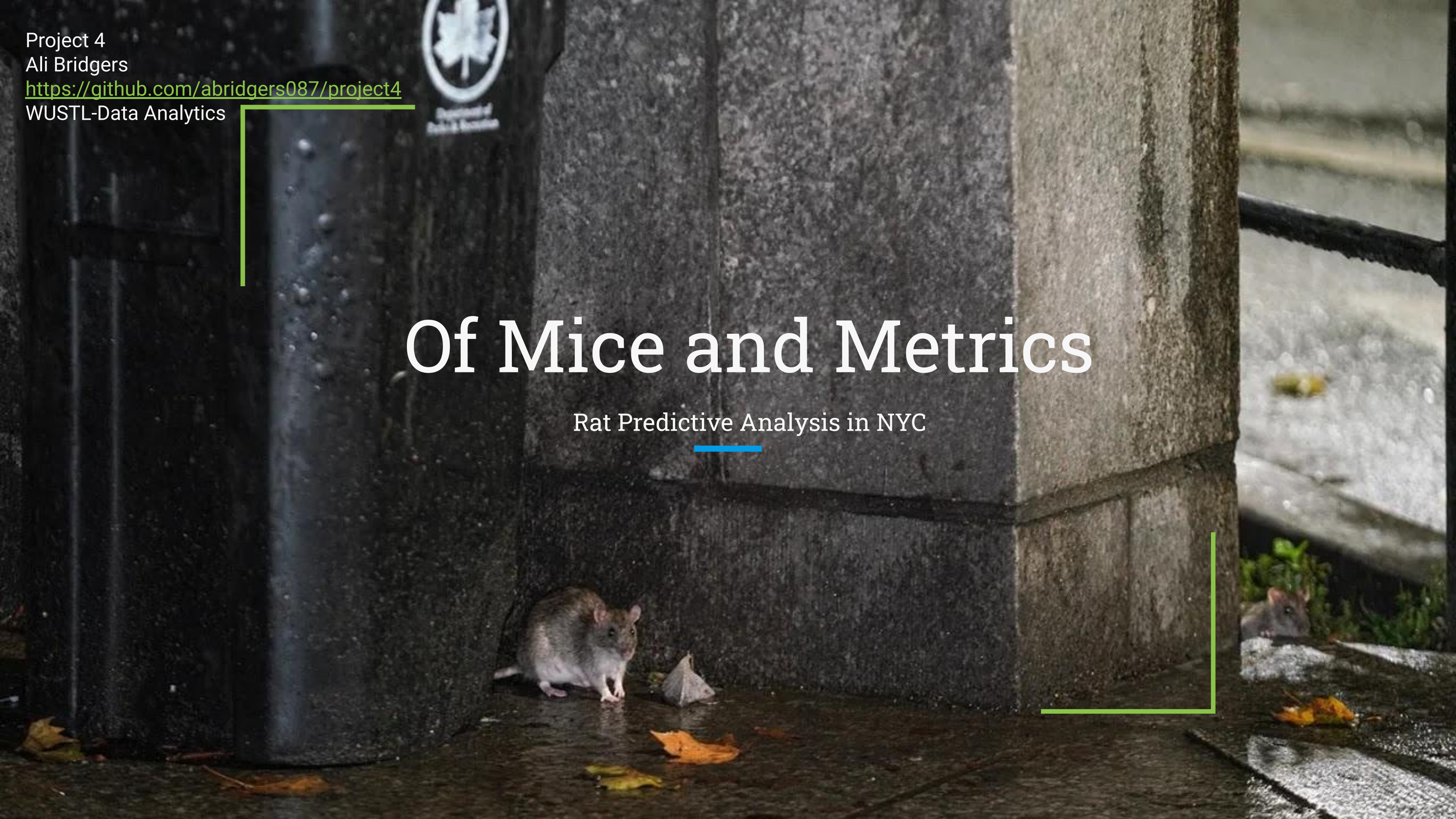


Project 4
Ali Bridgers
https://github.com/abridgers087/project4
WUSTL-Data Analytics

Of Mice and Metrics

Rat Predictive Analysis in NYC

RATS!

Rat Information

- Surge in NYC rat sightings (311 complaints), possibly from pandemic dining
- NYC's first rodent mitigation director appointed in April 2023
- Estimated 2023 rat population: 3,000,000

NYC Overall Strategy

- Multi-agency strategy targets conditions aiding rat growth
- Agencies: NYC Health, Parks & Rec, Sanitation, Education, Housing Authority.
- Goal: Reduce food, water, shelter, and exterminate rats.

RMZs and the RIP

- Rat Mitigation Zone (RMZ): city effort against rat growth conditions
- Rat Information Portal (RIP): Map detailing rat inspections and actions
- Together represent NYC's data-led commitment to rat mitigation transparency.

Data Source

https://data.cityofnewyork.us/Health/Rodent-Inspection/p937-wjvj

- Dataset provided by the Department of Health and Mental Hygiene
- Owned by NYC OpenData and updated daily
- 2.44M Rows (each row is a rodent inspection)
- 25 Columns

Data Limitations: if a property/taxlot does not appear in the file, that does not indicate an absence of rats - rather just that it has not been inspected. Similarly, neighborhoods with higher numbers properties with active rat signs may not actually have higher rat populations but simply have more inspections

Using NYC ‘Rat Data’, can we train a machine learning algorithm to predict rat activity based on location?

NYC 'RAT DATA' REFINEMENT

CENSUS TRACT

BIN

Y_COORD

LONGITUDE

NTA

BOROUGH

COUNCIL DISTRICT

JOB_PROGRESS

INSPECTION_TYPE

BLOCK

HOUSE_NUMBER

LATITUDE

BBL

LOT

COMMUNITY BOARD

JOB_TICKET_OR_WORK_ORDER_ID

RESULT

ZIP_CODE

X_COORD

STREET_NAME

JOB_ID

BORO_CODE

APPROVED_DATE

INSPECTION_DATE

LOCATION

NYC ‘RAT DATA’ REFINEMENT

LONGITUDE

BOROUGH

INSPECTION_TYPE

LATITUDE

RESULT

ZIP_CODE

INSPECTION_DATE

NYC 'RAT DATA' REFINEMENT

INSPECTION_TYPE	RESULT	BOROUGH	ZIP_CODE	LATITUDE	LONGITUDE	INSPECTION_DATE
(string)	(string)	(string)	(int32)	(number)	(number)	(change to 'Date' format)

```python
fields_to_check = [
    "INSPECTION_TYPE",
    "ZIP_CODE",
    "BOROUGH",
    "INSPECTION_DATE",
    "RESULT"
]

query = {"$or": []}
for field in fields_to_check:
    query["$or"].extend([
        {field: {"$exists": False}},
        {field: ""},
        {field: {"$regex": "^\s*$"}}
    ])
query["$or"].append({"ZIP_CODE": 0})

coord_fields_to_check = ["LATITUDE", 'LONGITUDE']

for field in coord_fields_to_check:
    query["$or"].extend([
        {field: {"$exists": False}},
        {field: ""},
        {field: {"$regex": "^\s*$"}},
        {field: 0}
    ])

result = collection.delete_many(query)

print(f"Deleted {result.deleted_count} documents.")
```

```
_id: ObjectId('64ec98afacbf970d297d8d3c')
INSPECTION_TYPE: "Initial"
ZIP_CODE: 11385
LATITUDE: 40.708494850783
LONGITUDE: -73.919695513998
BOROUGH: "Queens"
INSPECTION_DATE: 2023-06-05T15:20:34.000+00:00
RESULT: "Rat Activity"
```

```
_id: ObjectId('64ec98afacbf970d297d85ed')
INSPECTION_TYPE: "Compliance"
ZIP_CODE: 10459
LATITUDE: 40.82931148005
LONGITUDE: -73.896505867413
BOROUGH: "Bronx"
INSPECTION_DATE: 2023-03-22T21:00:25.000+00:00
RESULT: "Passed"
```

NYC 'RAT DATA' REFINEMENT

INSPECTION_TYPE

RESULT

BOROUGH

ZIP_CODE

LATITUDE

LONGITUDE

INSPECTION_DATE

INITIAL: conducted in response to 311 complaints or proactive neighborhood checks

COMPLIANCE: conducted if property fails initial

BAIT: bait/rodenticide applied by Health Dept

STOPPAGE: sealing of holes/cracks

CLEANUPS: removal of garbage/clutter around a property

PASSED

RAT ACTIVITY

BAIT APPLIED

FAILED FOR OTHER R

MONITORING VISIT

STOPPAGE DONE

CLEANUP DONE

STATEN ISLAND

QUEENS

BRONX

BROOKLYN

MANHATTAN

Date:

2023-06-07T21:36:13.000+00:00

```
#Remove all documents with an Inspection Date before 2023
# Define the date threshold
threshold_date = datetime(2023, 1, 1)

# Remove documents with an INSPECTION_DATE before 2023
result = collection.delete_many({
    "INSPECTION_DATE": {
        "$lt": threshold_date
    }
})

print(f"{result.deleted_count} documents were deleted.")
```

RAT-CHETING UP THE 'RAT DATA'

Data and Feature Engineering

In Mongo:
- ✓ Load Data
- ✓ Clean Data

In Python:
- ☑ Load Data into a Dataframe

```
cursor = collection.find({})
df = pd.DataFrame(list(cursor))
```

```
df.drop(columns=['_id'], inplace=True)
```

	INSPECTION_TYPE	ZIP_CODE	LATITUDE	LONGITUDE	BOROUGH	INSPECTION_DATE	RESULT
0	Initial	12345	40.817678	-73.941974	Manhattan	2023-03-08 15:21:41	Passed
1	Initial	11377	40.738373	-73.906470	Queens	2023-06-05 20:19:22	Passed
2	Initial	10457	40.850038	-73.894424	Bronx	2023-07-17 16:05:21	Passed
3	BAIT	11385	40.708495	-73.919696	Queens	2023-04-20 17:18:23	Bait applied
4	Initial	10470	40.897316	-73.863219	Bronx	2023-03-14 13:40:50	Failed for Other R
...	...	...	...	...	...	...	...
162372	Initial	10065	40.764009	-73.966893	Manhattan	2023-02-03 16:55:09	Passed
162373	Initial	10458	40.856980	-73.886359	Bronx	2023-02-03 20:30:05	Passed
162374	Compliance	11211	40.707009	-73.951506	Brooklyn	2023-05-26 17:10:32	Rat Activity
162375	Initial	11206	40.694630	-73.935954	Brooklyn	2023-02-03 21:00:12	Rat Activity
162376	Initial	11377	40.746886	-73.896421	Queens	2023-03-02 18:51:42	Passed

RAT-CHETING UP THE 'RAT DATA'

Back to our key question:

Can machine learning predict **rat activity** based on location?

Our dataset lacks a column for rat activity...

...by using feature engineering, we can create RAT_ACTIVITY based on what we know about the INSPECTION_TYPE and RESULTS columns!

```
#RAT ACTIVITY IS THE TARGET OF OUR ML MODEL
# Create a new column "Rat Activity" and initialize with 0
df['RAT_ACTIVITY'] = 0

# Set the "Rat_Activity" column to 1 where there is rat activity
df.loc[
    (df['INSPECTION_TYPE'] == 'Initial') &
    (df['RESULT'] == 'Rat Activity'),
    'RAT_ACTIVITY'
] = 1

df.loc[
    (df['INSPECTION_TYPE'] == 'Compliance') &
    (df['RESULT'] == 'Rat Activity'),
    'RAT_ACTIVITY'
] = 1

df.loc[
    df['INSPECTION_TYPE'].isin(['BAIT', 'STOPPAGE', 'CLEAN_UPS']),
    'RAT_ACTIVITY'
] = 1


df
```

	INSPECTION_TYPE	ZIP_CODE	BOROUGH	INSPECTION_DATE	RESULT	RAT_ACTIVITY
0	Initial	10469	Bronx	2023-03-10 20:10:27	Passed	0
1	Initial	10029	Manhattan	2023-03-24 12:30:00	Rat Activity	1
2	Initial	10027	Manhattan	2023-01-20 19:31:22	Passed	0
3	Initial	11221	Brooklyn	2023-05-12 18:22:44	Passed	0
4	Initial	10451	Bronx	2023-01-19 21:08:39	Passed	0
...	...	...	...	...	...	...
165367	Initial	10065	Manhattan	2023-02-03 16:55:09	Passed	0
165368	Initial	10458	Bronx	2023-02-03 20:30:05	Passed	0
165369	Compliance	11211	Brooklyn	2023-05-26 17:10:32	Rat Activity	1
165370	Initial	11206	Brooklyn	2023-02-03 21:00:12	Rat Activity	1
165371	Initial	11377	Queens	2023-03-02 18:51:42	Passed	0

RAT-CHETING UP THE ‘RAT DATA’

Getting Model-Ready: Standardizing Our Dataset

ZIP_CODE	INSPECTION_DATE	LAT & LON	INSPECTION_TYPE & BOROUGH
➢ Create a dictionary to link zip code to mean rat activity ➢ Transform original zip code column into ZIP_CODE_ ENCODED ➢ Drop ZIP_CODE	➢ Convert to standard pandas datetime ➢ Extract month, store in new column: INSPECTION_ MONTH ➢ Drop INSPECTION_DATE	➢ Normalize with MinMaxScaler	➢ Use pd.get_dummies ➢ One-hot encoded data to convert to a binary format ➢ Drop first column in each encoded category to avoid multicollinearity

TUNING TO 'RAT DATA'

(using Keras & Hyperband)

```python
def build_model(hp):
    model = tf.keras.models.Sequential()

    # Hidden Layers
    model.add(tf.keras.layers.Dense(
        units=hp.Int('units_layer1', 32, 256, 32),
        activation="relu",
        input_dim=X_train_scaled.shape[1]
    ))
    model.add(tf.keras.layers.Dense(
        units=hp.Int('units_layer2', 16, 128, 16),
        activation="relu"
    ))

    # Output Layer
    model.add(tf.keras.layers.Dense(1, activation="sigmoid"))
    model.summary()

    # Compile
    model.compile(
        optimizer=tf.keras.optimizers.Adam(
            learning_rate=hp.Choice('learning_rate', [1e-2, 1e-3, 1e-4])
        ),
        loss='binary_crossentropy',
        metrics=['accuracy']
    )
    return model
```

- Sequential Model: defines a neural network model using TensorFlow's Keras API using a sequential model (stacked layers).
- Hidden Layers: uses two densely connected hidden layers. The neuron counts (units) are made tunable through Hyperband's search space, with the first layer having between 32 to 256 neurons and the second layer having between 16 to 128 neurons. Both use the ReLU activation function.
- Hyperparameter Tuning: By integrating with the Hyperband algorithm, this model can be automatically tuned. The adjustable hyperparameters in this setup include the neuron count in the hidden layers and the learning rate for the optimizer.

TUNING TO 'RAT DATA'

(using Keras & Hyperband)

Best Hyperparameters:

First Hidden Layer = 224 Units

Second Hidden Layer = 96

Best Learning Rate: 0.001

Model Evaluation:

1269/1269 - 1s

Loss: 0.3975

Accuracy 0.8048

1s/epoch, 1ms/step

- Data-Driven Strength: 80.5% accuracy with our baseline model.
- Room for Improvement: 39.74% loss suggests potential areas for refinement.
- Complexity of Real-World Data: Loss underscores the dynamic nature of urban environments.
- Emphasis on Adaptation: Continuous model tweaking essential to address evolving rat behavior in cities.

Prediction Test

Now that we have created our learning model with the best parameters, let's test it on some data...

```
# Predict using the model
predicted_rat_activity = best_model.predict(df_ml2)

# Convert to DataFrame
predicted_df = pd.DataFrame(
    predicted_rat_activity,
    columns=['Predicted_RAT_ACTIVITY']
)

# Convert predictions to class labels
predicted_class_labels = (
    (predicted_rat_activity > 0.5).astype(int)
)

# Add to main dataframe
df2['Predicted_RAT_ACTIVITY'] = predicted_class_labels

# Drop unnecessary column
df2.drop(columns=['ZIP_CODE_ENCODED'], inplace=True)

# Display the dataframe
df2
```

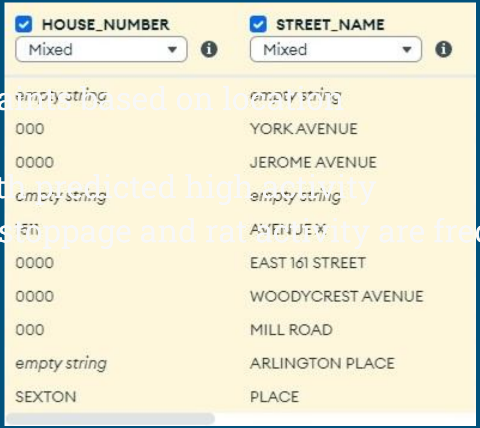
INSPECTION_TYPE	RESULT	Predicted_RAT_ACTIVITY
Compliance	Rat Activity	1
BAIT	Bait applied	1
Initial	Passed	0
BAIT	Monitoring visit	1
Initial	Passed	0
...	...	...
Initial	Rat Activity	0
BAIT	Bait applied	1
BAIT	Bait applied	1
BAIT	Bait applied	1
BAIT	Bait applied	1

Takeaways

Main Question: Using NYC ‘Rat Data’, can we train a machine learning algorithm to predict rat activity based on location?

Answer: Yes, but…

Challenges: Location: I used Borough, Zip Code, Latitude and Longitude values for the location fields. Street Name would have been my preferred column, however…



Future Use: Utilize our model to prioritize 311 complaints based on location
Expand or shrink current RMZ’s
Target ‘Rat Info’ campaigns in areas with predicted high activity
Invest in infrastructure in areas where stoppage and rat activity are frequent.