Time Series Analysis of Hotel Receipts in Texas

Sarthak Banerjee, Anthony LaConte, Francis Obeng, Aaditya Parameswaran, Om Patel

Problem Statement

What we have:

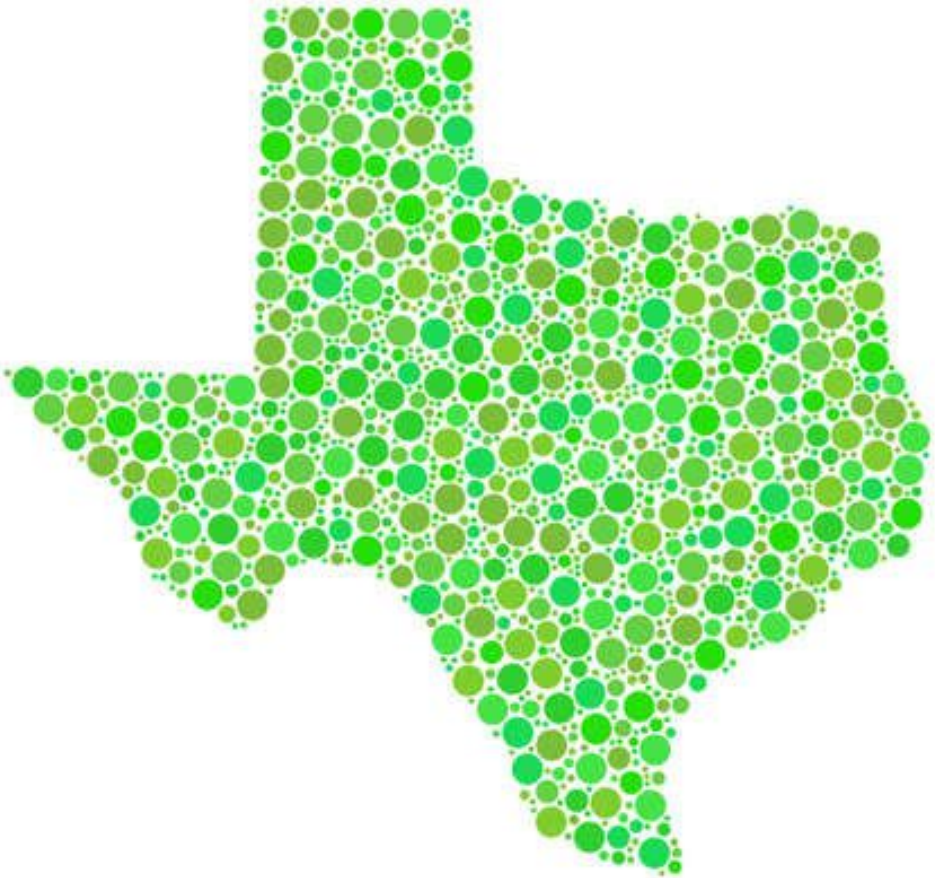
- Historical Data of Texas Hotel Taxes
  - Includes data such as Location Name (Hotel Name), Location County, Unit Capacity, Total Room Receipts, etc.,
  - Data from 2000 - 2022 (sans 2001)

- Historical Data of Texas Population from 1970 - 2023

Problem Statement

What we want to achieve:

- Time Series Analysis
  - Determine which locations are losing revenue annually, which locations are maintaining their revenue, and which locations are still growing
  - Incorporate Quarterly markers to quantify change between seasons



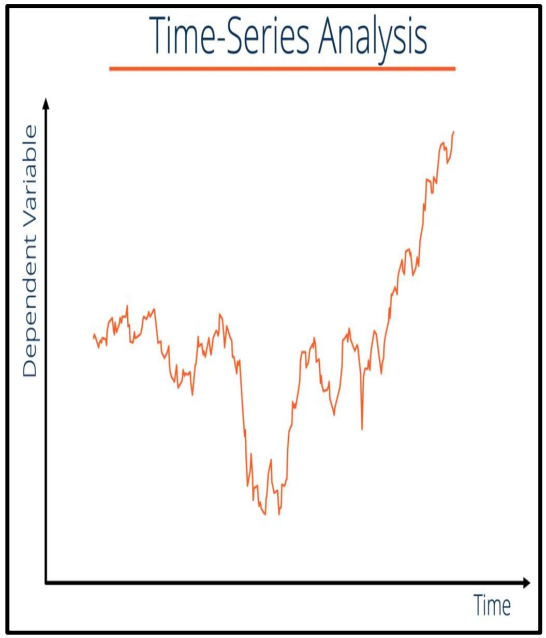
Use-Case Scenarios

- Research can be used to shut down locations that are projected to lose revenue
- New hotels can be built in locations that demonstrate a steady growth in revenue
- Model provides a method to estimate total taxable receipts, which can be used to allocate money across locations

AI Algorithm and Model(s)

Time-series analysis involves looking for:

- Trends, Seasonal patterns,
- Cycles and other characteristics in data that change over time.



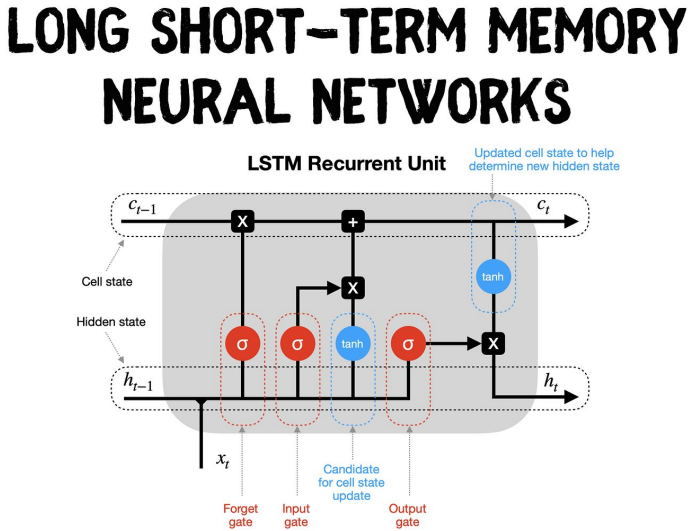
SARIMA MODEL :

- Seasonal Autoregressive Integrated Moving Average.
- The seasonal component, captures seasonality in time series data.
- Such as in our historical data which has varying seasons.

AI Algorithm and Model(s)

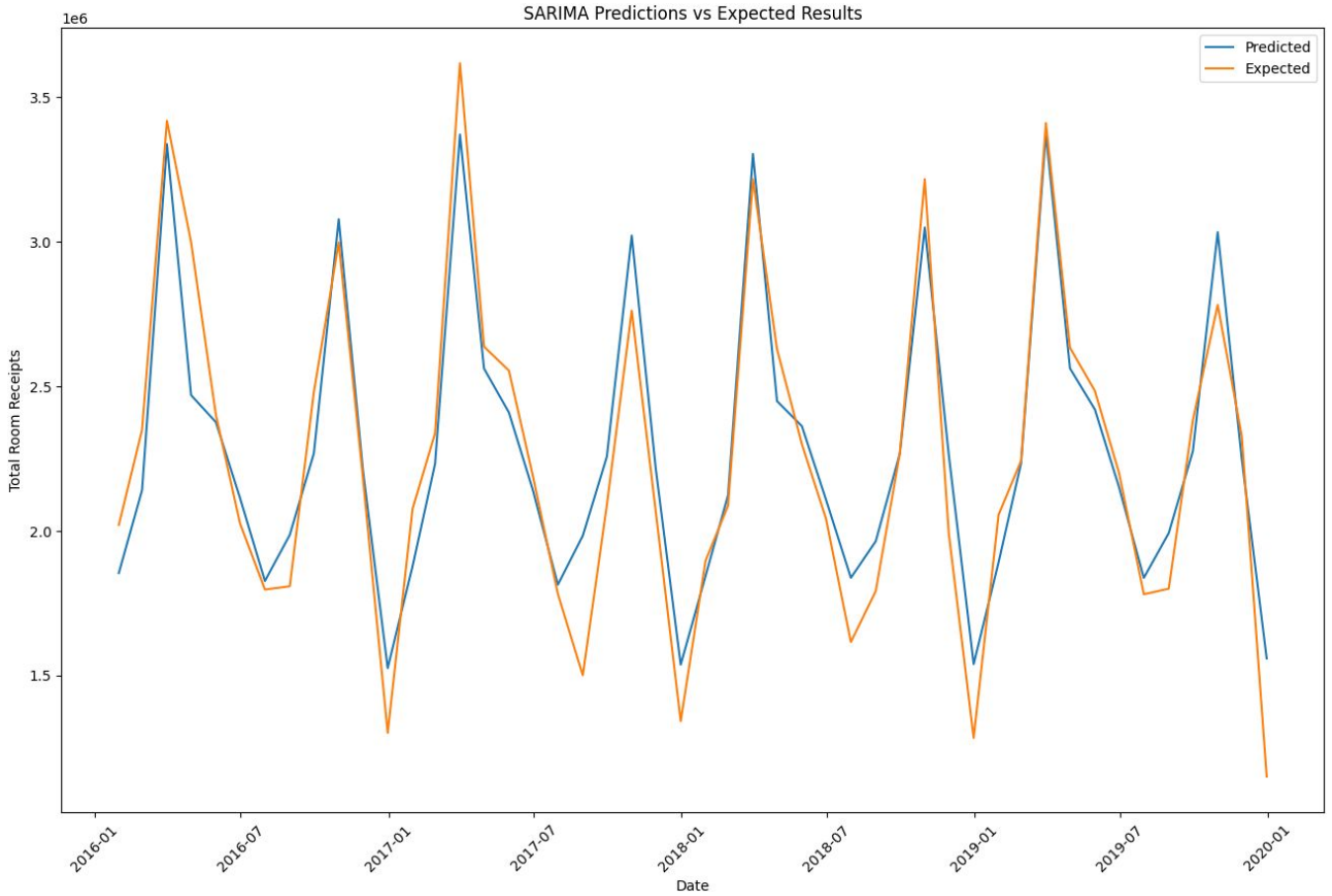
LSTM Neural Network

- Type of recurrent neural network (RNN) used in deep learning
- Designed to prevent the output from decaying or exploding as it cycles through feedback loops.
- Can learn order dependence in sequence prediction



Demonstration

Results: SARIMA



Results LSTM Neural Network

```
Epoch 1/10
60256/60256 [==============================] - 379s 6ms/step - loss: 0.0659 - val_loss: 0.1703
Epoch 2/10
60256/60256 [==============================] - 334s 6ms/step - loss: 0.0656 - val_loss: 0.1749
Epoch 3/10
60256/60256 [==============================] - 341s 6ms/step - loss: 0.0656 - val_loss: 0.1612
Epoch 4/10
60256/60256 [==============================] - 338s 6ms/step - loss: 0.0656 - val_loss: 0.1673
Epoch 5/10
60256/60256 [==============================] - 337s 6ms/step - loss: 0.0656 - val_loss: 0.1703
Epoch 6/10
60256/60256 [==============================] - 333s 6ms/step - loss: 0.0656 - val_loss: 0.1630
Epoch 7/10
60256/60256 [==============================] - 335s 6ms/step - loss: 0.0656 - val_loss: 0.1691
Epoch 8/10
60256/60256 [==============================] - 340s 6ms/step - loss: 0.0656 - val_loss: 0.1752
Epoch 9/10
60256/60256 [==============================] - 334s 6ms/step - loss: 0.0656 - val_loss: 0.1665
Epoch 10/10
60256/60256 [==============================] - 334s 6ms/step - loss: 0.0656 - val_loss: 0.1657
```

Lessons Learned

- Data preprocessing takes up a majority of the time when training a new model
  - Make sure the data has some form of relation between them
  - Clearing data of noise
  - Feature extraction and addition
- More number of epochs is not always better
  - Can lead to overfitting and decrease in accuracy
- Researching and using multiple models for one task
  - Even if a particular model feels like it best fits the data, the second or third best can also yield strong (if not stronger) results


Questions?