

This project uses the California Housing Prices dataset at <https://www.kaggle.com/camnugent/california-housing-prices>. The data contains information from the 1990 California census. It has an easily understandable list of variables, and has 20640 rows of data, which is manageable in size.

I have borrowed heavily from Aurélien Géron's recent book 'Hands-On Machine learning with Scikit-Learn and TensorFlow'.

Data Cleaning and Wrangling

There are 207 non-available values in the total-bedrooms column, and the correlation of total-bedrooms to medium_housing_value is very low, at 0.049686. Therefore, I have decided to drop total-bedrooms from the dataset.

The ocean_proximity column contains only string values. To make it easier to plot, I have replaced the string values by integers['ISLAND'=0, 'NEAR BAY'=1, 'NEAR OCEAN'=2, 'H OCEAN'=3, 'INLAND'=4].

Since the correlations of medium_housing_value to longitude(-0.0459), population(-0.0246), and households(0.065843) are also low, I have dropped these 3 columns as well.

The median_income column is measured in tens of thousands of US Dollars. I have converted it to US Dollars.

Different Models

Using Statsmodel to fit Linear Regression is very easy and straightforward. Sklearn, on the other hand, produces somewhat close results. The f-statistics are about 4586 and 4341, respectively.

The Decision Tree Regression is not suitable for this dataset, the f-statistic is 2776.

The Random Forest Regression, on the other hand, produces the best result. The f-statistic is 7650.

The SVR has f-statistic 1373.

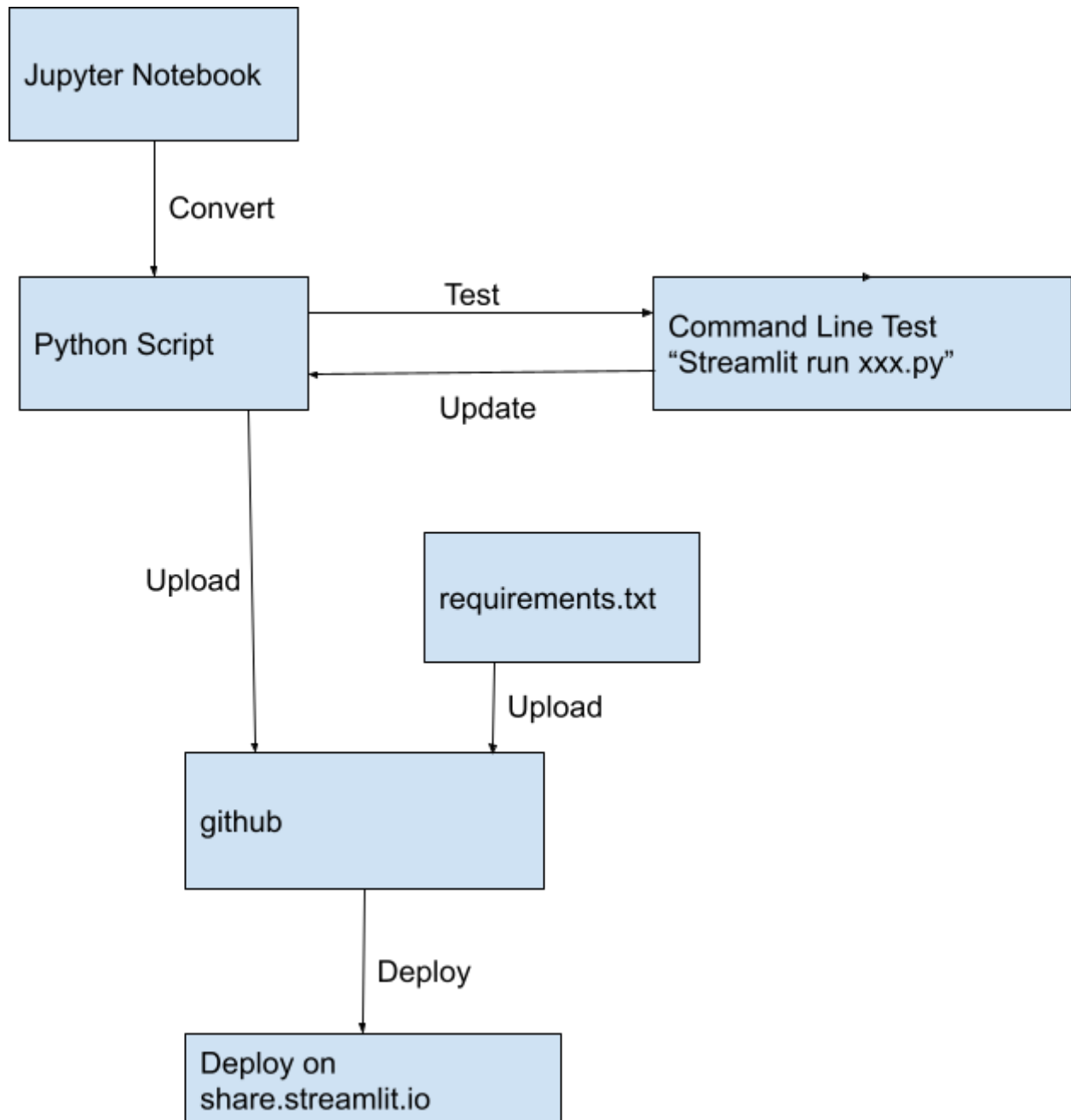
Model Fine Tuning

I have used GridSearchCV and RandomizedSearchCV to fine-tune the Random Forest Regression model. The f-statistics has reached 7723 and 7964, respectively, which are better than 7650 as before.

Deployment

I have decided to use streamlit to deploy my project, because it is the easiest in the market. I am aware that there are far more mature tools out there for machine learning deployment, I would like to investigate more once I have time.

Here is the diagram to illustrate the deployment steps:



Project Files

- Data file - housing.csv.zip
- Jupyter Notebook file - capstone.ipynb
- Jupyter Notebook file (using postgres database) - capstone_scaled2.ipynb
- Python script - housing.py
- Requirements.txt
- README

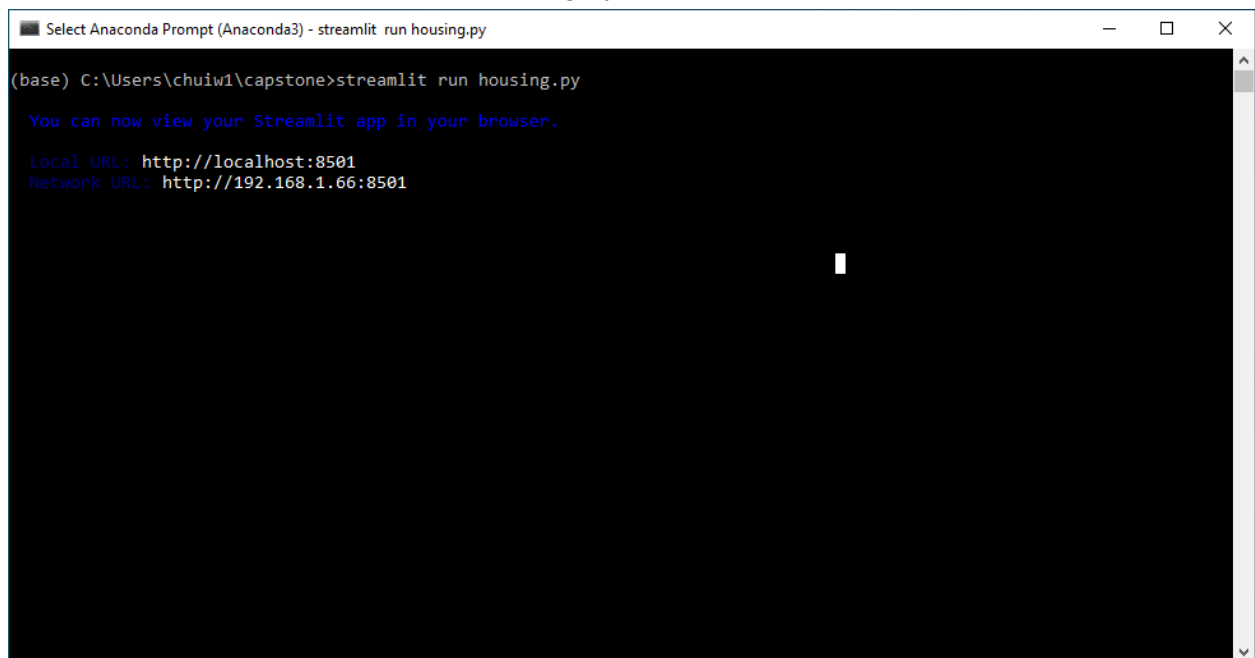
Deployed URL

<https://share.streamlit.io/tiffany88888/capstone/main/housing.py>

Screen Shots

To run as Web App Locally

- Use command “streamlit run housing.py”

A screenshot of an Anaconda Prompt terminal window. The title bar reads "Select Anaconda Prompt (Anaconda3) - streamlit run housing.py". The terminal content shows the command "(base) C:\Users\chuiw1\capstone>streamlit run housing.py" followed by the output: "You can now view your Streamlit app in your browser.", "Local URL: http://localhost:8501", and "Network URL: http://192.168.1.66:8501". A white cursor is visible on the line following the network URL.

```
Select Anaconda Prompt (Anaconda3) - streamlit run housing.py

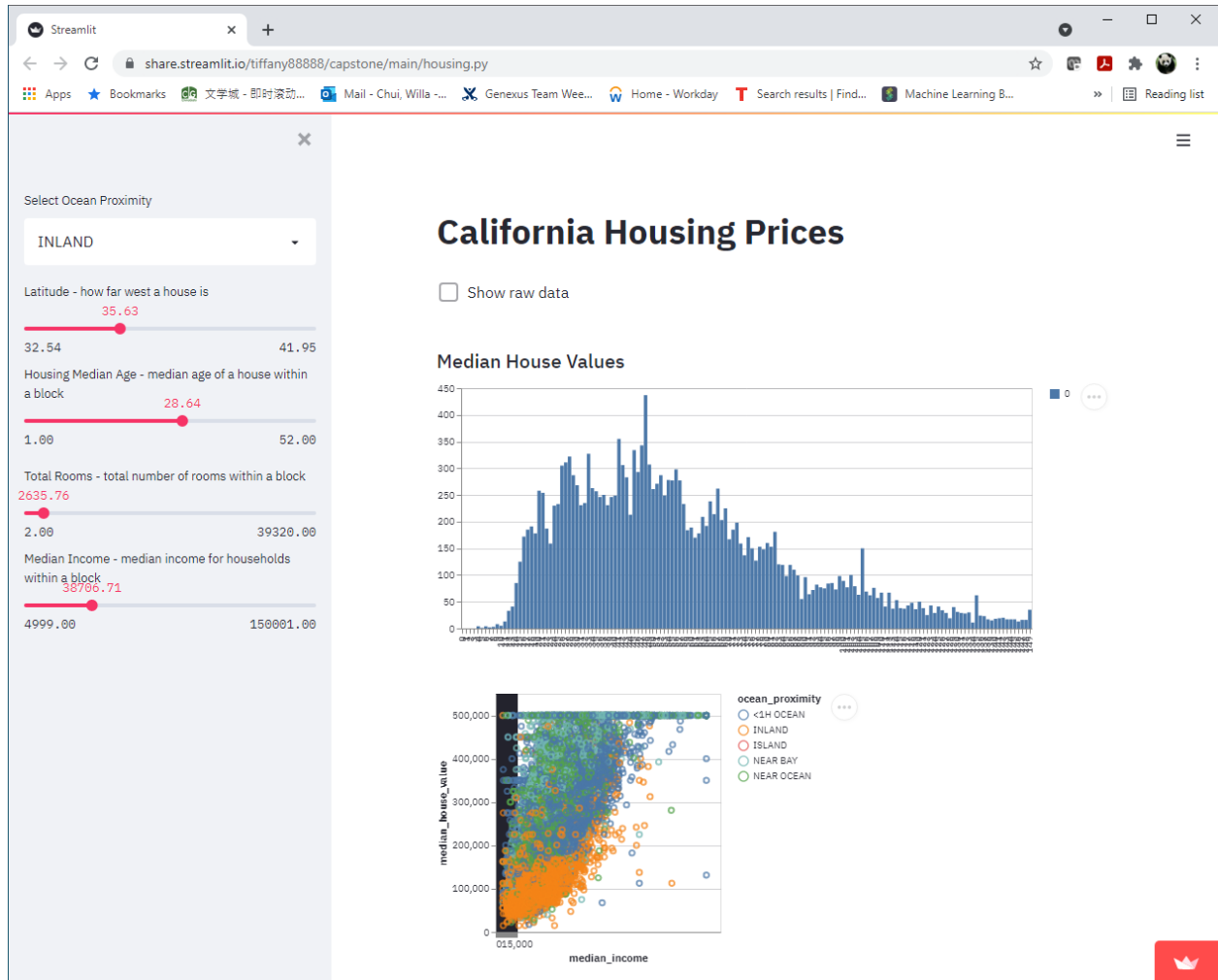
(base) C:\Users\chuiw1\capstone>streamlit run housing.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.66:8501
```

Web App - Page One

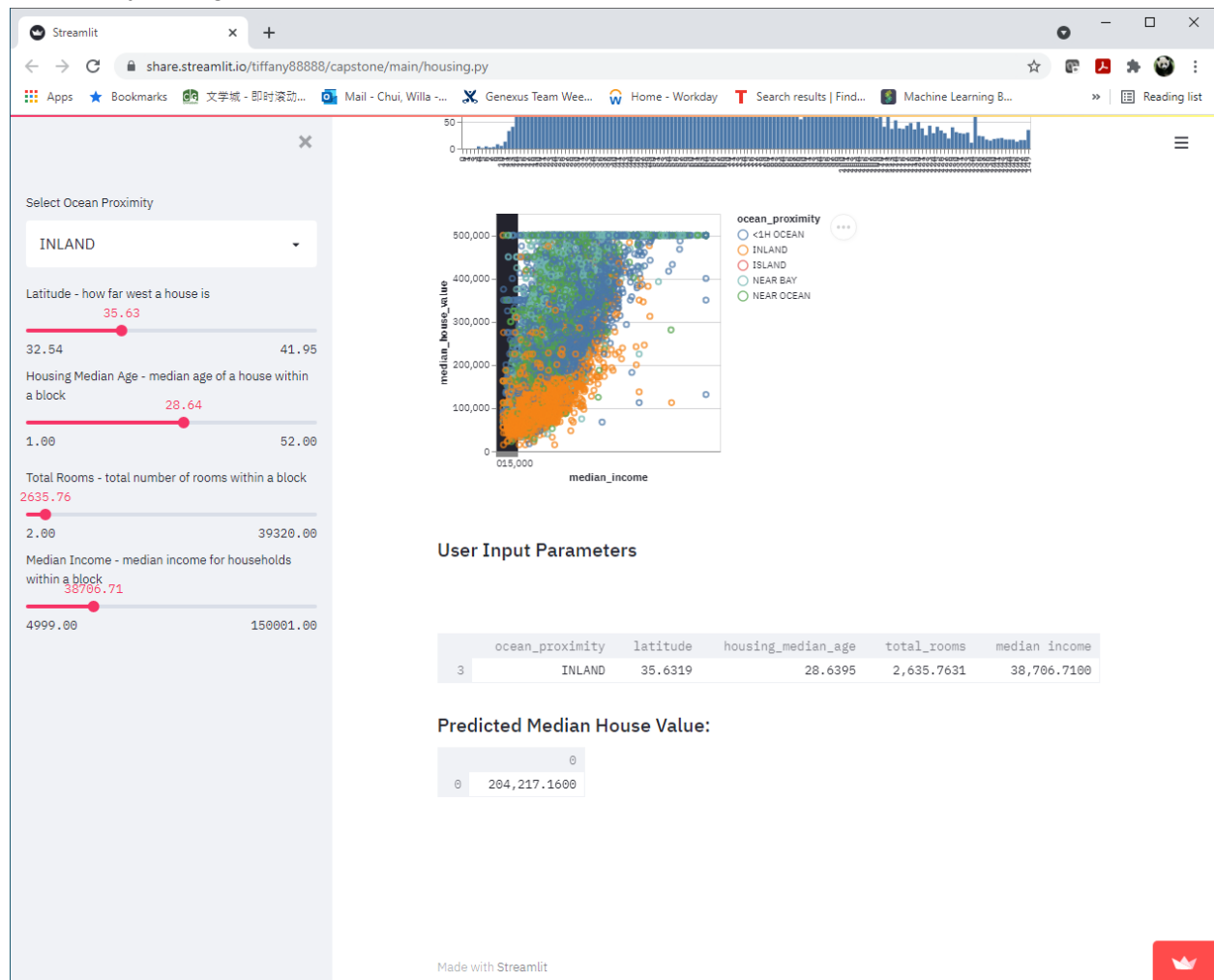
- User Input Panel on the left
- Median_House_Value Histogram on top
- Median_Income vs Median_House_Value Chart at the bottom



Web App - Page Two

- User Input Panel on the left
- Median_House_Value Histogram on top
- Median_Income vs Median_House_Value Chart in the middle
- User Input Parameter Summary and Predicted Median House Value at the bottom

Users may change the parameters on the left to obtain different predictions.



Web App - Page One (with Raw Data Listing)

