Cat Adoption Rate in Animal Shelter

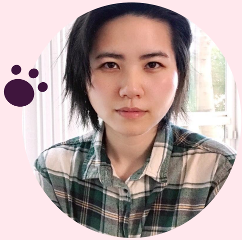
Our Team

Gracy Lu

Bingbing Ma

Minh Tran

Amy Huang



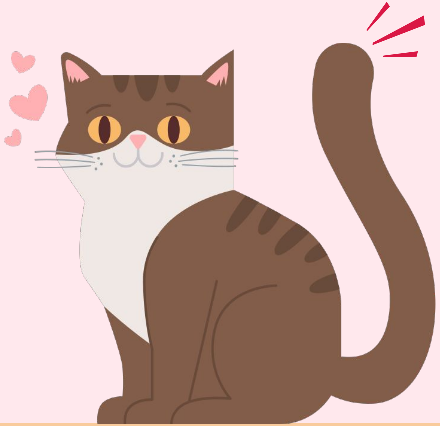
Li Wan

Pok Chi Yang

Table of Contents

01

Project Overview



What Kind Of Cats Are Most Likely To Be Adopted?

We will be discovering dataset from Long Beach Animal Shelter.

Stakeholders:

- Animal shelter staff. Aimed to discover factors that may make certain cats more likely to be adopted in an animal shelter.

Project Motivation:

- Decrease euthanasia rate in animal shelter, help shelter understand what feature impact people's adoption decision for improvement.

Dataset

1. Data Source

❖ Long Beach animal shelter dataset

2. Data Format

❖ CSV file

3. Data Contents

❖ Time Frame: January 1, 2017 - Present

❖ Category: Cat, Dog, Bird, Reptile, Other

❖ Publish Frequency: Daily

❖ Cat has the largest population

02

Exploratory Data Analysis

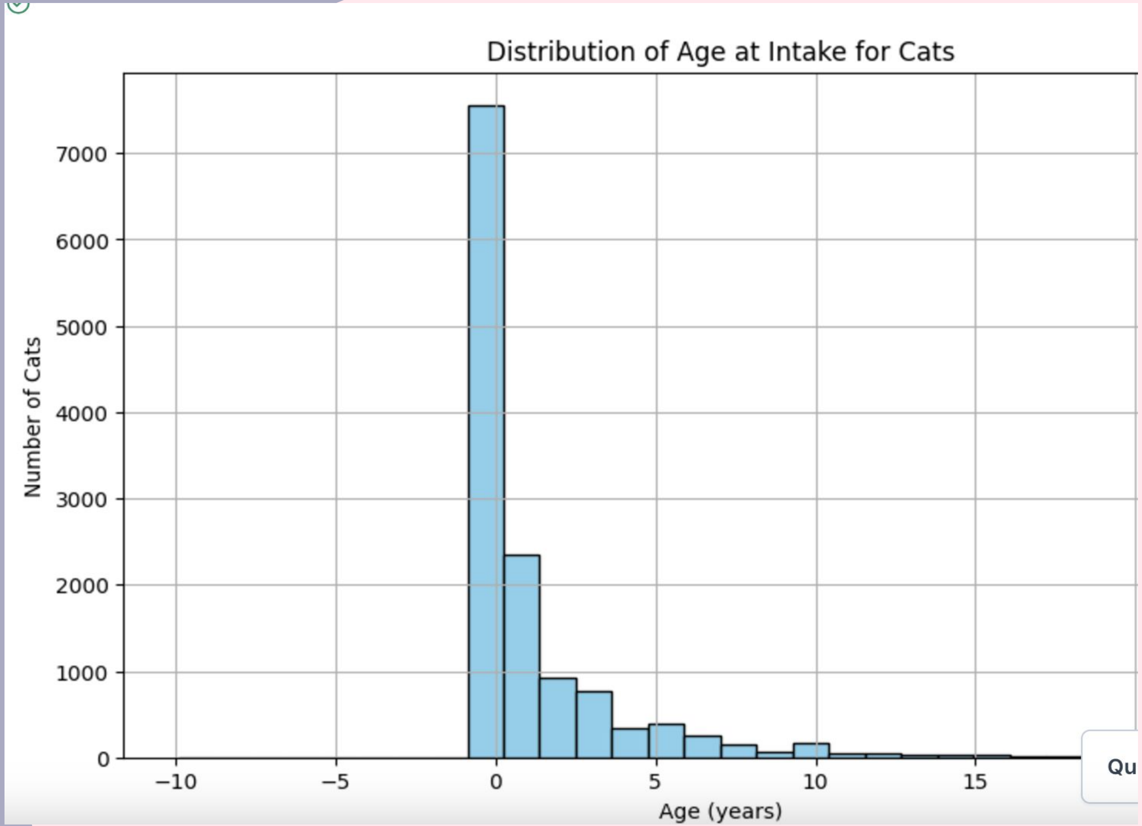
Data Bias Underlying Our Dataset

What is Data Bias?

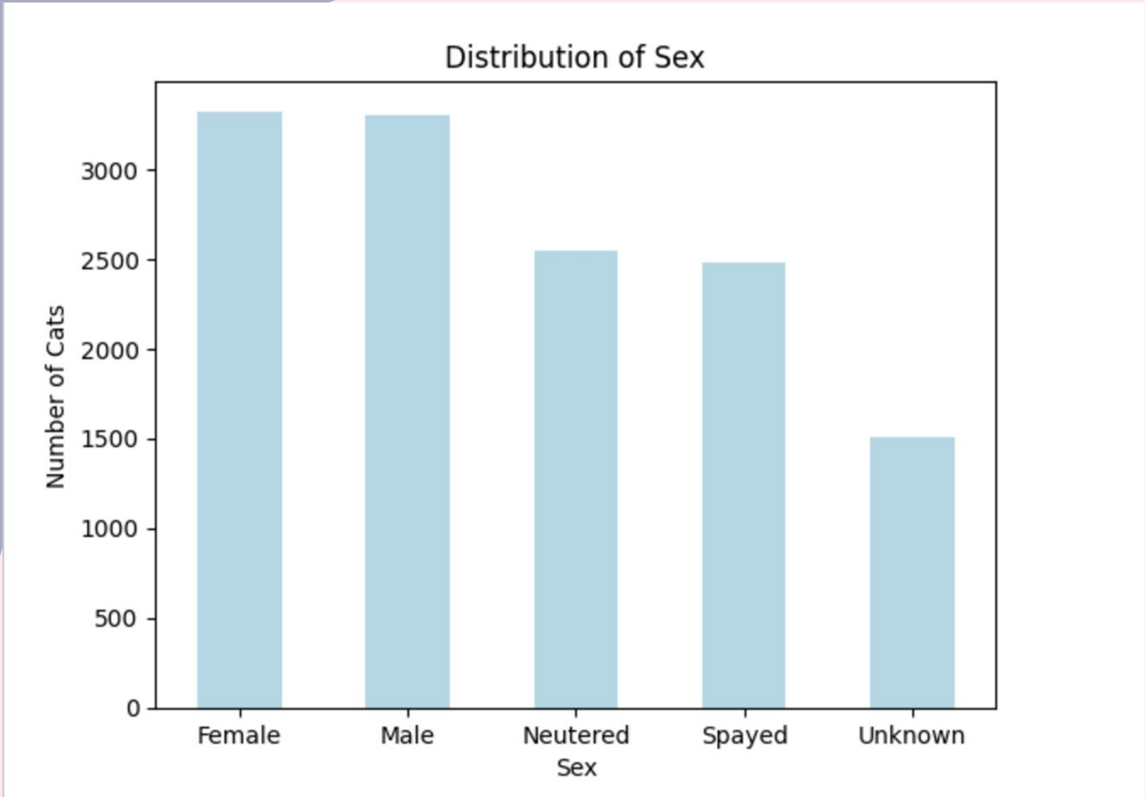
- Data bias occurs when certain elements in data are overrepresented or underrepresented, which can affect the outcomes of our analysis or predictions.

About Our Dataset

- We are looking at records from an animal shelter, focusing on the age and sex of cats to identify potential biases.



- Most cats in our data are under 5 years old.
- Fewer older cats in our data could mean our analysis might not work as well for them.



- No bias in distribution of sex

Real-World Implications of Data Bias

- **Bias Shows Real Issues:**
  - Bias helps us see patterns, like which age groups of cats are most often in shelters.
  - Understanding these patterns helps shelters improve how they care for cats.
- **Understanding Bias:**
  - By knowing about bias, we can better address the needs of all cats, not just the ones most commonly seen in shelters.

Reasons Cats are Surrendered:

- **Young Cats:** Many young cats end up in shelters due to over breeding in spring and summer, when stray cats give birth to kittens. Owners may also surrender unplanned litters if they lack the resources or space to care for them.
- **Old Cats:** Older cats are often brought to shelters due to health issues or because their owners can no longer take care of them.

Future Research Directions:

- Explore deeper into why different ages of cats are given up to shelters.
- Develop plans and support systems to help keep cats with their families and prevent unnecessary surrenders.

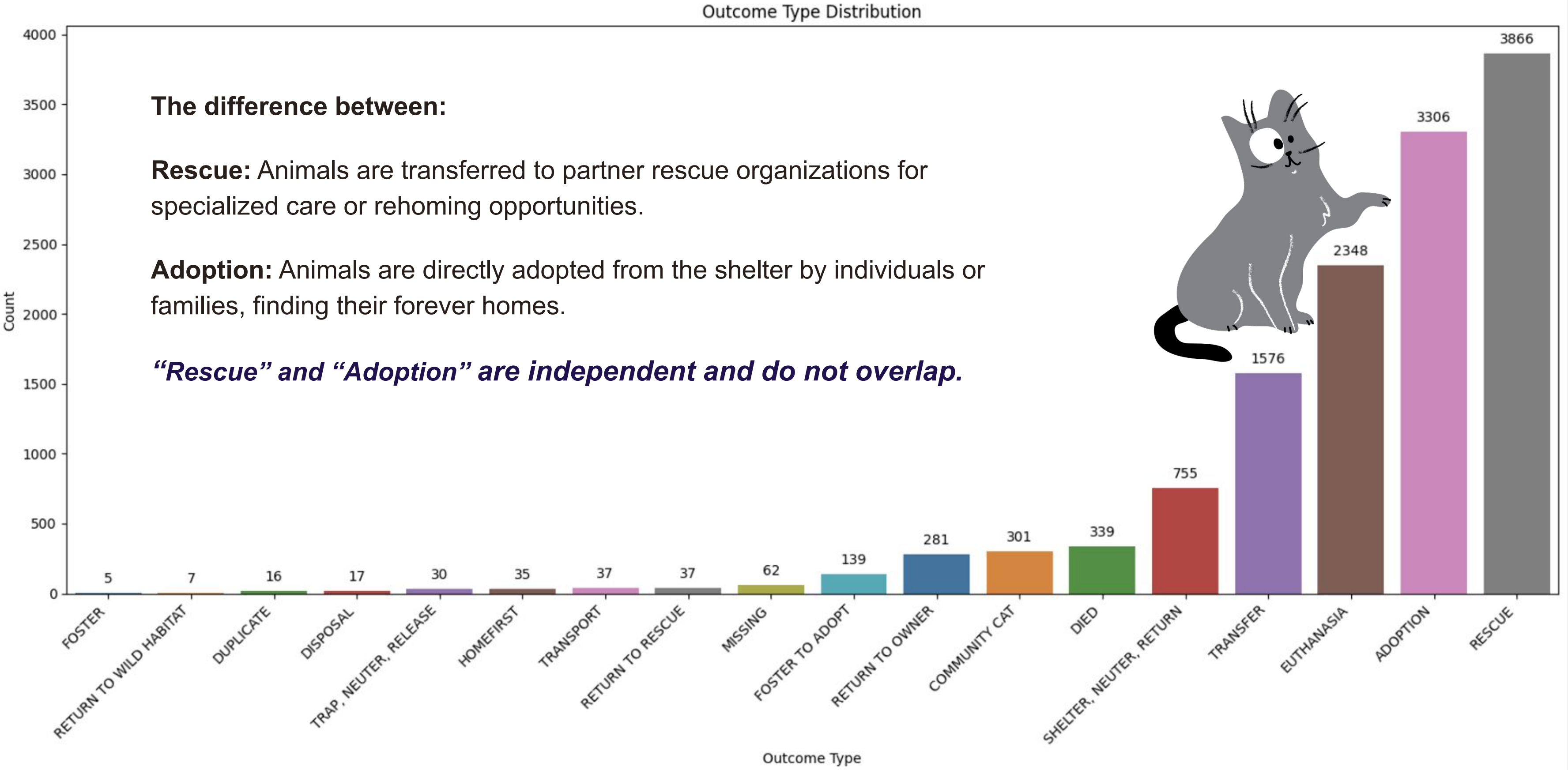
Data source:
https://vethelpdirect.com/vetblog/2024/10/09/why-are-cats-being-relinquished-to-shelters-and-what-can-be-done-about-it/
https://www.mdpi.com/2076-2615/14/17/2606

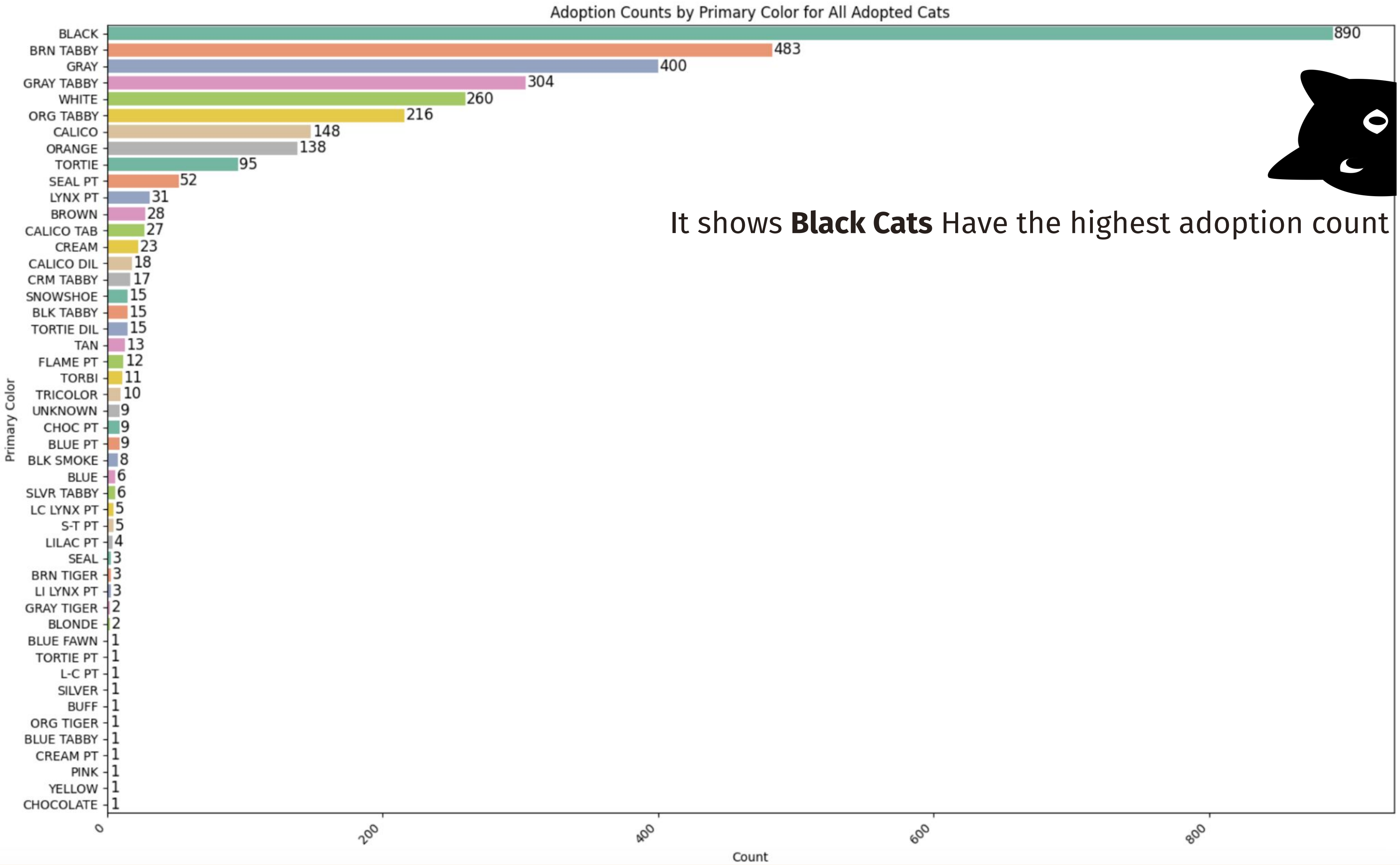
The difference between:

Rescue: Animals are transferred to partner rescue organizations for specialized care or rehoming opportunities.

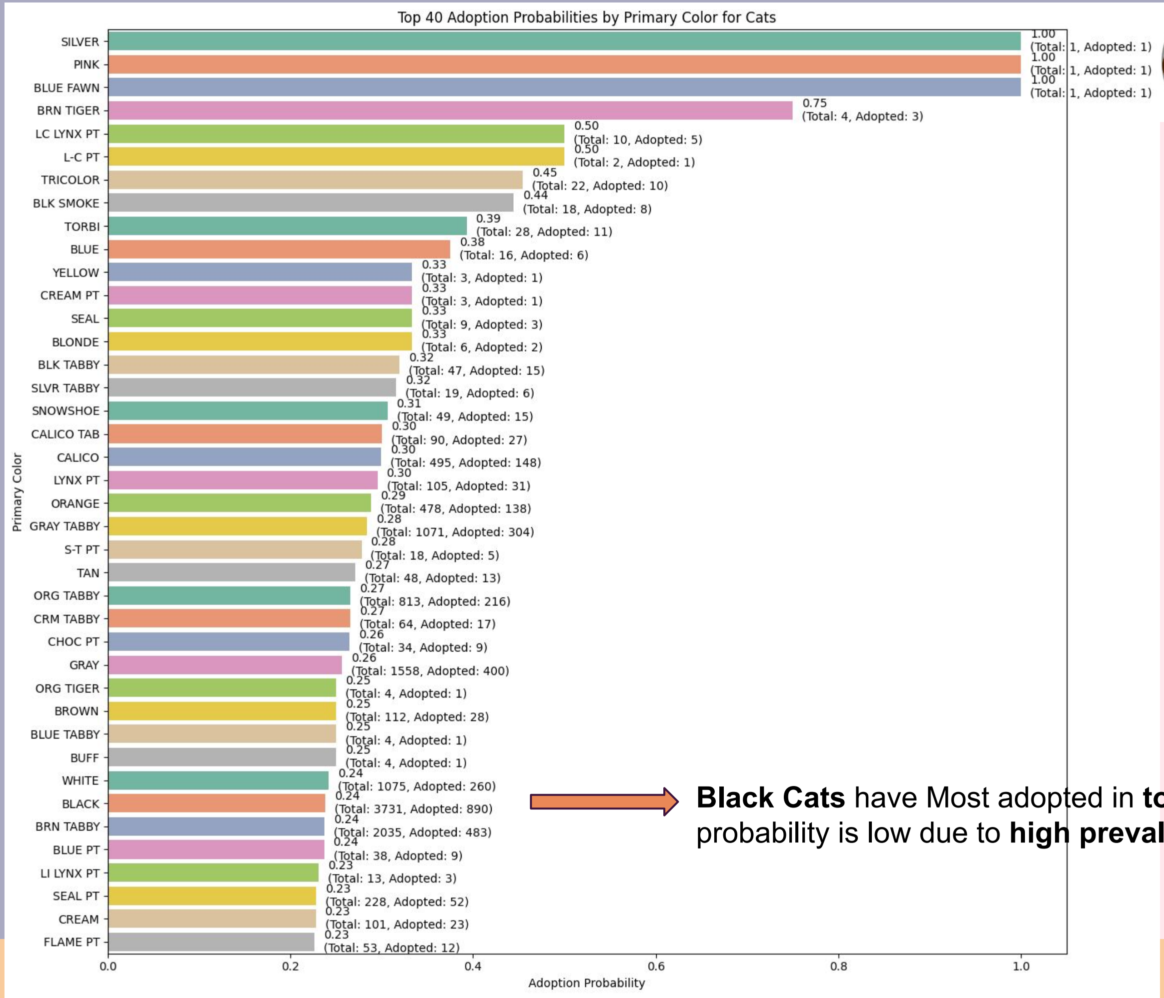
Adoption: Animals are directly adopted from the shelter by individuals or families, finding their forever homes.

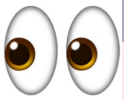
"Rescue" and "Adoption" are independent and do not overlap.





It shows **Black Cats** Have the highest adoption count





100% looks good, but N = 1?

Importance of considering both **probabilities** and **absolute counts** when analyzing adoption trends.

Black Cats have Most adopted in **total numbers**, but their adoption probability is low due to **high prevalence**.

```
average shelter duration by primary color:
Primary Color
PINK          169.000000
CALICO PT      43.500000
BRN TIGER      41.000000
TRICOLOR       40.000000
BLUE PT        36.184211
LC LYNX PT     35.000000
BLONDE         32.666667
BLK SMOKE      31.888889
CRM TABBY      31.812500
BL LYNX PT     31.500000
YELLOW         31.333333
UNKNOWN        30.250000
GRAY TABBY     30.175537
BROWN          29.580357
BLACK          27.040472
CALICO         26.937374
WHITE          26.786977
BLK TABBY      26.106383
GRAY           26.034018
BRN TABBY      25.742506
ORG TABBY      25.738007
TORTIE         25.130137
LILAC PT       24.600000
L-C PT         24.500000
TORBI          23.750000
SEAL           23.333333
SLVR TABBY     23.052632
CALICO TAB     22.188889
```

The Pink Cat has the longest average shelter duration (169 days),
but this could be due to their rarity (likely just one or very few cats).

Possible Interpretations:

- Rare colors may stay longer in the shelter due to lower visibility or niche appeal.
- Common colors tend to have shorter durations due to higher adoption turnover.

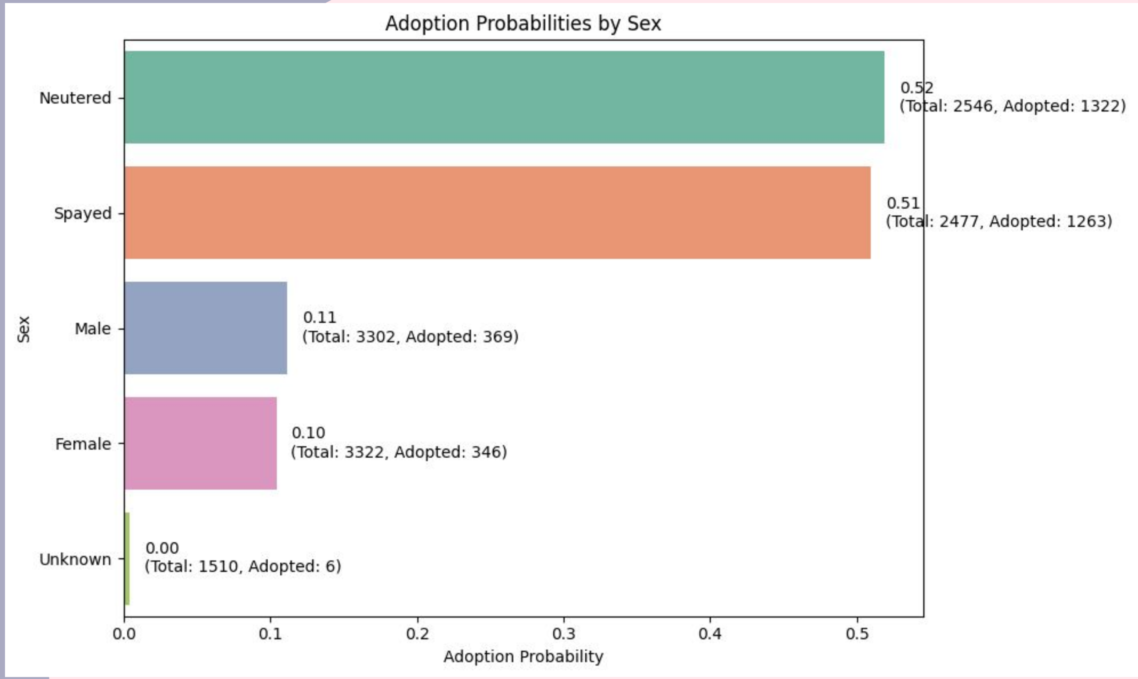
```
average shelter duration by sex:
Sex
Spayed      44.402099
Neutered    41.889631
Female      16.990367
Male        16.457299
Unknown      9.155629
```

NOTE: Spayed/Neutered cats and Male/Female cats are mutually exclusive categories and do not overlap.

Spayed/neutered cats stay longer because:

- Shelters wait to sterilize them before adoption.
- Sterilized cats are often promoted for adoption.

Spayed Cats: Refers to female cats that have undergone a surgical procedure called spaying.
Neutered Cats: Refers to male cats that have undergone a surgical procedure called neutering

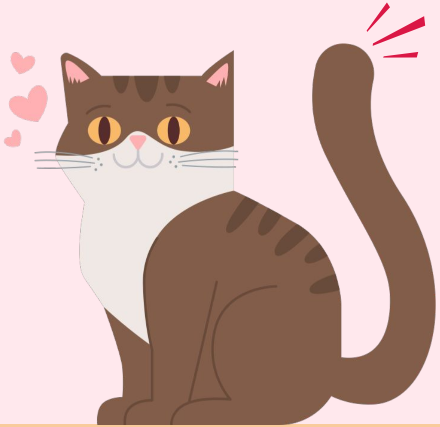


Sterilization plays a key role in adoption: Neutered or spayed cats are more likely to be adopted, possibly due to behavioral and health benefits.

Animals with **unknown sex** may face barriers due to incomplete or unclear records.

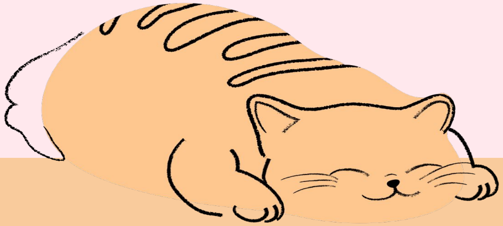
03

Data Processing



Raw Dataset

Animal ID
Animal Name
Animal Type
Primary Color
Secondary Color
Sex
DOB
Intake Date
Intake Condition
Intake Type
Intake Subtype
Outcome Date
Outcome Type
Outcome Subtype



Data source: https://data.longbeach.gov/explore/dataset/animal-shelter-intakes-and-outcomes/table

Data Cleaning Process

- Filter out duplicate Animal ID.
- Filter out animal other than cat.
- Keep only alive cat.
- Drop rows with null Date of Birth.
- Add column to mark if a cat has 2 colors.
- Using the outcome date to minus the date of birth to get the cats' age in human year.

```
lbdf = pd.read_csv('longbeach_animalshelter.csv')
lbdf['Outcome Date'] = pd.to_datetime(lbdf['Outcome Date'])
lbdf = lbdf.sort_values(by=['Animal ID', 'Outcome Date'], ascending=[True, False])
lbdf = lbdf.drop_duplicates(subset='Animal ID', keep='first')
```

```
lbdf = lbdf[lbdf['Animal Type']== 'CAT']
filtered_lbdf = lbdf[lbdf['was_outcome_alive']== 1]
filtered_lbdf = filtered_lbdf.drop(columns=['Animal Type', 'intake_is_dead', 'Animal Name',
                                            'outcome_is_dead', 'was_outcome_alive', 'geopoint'])
```

```
#Missing value
filtered_lbdf = filtered_lbdf[filtered_lbdf['DOB'].isnull() == False]
filtered_lbdf.loc[:,'Have2Color'] = filtered_lbdf['Secondary Color'].notna().astype(int)
filtered_lbdf
```

```
dob, outcome = pd.to_datetime(filtered_lbdf['DOB']), pd.to_datetime(filtered_lbdf['Outcome Date'])
agecol = outcome.apply(lambda x: x.year).astype('float') - dob.apply(lambda x: x.year).astype('float')
filtered_lbdf.loc[:,'Age in Human Year'] = agecol
filtered_lbdf
```

Data Cleaning Cont.

- Add column on when the event happened.
- Add column to mark if the cat is adopted or not.
- Keep only five features to fit the model.
    - Primary Color
    - Sex
    - Outcome Month
    - Have 2 Color
    - Age in Human Year

```
#adding outcome month since season might affect adoption and rescuetion
mont_map =  {
    1: 'Jan', 2: 'Feb', 3: 'Mar', 4: 'Apr', 5: 'May',
    6: 'Jun', 7: 'Jul', 8: 'Aug', 9: 'Sep', 10: 'Oct',
    11: 'Nov', 12: 'Dec'
}
filtered_lbdf.loc[:,'Outcome Month'] = outcome.dt.month
med_month = int(filtered_lbdf['Outcome Month'].median(skipna=True))
filtered_lbdf.loc[:,'Outcome Month'] = filtered_lbdf['Outcome Month'].fillna(med_month).astype(int).map(mont_map)
filtered_lbdf['Outcome Month']
```

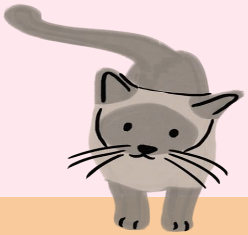
```
filtered_lbdf.loc[:,'isAdopted'] = (filtered_lbdf['Outcome Type'] == 'ADOPTION').astype('int')
drop_lbdf = filtered_lbdf.loc[:,["Primary Color", "Sex","Outcome Month","Have2Color","Age in Human Year"]]
ageMean = np.mean(filtered_lbdf['Age in Human Year'])
drop_lbdf.loc[:,'Age in Human Year'] = filtered_lbdf['Age in Human Year'].fillna(ageMean)
drop_lbdf.head()
```

Visualize

Primary Color obj...	Sex object	Outcome Month o..	Have2Color int64	Age in Human Year
GRAY	Neutered	Jul	1	11
BLACK	Spayed	Jan	0	9
BLACK	Spayed	Mar	0	10
BROWN	Neutered	Sep	0	12
BLACK	Spayed	Dec	0	15

After Data Cleaning

Animal ID			
Animal Name			
Animal Type			
Primary Color		string -> int	
Secondary Color		string -> int	
Sex		string -> int	
DOB			
Intake Date	>>>>>>>>>>>>>>>		
Intake Condition			
Intake Type			
Intake Subtype			
Outcome Date			
Outcome Type			
Outcome Subtype			
		Have2Color	
		AgeInHumanYear	Cat's actual age
		OutcomeMonth	How long spent in shelter
		isAdopted	

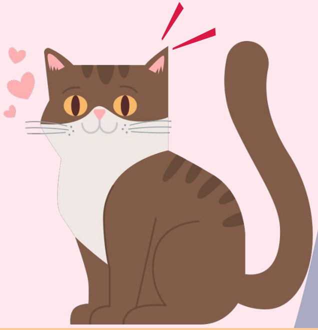


04

Model Presenting

Train-Test Split for All Models

```
target = 'isAdopted'
features = ["Primary Color", "Sex","Outcome Month","Have2Color","Age in Human Year"]
X = drop_lbdf.loc[:,features]
y = drop_lbdf[target]
print(X.columns)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```



1. Random Forest

- Reduces the chance of Overfitting.
- Enhances Model Stability and Accuracy.

(random forest cat in a random forest)

Image source: https://www.reddit.com/r/aww/comments/pfa60u/forest_cat_being_a_forest_cat/?rdt=60927

Fitting the model

By using grid search combine with cross validation, I tried to find the optimal value for max features.

```python
#RandomForest: first using cross validation to see the best generalized model
param_grid = {
    'max_features': range(1, X_train.shape[1])
}

rf = RandomForestClassifier(n_estimators=500, random_state=42)
cvrfgrid_search = GridSearchCV(estimator=rf,
                               param_grid=param_grid, cv=5,
                               scoring='accuracy')
cvrfgrid_search.fit(X_train, y_train)

best_cvrf_model = cvrfgrid_search.best_estimator_
cvrf_y_pred = best_cvrf_model.predict(X_test)
```

```
best accuracy:  0.7581506906341463
best "max features": {'max_features': 2}
Test accuracy: 0.75765225623225
```

2. LightGBM

Why?

- combining categorical and numerical features
- Flexibility (num_leaves, learning_rate, max_depth)
- Cross-Validation support
- Can handle missing values

<u>**accuracy, speed, and interpretability**</u>

How We Approach

- K-Folds
- Cross-Validation by creating LightGBM datasets and training the model

```python
# Initialize KFold
kf = KFold(n_splits=5, shuffle=True, random_state=42)

# LightGBM parameters
lgb_params = {
    'objective': 'binary',
    'metric': 'binary_error',
    'boosting_type': 'gbdt',
    'learning_rate': 0.05,
    'num_leaves': 31,
    'max_depth': -1,
    'feature_fraction': 0.8,
    'bagging_fraction': 0.8,
    'bagging_freq': 5,
    'verbose': -1
}

accuracy_scores = []

# Cross-Validation Loop
for train_index, val_index in kf.split(X):
    # Splitting the data
    X_train, X_val = Xg.iloc[train_index], Xg.iloc[val_index]
    y_train, y_val = yg.iloc[train_index], yg.iloc[val_index]

    # Creating LightGBM datasets
    train_data = lgb.Dataset(X_train, label=y_train)
    val_data = lgb.Dataset(X_val, label=y_val)
```

```python
# Train the model
model = lgb.train(
    lgb_params,
    train_data,
    valid_sets=[train_data, val_data],
    num_boost_round=100,
    #early_stopping_rounds=50,
    #verbose_eval=100
)

# Predict on the validation set
y_pred = model.predict(X_val, num_iteration=model.best_iteration)
y_pred_binary = [1 if pred > 0.5 else 0 for pred in y_pred]

# Calculate accuracy
accuracy = accuracy_score(y_val, y_pred_binary)
accuracy_scores.append(accuracy)
```

- Problems & Resolution

❖ Long time run → adjusting the num_boost_round parameter in LightGBM training by starting with a small number

- Results!

```
# Display results
print(f'Average Training Accuracy: {np.mean(training_accuracies):.4f}')
print(f'Average Validation Accuracy: {np.mean(validation_accuracies):.4f}
print(f'Test Accuracy: {test_accuracy:.4f}')
```

```
Average Training Accuracy: 0.8028
Average Validation Accuracy: 0.7822
Test Accuracy: 0.7674
```

3. XGBoost (eXtreme Gradient Boosting)

Why?

- High accuracy
- Good for large dataset

Baseline Model (default)

```python
model = XGBClassifier(random_state=42, eval_metric='logloss')
model.fit(X_train_full_xgb, y_train_full_xgb)

y_train_pred = model.predict(X_train_full_xgb)
y_test_pred = model.predict(X_test_xgb)


print(f'Baseline Training Accuracy: {accuracy_score(y_train_full_xgb, y_train_pred):4f}')
print(f'Baseline Test Accuracy: {accuracy_score(y_test_xgb, y_test_pred):4f}')
```

```
Baseline Training Accuracy: 0.835293
Baseline Test Accuracy: 0.767024
```

Hyperparameter Tuning

- Optuna

```python
!pip install optuna
import optuna
def objective(trial):
    params = {
        'learning_rate': trial.suggest_float('learning_rate', 0.01, 0.3),
        'objective':'binary:logistic',
        'max_depth': trial.suggest_int('max_depth', 3, 10),
        'min_child_weight': trial.suggest_int('min_child_weight', 1, 10),
        'subsample': trial.suggest_float('subsample', 0.6, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.6, 1.0),
        'n_estimators': trial.suggest_int('n_estimators', 50, 300),
        'random_state': 42
    }
    model = XGBClassifier(**params,eval_metric='logloss')
    model.fit(X_train_full_xgb, y_train_full_xgb)
    y_pred = model.predict(X_test_xgb)
    return accuracy_score(y_test_xgb, y_pred)

# Optimize hyperparameters
study = optuna.create_study(direction='maximize')
study.optimize(objective, n_trials=100)  # Number of trials to run

best_params = study.best_trial.params
# Best trial and hyperparameters
print("Best Hyperparameters:", best_params)
```

```
Best Hyperparameters:
      {'learning rate': 0.09716861273053744,
       'max depth': 6,
       'min child weight': 1,
       'subsample': 0.9949938930288234,
       'colsample bytree': 0.9658761587200018,
       'n_estimators': 80}
```

- Final Model

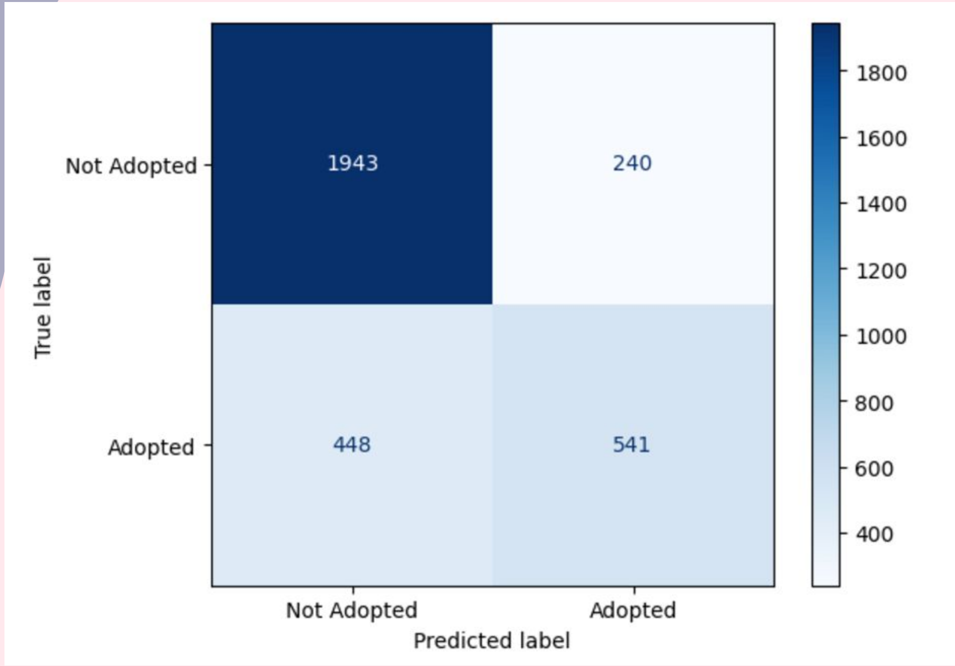
```python
#final XGBoost Model
xgb = XGBClassifier(learning_rate = 0.09716861273053744,
                    max_depth = 6,
                    min_child_weight=1,
                    subsample=0.9949938930288234,
                    colsample_bytree=0.9658761587200018,
                    n_estimators=80,
                    random_state=42)

xgb.fit(X_train_full_xgb, y_train_full_xgb)
y_train_pred_xgb = xgb.predict(X_train_full_xgb)
y_test_pred_xgb = xgb.predict(X_test_xgb)

print(f'Train Accuracy: {accuracy_score(y_train_full_xgb, y_train_pred_xgb):4f}')
print(f'Test Accuracy: {accuracy_score(y_test_xgb, y_test_pred_xgb):4f}')
```

```
Train Accuracy: 0.802054
Test Accuracy: 0.783102
```

Confusion Matrix sklearn



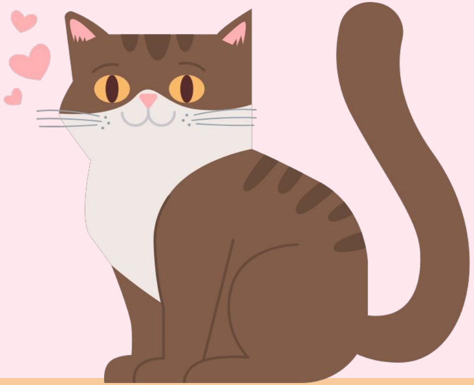
True Positive Rate: 0.547017

True Negative Rate: 0.890060

F-score: 0.611299

05

Conclusion



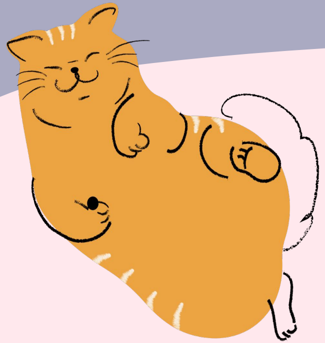
Model selecting

	Training Accuracy	Testing Accuracy
Random Forest	0.7582	0.7576
LightGBM	0.8028	0.7674
XGBoost	0.8082	0.7812

Accuracy Order: Random Forest < LightGBM < XGBoost

(0.7576 < 0.7674 < 0.7812)

No overfitting in any model



How it Benefit Stakeholders

Getting know what impact adoption rate for different kind of cats and promote low adoption rate genre.

- creating TikTok account and upload cute videos about them
- Rearrange shelter
- Award giving

Things Shelter can do right now

Checking the Data:

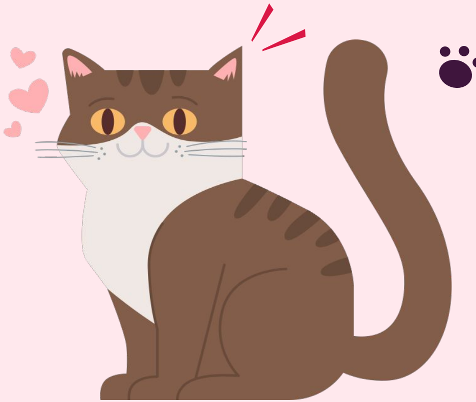
- Make sure to only counting the animals that truly come into and go out of the shelter.

Opening Hours:

- Keep the shelter open later and on weekends when more people can visit.

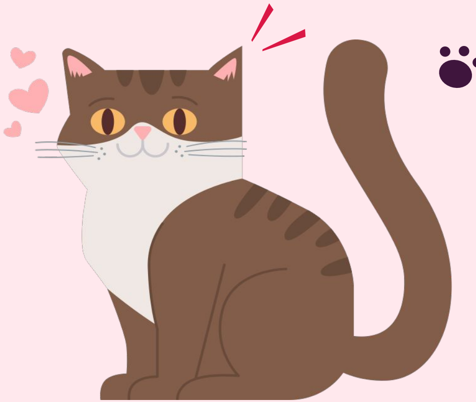
Educate the Publics:

- Let people know the lifesaving impact they can make when they adopt a pet from a shelter instead of buying one.



Things We can do right now

- Adopt
- Donate
- Volunteer



Thank You

Do you have any questions?


CAT
BOOST