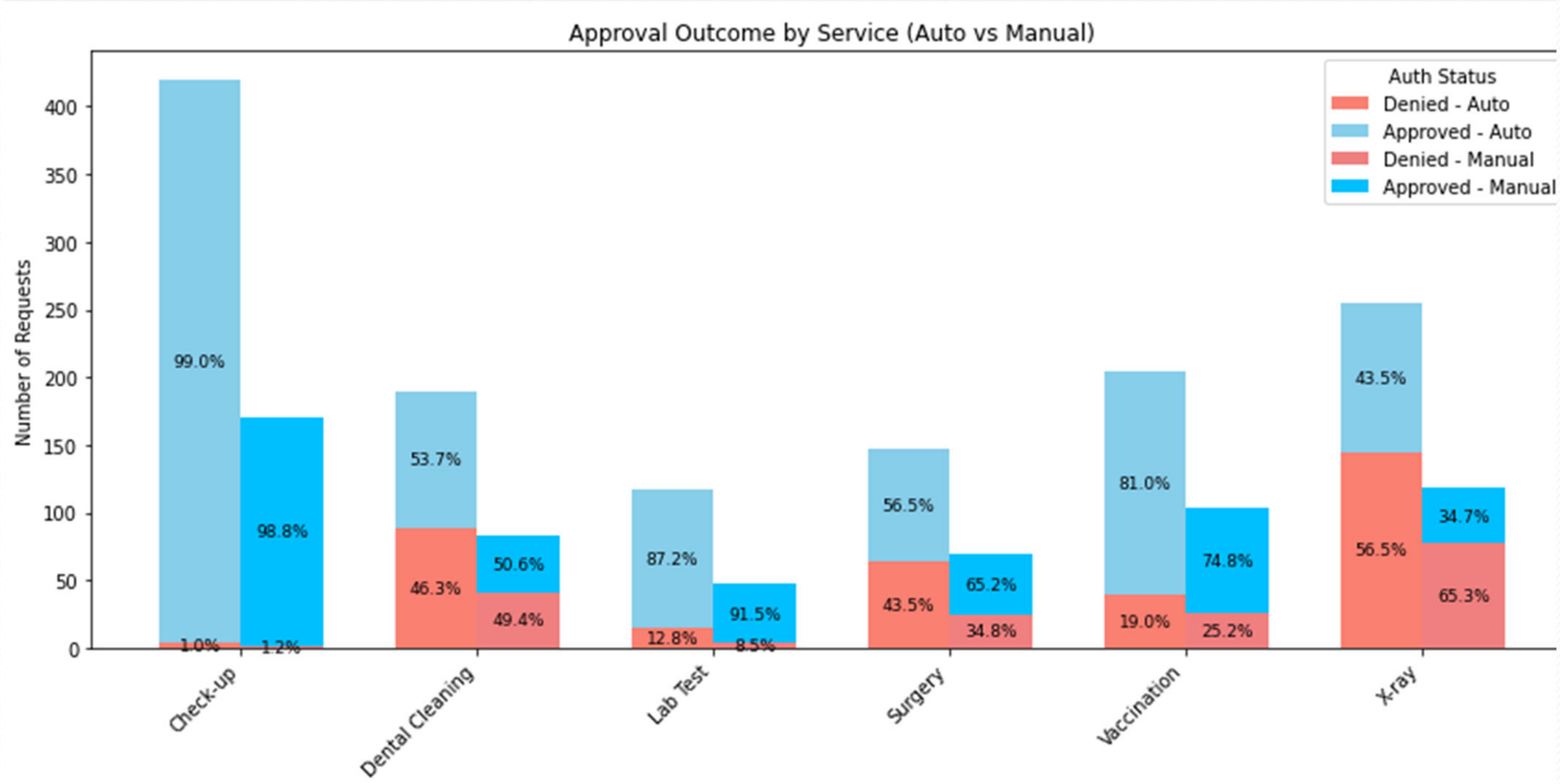
- Healthy Pets manages prior authorization requests between veterinarians and pet insurance providers
- These requests ensure procedures are clinically appropriate and covered by insurance
- Currently, some requests are automatically approved using rule-based logic
- Others require manual review, which can:
  - Delay necessary procedures
  - Increase operational burden
  - Create inconsistent outcomes
- The goal: reduce manual reviews by introducing a machine learning model to predict auto-approvals

Objective

- **Build a machine learning model that predicts whether a prior authorization request should be auto-approved**
- **Apply the model at the time of submission, before manual review begins**
- **Ensure the model:**
  - **Complements existing rule-based logic**
  - **Reduces manual workload for clinical reviewers**
  - **Maintains clinical appropriateness and safety**

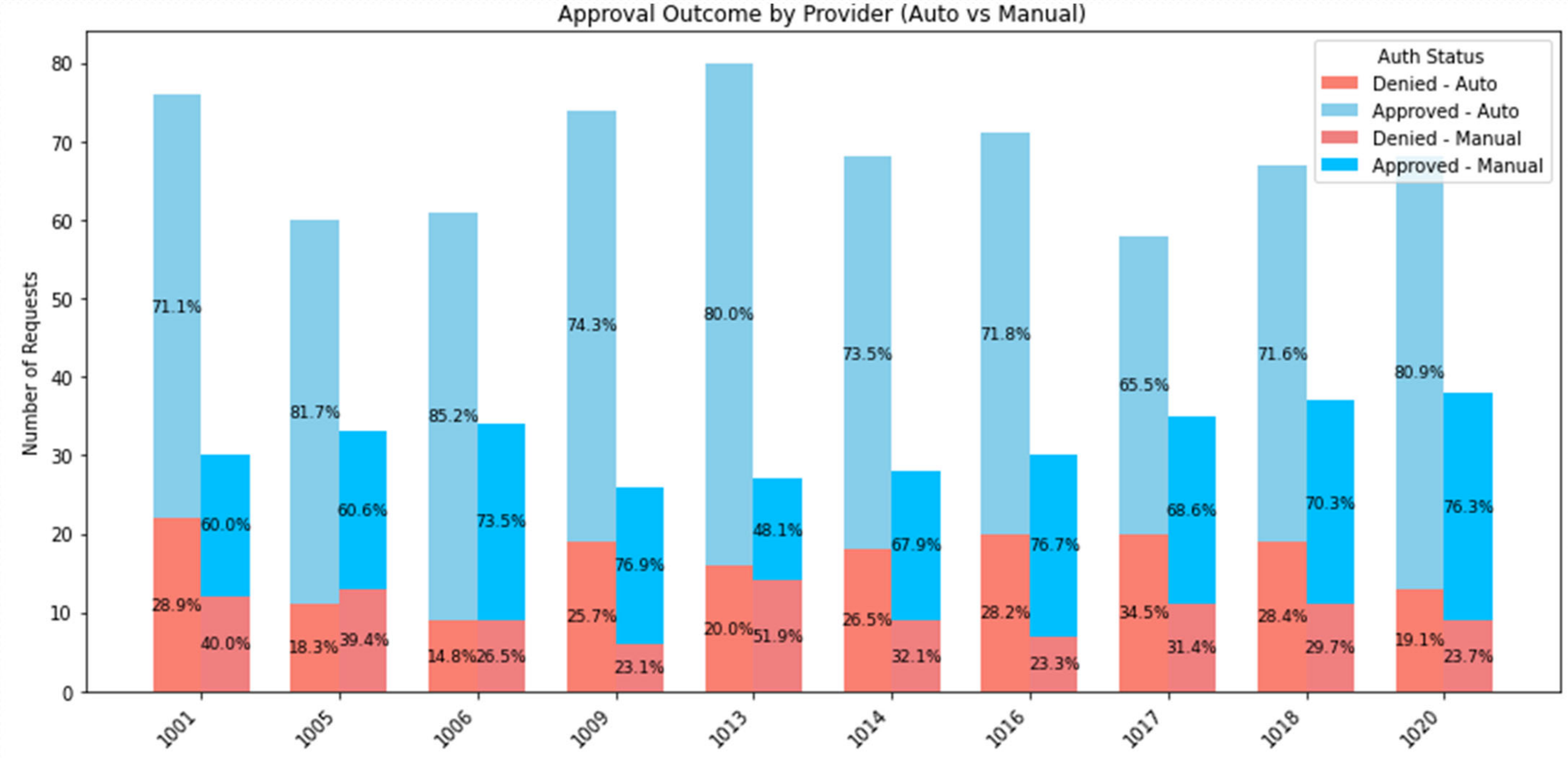
Data Exploration Insights

- **72.5% of prior auths are approved: 1,396 out of 1,924 requests**
- **Some procedures (e.g., vaccinations, check-ups) are more likely to be approved than others (e.g., surgeries)**



Data Exploration Insights

- **Certain providers have higher approval rates**



Data Exploration Insights

- **Pets with prior claims, especially for the same provider or service, are more likely to receive approval**
- **Time-based features matter:**
  - **Shorter time since last claim or auth correlates with higher approval rates**
  - **“Cold cases” (long time gaps) are less likely to be approved**
- **Requests without prior claim or auth history are more likely to be denied**

Modeling Approach

- Evaluated 3 classification models: Logistic Regression, Random Forest, XGBoost
- Used pipeline architecture for preprocessing: encoding, scaling, and missing value handling
- Engineered features from PriorAuth and Claims data:
  - Prior approvals, claim patterns, provider/service history, timing signals
- Used F1 score for model selection to balance precision and recall
- Random Forest was chosen as the final model for its strong performance and interpretability

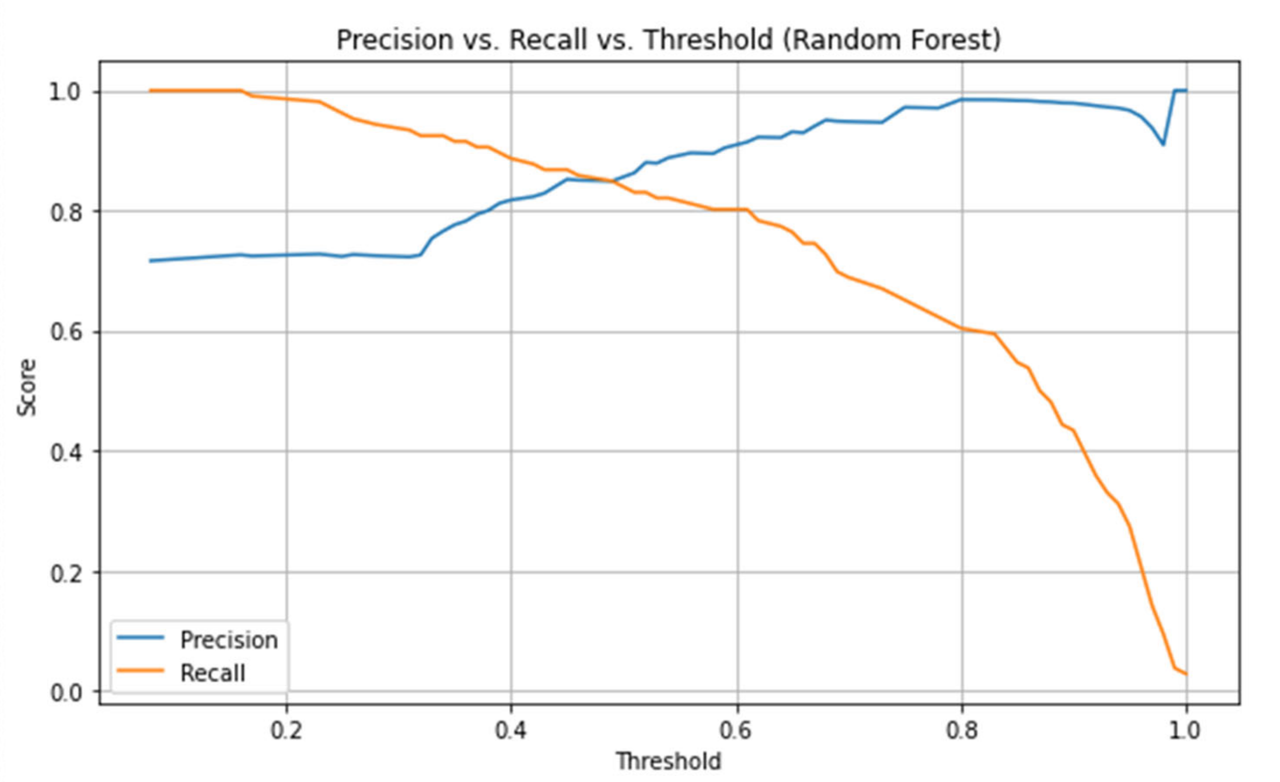
Model	Accuracy	F1 Score	Precision	Recall
Random Forest	0.871622	0.872914	0.875002	0.871622
XGBoost	0.851351	0.851351	0.851351	0.851351
Logistic Regression	0.763514	0.764344	0.765248	0.763514

Evaluation & Results

- **We evaluated all models using multiple metrics. Random Forest outperformed other models:**
  - **F1 Score: 0.873**
  - **Precision: 0.875**
  - **Recall: 0.872**
- **Feature importance was also reviewed for interpretability**
  → Top predictors included prior claim amounts, recent activity, and service type

Evaluation & Results

- **Classification models output probabilities, not just yes/no decisions.**
- **We tested thresholds from 0.1 to 0.95 to optimize performance**
- **Threshold = 0.6 gave the best F1 score**
- **Threshold = 0.9 simulates a conservative, production-safe auto-approval setting**



Evaluation & Results

We evaluated two thresholds to balance automation and confidence:

- **Threshold = 0.47 (Balanced)**
  - F1 Score: 0.87 strong balance between precision and recall
  - Precision: 88% most auto-approvals were correct
  - Recall: 86% captured the majority of true approvals
  - Accuracy: 82% good overall performance
  - Auto-Approval Rate: ~74% high operational efficiency
- **Threshold = 0.90 (Conservative)**
  - F1 Score: 0.58. expected trade-off in conservative setup
  - Precision: 98% very high confidence in approvals
  - Recall: 42% more selective, fewer approvals
  - Accuracy: 57% reflects narrower decision scope
  - Auto-Approval Rate: ~29% suitable for cautious deployment

Recommendations & Next Steps

Recommendations

- Start with a conservative threshold (0.90) for initial deployment
- Monitor model performance continuously
- Track precision, recall, and approval rate in real-time
- Set up alerts for unusual behavior or drift
- Use human review for all non-auto-approved cases

Next Steps

- Deploy in a pilot phase with a limited set of providers or services
- Evaluate real-world performance against manual review outcomes
- Retrain model periodically with new data and feedback
- Reassess threshold strategy after early success - consider scaling to 0.47 for broader automation