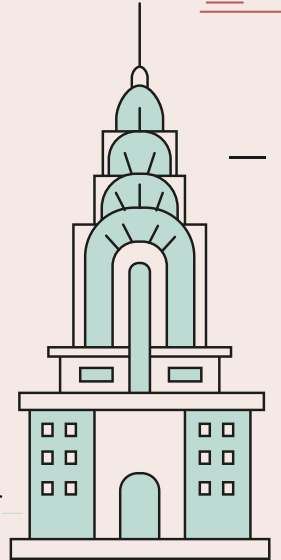
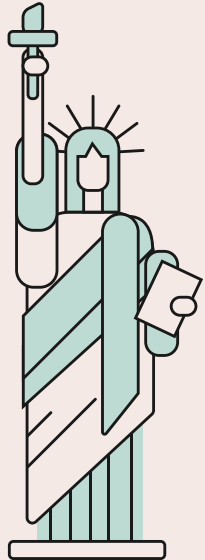




# NEW YORK

## House Price Estimator

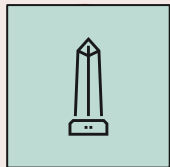
Leticia Genao



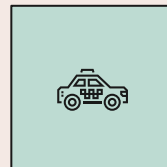
# PROJECT DEFINITION



## Problem Statement:



The goal is to provide accurate price predictions for residential properties in New York City. This tool aids buyers, sellers, and investors by utilizing advanced machine learning techniques to evaluate various property features.



## Dataset Overview?



**Source:** Kaggle

**Final Features:** bedrooms, bathrooms, square footage, type (one-hot encoded), borough (one-hot encoded)

**Target:** price

## Why Machine Learning?



Traditional methods often involve manual appraisal, which can be biased and inconsistent. Machine learning offers a scalable, objective, and efficient alternative that can handle complex datasets and uncover hidden patterns in the real estate market.

**Final Shape:** 3878x12



# DATA EXPLORATION AND PREPROCESSING



01


## Initial Data Analysis

The data exploration phase involved generating statistical summaries to understand distributions and potential outliers.

03

## Feature Engineering

The data was refined by creating a 'borough' feature based on 'locality' mappings, simplifying property 'type' into more general categories, and dropping rows categorized as 'Other' under 'type'.



02

## Data Cleaning Steps

The preprocessing involved handling missing values through imputation, and outlier detection using IQR score methods to ensure model accuracy

04

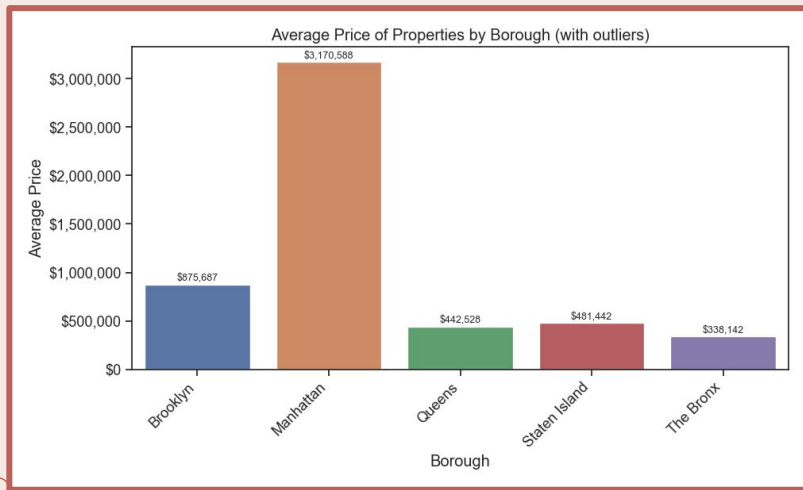
## Final Model Features

**price:** Housing price  
**beds:** Number of bedrooms  
**bath:** Number of bathrooms  
**propertysqft:** Property square footage  
**borough\_Manhattan,**  
**borough\_Queens, borough\_Staten**  
**Island, borough\_The Bronx**  
**type\_Condo, type\_House,**  
**type\_Multi-family, type\_Townhouse**

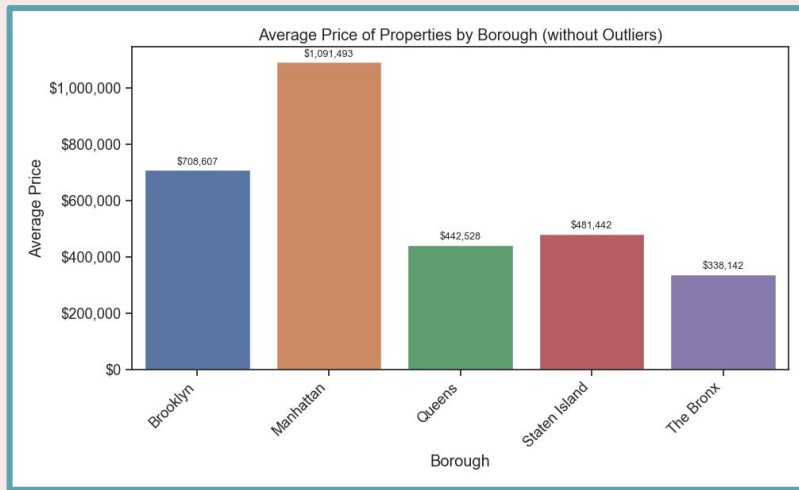


# DATA EXPLORATION AND PREPROCESSING

## Price with Outliers



## Price without Outliers



# MODEL DEVELOPMENT



## Algorithm Selection:

**Tested:** Linear Regression,  
RandomForest, and GradientBoosting

**Selected:** GradientBoosting



## Performance Metrics:

### Validation Performance:

**R<sup>2</sup> Score:**

0.610

**Root Mean Squared Error (RMSE):**

386,537.91



## Training Process:

**60%** Training split

**20%** Validation split

**20%** Testing split

### Test Performance:

**R<sup>2</sup> Score:**

0.592

**Root Mean Squared Error (RMSE):**

399,797.70



# CONTAINERIZATION WITH DOCKER

## Docker Setup

Python environment and dependencies are encapsulated within the Docker container

## Docker Hub

Docker image is made available for public use, for sharing/reproducing the project setup easily

## Local Testing

Docker container is tested locally to ensure it functions correctly, mimicking production environment behavior

```
Week4 > my_housing > Part1_Local_and_Heroku_Deployment > Dockerfile > ...
You, 11 hours ago | author (You)
1 # Using an official Python runtime as a parent image
2 FROM python:3.10-slim
3
4 # Setting the working directory to /app
5 WORKDIR /app
6
7 # Copying the current directory contents into the container at /app
8 COPY . /app
9
10 # Installing any needed packages specified in requirements.txt
11 RUN pip install --no-cache-dir -r requirements.txt
12
13 # Making port 5000 available to the world outside this container
14 EXPOSE 5000
15
16 # Defining environment variable
17 ENV NAME World
18
19 # Running app.py when the container launches
20 CMD ["gunicorn", "app:app", "--config", "gunicorn_config.py"]
```

leticiagenao/ny-house-price-estimator

By leticiagenao · Updated about 9 hours ago  
New York Housing Price Estimator: Uses ML to predict NYC housing prices.

Overview Tags

Docker Repository: NY House Price Estimator

This docker image contains a Flask application for estimating housing prices in New York City. It's built to offer users a quick and reliable way to predict real estate values based on input features like number of bedrooms, bathrooms, and location.

How to Use This Image

Pull the image:

- docker pull leticiagenao/ny-house-price-estimator:latest

Here's an example:

```
docker run -p 5000:5000 leticiagenao/ny-house-price-estimator:latest
```

After running the container, access the application via <http://localhost:5000> in your web browser.

Dependencies

- Python 3.8
- Flask
- Pandas
- Scikit-Learn
- NumPy

This image is maintained and updated by Leticia Genao. It is intended for educational and demonstration purposes in the field of data science and machine learning.

```
P5 C:\Users\leticia\coding_projects\my_projects\NationalUniversity\WANA680\Week4\my_housing> docker pull ny-house-price-est
"docker pull" requires exactly 1 argument.
See "docker pull --help".

Usage: docker pull [OPTIONS] NAME[:TAG][@DIGEST]

Download an image from a registry
P5 C:\Users\leticia\coding_projects\my_projects\NationalUniversity\WANA680\Week4\my_housing> docker run -p 5000:5000 letic
>>
[2024-06-01 19:45:20 +0000] [1] [INFO] Starting gunicorn 22.0.8
[2024-06-01 19:45:20 +0000] [1] [INFO] Listening at: http://0.0.0.0:5000 (1)
[2024-06-01 19:45:20 +0000] [1] [INFO] Using worker: sync
[2024-06-01 19:45:20 +0000] [7] [INFO] Booting worker with pid: 7
[2024-06-01 19:45:20 +0000] [8] [INFO] Booting worker with pid: 8
[2024-06-01 19:45:21 +0000] [24] [INFO] Booting worker with pid: 24
[2024-06-01 19:46:11 +0000] [1] [CRITICAL] WORKER TIMEOUT (pid:8)
[2024-06-01 19:46:11 +0000] [8] [ERROR] Error handling request (no URI read)
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/site-packages/gunicorn/workers/sync.py", line 134, in handle
    req = next(parsers)
  File "/usr/local/lib/python3.10/site-packages/gunicorn/http/parser.py", line 42, in __next__
    self.msg = self.msg_class(self.cfg, self.unreader, self.source_addr, self.req_count)
  File "/usr/local/lib/python3.10/site-packages/gunicorn/http/message.py", line 237, in __init__
    super().__init__(cfg, unreader, peer_addr)
  File "/usr/local/lib/python3.10/site-packages/gunicorn/http/message.py", line 60, in __init__
    unread = self.parse(self.unreader)
  File "/usr/local/lib/python3.10/site-packages/gunicorn/http/message.py", line 269, in parse
    self.get_data(unreader, buf, stop=True)
  File "/usr/local/lib/python3.10/site-packages/gunicorn/http/message.py", line 260, in get_data
    data = unreader.read()
  File "/usr/local/lib/python3.10/site-packages/gunicorn/http/unreader.py", line 37, in read
    d = self.chunk()
  File "/usr/local/lib/python3.10/site-packages/gunicorn/http/unreader.py", line 64, in chunk
    return self.sock.recv(self.mechunk)
  File "/usr/local/lib/python3.10/site-packages/gunicorn/workers/base.py", line 203, in handle_abort
    sys.exit(1)
SystemExit: 1
[2024-06-01 19:46:11 +0000] [8] [INFO] Worker exiting (pid: 8)
[2024-06-01 19:46:11 +0000] [103] [INFO] Booting worker with pid: 103
```

# DEPLOYMENT TO HEROKU



## WHY HEROKU DEPLOYMENT?

The Flask-based application was deployed on Heroku, leveraging its robust ecosystem and easy-to-use platform, which simplifies application scaling and management.




## LIVE DEMO

Experience the application firsthand via the live demo on Heroku. Interact with the model to see how it estimates New York housing prices based on various input features!

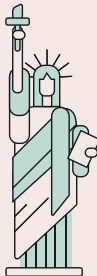
## CHALLENGES:

Deploying on Heroku involved overcoming challenges related to dyno management and Github Actions

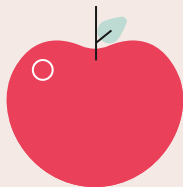
Efforts were made to ensure that the application remains responsive and efficient by managing how dynos sleep and wake up to handle user requests without delay.



<https://nyhouseprice-c155a038476a.herokuapp.com/>

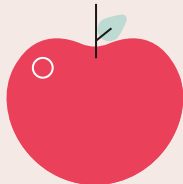


# AWS INTEGRATION AND CHALLENGES



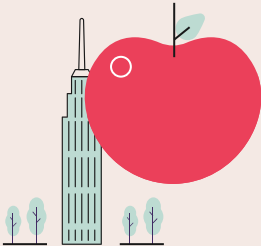
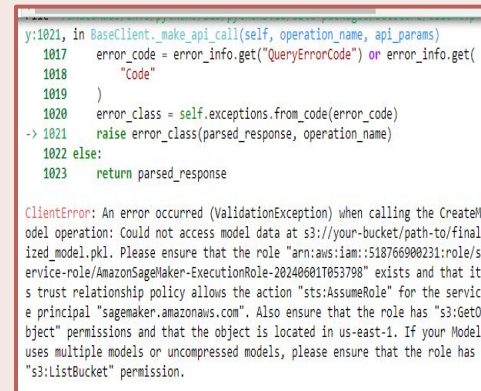
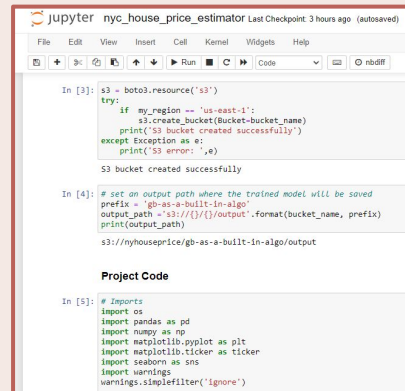
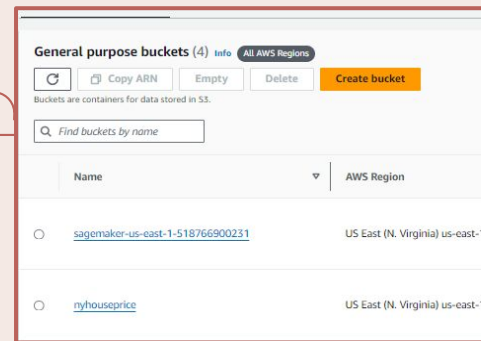
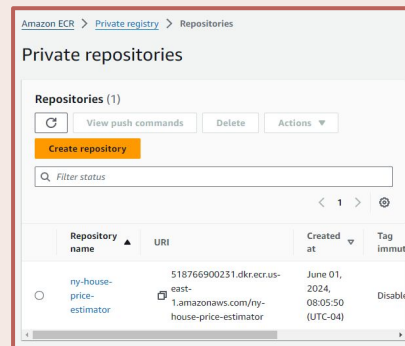
## AWS Services Used

Utilized AWS Elastic Container Registry (ECR) for Docker image management and S3 for secure model data storage



## SageMaker Notebook

Leveraged AWS SageMaker's cloud-based Jupyter notebooks for model development



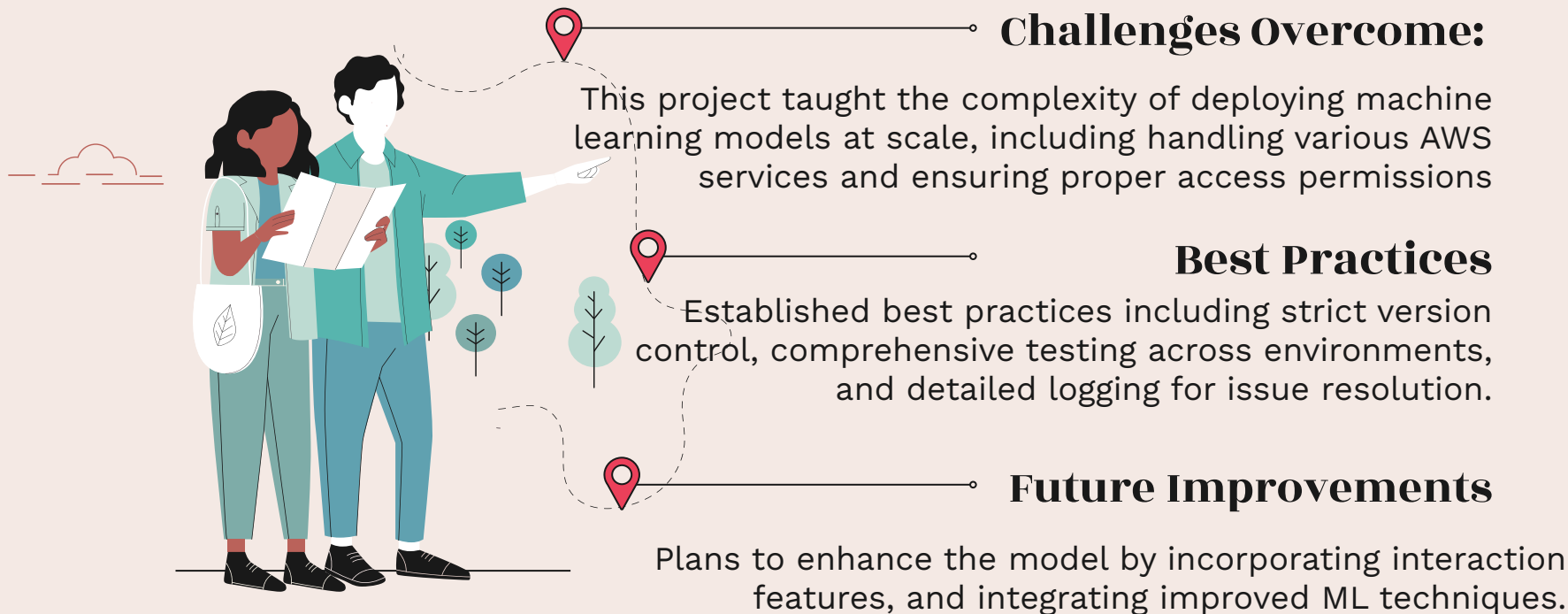
## Deployment Issues

Encountered significant challenges with SageMaker deployments, particularly issues related to IAM permissions and endpoint configurations





# Lessons Learned and Best Practices

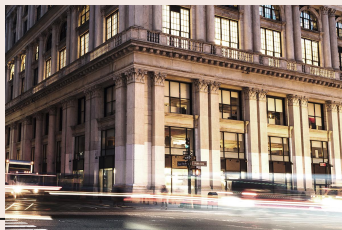


# Conclusion and Q&A



## Summary

The NY Housing Price Estimator is a testament to the power of machine learning in transforming real estate analysis and prediction.



## Future Directions

Aiming to incorporate more predictive factors to enhance its accuracy and utility



## Q&A

Comments & Questions?



# RESOURCES



- Kaggle. (n.d.). *New York housing market*. Retrieved from <https://www.kaggle.com/datasets/nelgiriyeewithana/new-york-housing-market/data>
- NYC Office of the Comptroller. (2024). *Spotlight: New York City's homeowner housing market*. Retrieved from <https://comptroller.nyc.gov/reports/spotlight-new-york-citys-homeowner-housing-market/>
- Investopedia. (2024). *Top U.S. housing market indicators*. Retrieved from <https://www.investopedia.com/articles/personal-finance/033015/top-us-housing-market-indicators.asp>
- Krish Naik. (n.d.). *Tutorial 7-Build, Train, Deploy Machine Learning Model AWS SageMaker- Predicting Test Data Endpoints*
- YouTube. Retrieved from [https://www.youtube.com/watch?v=XSsnPtHRZ6A&ab\\_channel=KrishNaik](https://www.youtube.com/watch?v=XSsnPtHRZ6A&ab_channel=KrishNaik)

