CS 4277 Spring 2024 Final Report

Kennesaw State University

College of Computing and Software Engineering

Department of Computer Science

CS 4277: Spring 2024

Deep Learning

Chen Zhao

Thomas Fitzgerald

tfitzge9@students.kennesaw.edu

April 28, 2024

Table of Contents

Abstract

This paper will be detailing the construction process of a deep learning model, designed to predict future apartment rental prices within Georgia. This model was created to aid the author with housing decisions by providing insight to price trends. The created model was originally going to be a convoluted neural network based on the findings from background research. Unresolvable errors in the construction process required the neural network to be replaced by a multilayered perceptron. The perceptron was constructed using two hidden layers with ReLu activation and a mean absolute error (MAE) loss function. The dataset used for the model was compiled from the website RentData.org and consists of the average rental prices in a county for different apartment types. Each entry in the dataset comprised the year, county, apartment type, and current rent price. The model was tested using a batch of size of 32 datapoints on 100 epochs and produced a final MAE of 46.17 and mean absolute percentage error (MAPE) of 5.46%. The results of this model were then compared to other attempts and models, revealing that the model performed better than previous research papers but only slightly worse than a Long-Short Term Memory (LSTM) model.

Introduction

A rental lease is a contract agreed upon between two parties where one party (lessee) will pay the other (lessor) for the right to live on a chosen property. Many companies have built a business around facilitating apartments for individuals to rent out of. Often these companies, about halfway through the lease, will present the lessee with an option to renew the lease or not. When this occurs, the lessee typically only has a few weeks to decide. Such an arrangement puts the lessee at a disadvantage for numerous reasons.

First, offers with a move in date several months out are nearly impossible to find. This means without a guarantee offer, whatever rental prices are presented may not reflect actual rates at the end of the current lease. The real estate market is volatile and subject to change based on numerous outside factors. One example would be in the year 2020 an outbreak of COVID-19 negatively impacted the real estate market causing the median asking price for vacant rental units to jump from $1,041 to $1,190 [12] [13]. The asking rent has continued to increase, making it more difficult for lower income individuals to find housing.

Second, when a lessor offers this decision, they may also offer varying month-long lease plans. Such plans are usually set at a significantly higher rental price than the standard one-year lease. This causes the cheapest option for the lessee to typically be the yearlong lease. If something were to occur that requires the lessee to move out prior to the end of the lease, they would have to pay a significant lump sum as violation for breaking the lease. This action forces the lessee into continuing the lease for a longer term and could potentially miss out on opportunities for cheaper housing.

This paper's researcher addressed a similar situation as described above. In the researcher's case, it was decided to renew the lease with the existing leaser. What largely influenced this decision was the inability to find locations with a guaranteed move in date that would line up with the lease's termination. This meant that whatever offers were shown would not be guaranteed if the researcher decided to terminate the lease. Without this assurance the rental prices could increase further, making the researcher pay more than if they decided to continue the lease. This led to the motivation to develop a deep learning algorithm that would be able to predict future rental prices. Such a tool would be instrumental in helping to determine the optimal choice for the decision.

After numerous months of research and work, the author has produced a deep learning model that accurately predicts future rent trends. The deep learning model is a multilayered perceptron, a feed-forward neural network that uses fully connected layers and activation functions to learn. More information will be revealed about the model in the next sections, but the model learns from the previous annual rent prices to predict the future prices. This model serves as an easy and efficient tool, that the author will use to help him decide to maintain his apartment lease or find a different one.

Related Work

Prior to carrying out this project, research was conducted into identifying what other models and attempts were made at predicting future rent prices. This research was conducted through identifying and reading published research papers, and some of these papers will be discussed in this section.

One of the first papers that was discovered during research was an article published in 2019 called “House Price Prediction Approach based on Deep Learning and ARIMA Model” [9]. In the paper, the authors created an ARIMA model and used the python framework Scr1apy to collect their dataset. The authors extracted their data from the internet using scrapy, then processed the data to convert it into a compatible format with the autoregression integrated moving average (ARIMA) model. The ARIMA works to predict future values based on the past values given to it. The ARIMA model is a combination of three other models to reduce prediction errors. This is done through a series of steps to convert the dataset into a stationary series that can be fed into the three sub models to make predictions. In the research paper, after the ARIMA model was trained it was able predict future house prices with basic consistency. The ARIMA model was tested against a support vector machine, with the results showing the ARIMA model performing better with a mean relative error of 0.22 when compared to the support vector machine’s relative error of 0.28.

While the ARIMA model shows potential in further application of rent predictions, this project is unable to use it. The ARIMA model is not a deep learning algorithm, which is a requirement for this project. This paper introduced the support vector machine algorithm which could be a potential deep learning algorithm for this project.

Another research paper called “Predicting property prices with machine learning algorithms” [8], sought to investigate the potential of machine learning algorithms in the role of predicting housing prices. The paper’s researchers utilized three machine learning algorithms: a support vector machine (SVM), random forest (RF), and a gradient boosting machine (GBM). The researchers utilized three performance metrics to tell the accuracy of the algorithms and decide which one was the most efficient. The experiment consisted of feeding a dataset of 40,000 housing transactions (over the span of 18 years) into the algorithms and measuring the mean squared error, root mean squared error, and mean absolute percentage error. The results of this experiment found that the GBM and RF algorithms were the most accurate.

The researchers then went on to discuss how their findings compared to other research papers. From this comparison, the researchers derived that no one algorithm can be considered the most efficient as there are different criteria depending on the features selected and the dataset present. While the researchers do discuss important points about different criteria affecting the efficiency of a model, the research paper fails to give recommendations on controlling the criteria to improve accuracy. The paper focused too much on which algorithm would be the most efficient, that its conclusion was irrelevant to what they researched. To prevent this error from occurring in the current project, the author will focus more on the results of the experiment rather than the authenticity of it. Additionally, the related research paper failed to offer meaningful insight into which type of algorithm would be best suited but did offer an important consideration into what dataset features are being measured.

A third research paper that was discovered was called “Enhancing Real Estate Market Insights through Machine Learning: Predicting Property Prices with Advanced Data Analytics” [7]. This paper conducted its investigation on predicting house prices by primarily identifying which features in a dataset played the greatest role in determining future prices. Before the dataset was processed for the model, an exploratory data analysis was conducted. The analysis was conducted to better understand the dataset features and identify relationships between the features. The results of this analysis were presented in various charts, but never discussed throughout the rest of the paper.

Much like the previously analyzed one, this research paper also utilizes multiple deep learning algorithms. More specifically, this paper employs multiple linear regression (MLR), Support Vector Regression (SVT), Decisions Tree (DT), Random Forest (RF), and XGBoost. After the datasets were preprocessed and the models trained, the accuracy of the models were measured using both the mean absolute error and mean squared error. The results of these accuracy metrics indicated that the random forest model was the most accurate.

This paper struggles heavily with introducing too many elements to focus on in the paper. The researchers start out by discussing in the introduction how their purpose is to identify which characteristics have the greatest impact in accurately predicting prices. This would be a good premise for the research paper to focus on, but then it shifts focus to comparing the accuracy of five different deep learning algorithms. The conclusion section of the paper doesn't discuss the dataset features and only discusses the algorithms. To prevent the current project from repeating the same error, the project will utilize only one deep learning algorithm and a set of features.

A different research paper that was chosen as a point of reference was called "Modeling Housing Rent in the Atlanta Metropolitan Area Using Textual Information and Deep Learning" [1]. This research paper compared the results of three categories of deep learning algorithms to determine the most efficient way to predict housing prices.

The first algorithm category measured the spatial interpolation of the dataset. The second type of algorithms were a set of models that measured the textual information of the dataset. The last category utilized a combination of measuring the spatial and textual information of the dataset. After experimenting with a dataset produced from craigslist, the researchers found that a bagged decision tree performed the best with a mean absolute percent error of 11.7%. The bagged decision tree was an algorithm that measured both the spatial and textual information and works by sampling the training data with replacement and creating a model for each sample. The results of these models are combined and averaged for predictions.

This research paper's goal was to find the most accurate model for predicting future housing rent. Despite the bagged decision tree algorithm performing well, the accuracy of the dataset is called into question. The researchers utilized a dataset created from craigslist that was heavily cleansed. Regardless posts on craigslist are user generated and may not reflect actual housing market trends. To prevent this error from occurring in this project, the dataset will be sourced from a government authenticated source.

Another research paper, "Prediction of House Price Based on The Back Propagation Neural Network in The Keras Deep Learning Framework" [3], constructed a feed forward neural network to predict future pricing on secondhand housing in Shanghai. The dataset was constructed through using BeautifulSoup parser to gather information from the home link network in Shanghai. The researchers then performed various methods to find the optimal number of hidden layers and size of learning rate. Once these values were calculated, the model was trained using dropout to prevent overfitting. Through experimentation, the researchers were able to produce a model with an accuracy score of 94.59%, higher than their previously researched works.

This research paper gave amazing tips for how to make a more accurate deep learning algorithm. Three formulas were provided to find the most optimal number of hidden layers. These formulas will be used in the project to identify the optimal number of hidden layers, ensuring greater accuracy with the project's algorithm.

Unlike the previous research papers, the next analyzed research paper, "Apartment Rent Prediction Using Spatial Modeling" [5] focused exclusively on measuring the spatial relationship between pairs of datapoints. The researchers utilized a dataset provided by the REIS Inc (real estate information provider) which consisted of properties from eight different cities. The researchers then used a Bayesian model with a specified spatial equation to generate a visual map of the predicted changes. Using this model, the researchers were able to generate a conceptual map of predicted rent values. The researchers also noted that there was significant difference in the spatial ranges between different markets.

This paper did well at detailing the processes used and presenting their findings, but the paper did not utilize a deep learning algorithm. It is important to note that the paper's findings on the relationship between spatial distance and rent prices are only unique for the city measured in. This likely indicates there is not a strong relationship between the two variables, meaning this project will not consider such variables when predicting apartment rent.

The next research paper, “Spatial prediction of apartment rent using regression based and machine learning-based approaches with a large dataset” [4], sought to enhance existing arguments on the utilization of either regression or machine learning models for rent prediction. What this paper does different than previously referenced research, is apply their experiment to a substantially large dataset, as high as 10^6 samples. For regression side of the comparison, the researchers utilized variations of the nearest neighbor Gaussian processes (NNGP) model. For the machine learning models, the researchers utilized XGBoost, random forest, and a deep neural network. Four tests were conducted utilizing samples size of 10^4, 10^5, and 10^6. From the results of the experiment, the researchers found that the machine learning models outperformed the regression models for every test regardless of sample size. The XGBoost specifically performed the best out of all the models present.

This paper presented good findings on how the regression and machine learning models compare to each other. Where the other research papers became overwhelmed with different variables, this paper successfully presented and discussed all its points. While the paper did find that an XGBoost model performed the best, this project will not be able to use said model. This is due to the XGBoost model not being a deep learning algorithm, disqualifying it from consideration.

The next research paper, “Better Accuracy for House or Apartment Rent Prediction in Metro Cities Using CNN Compared to Support Vector Machine” [6], was another comparison between the use of convolutional neural networks (CNN) vs support vector machines. The dataset that was used for this research paper was procured through Kaggle and was composed of the prices and features of over 1300 houses. Ten tests were performed using samples of various sizes. The results of the experiment ended with the CNN having an average accuracy of 94.10% while the support vector machine had an average accuracy of 90.60.

This was a good research paper that focused exclusively on the performance of one type of deep learning algorithm. The CNN performed extremely well in this research paper when compared to the previous ones that utilized the same algorithm. The researchers do not detail how their neural network was constructed, so it is difficult to tell what caused this algorithm to perform better. Due to the results of the experiment, a CNN should be considered for the experiment.

The final research paper, “Exploring a Pricing Model for Urban Rental Houses from a Geographical Perspective” [2], noted the shortcomings of both regression models and typical deep learning models and decided to create a hybrid algorithm of the two. This new model was a fully connected neural network with geographically weighted regression that heavily relied on location and nearby places of interest. The dataset consisted of locations in four major Chinese cities, with a datapoint comprising attributes about the location and nearby commodities (food, recreation, shopping, etc.). The paper’s experiment compared the results of the proposed model to three other standard models. The results of the experiment showed the proposed model having the lowest mean absolute error of 6.07%.

None of the other research papers considered the proximity of nearby places of interest when calculating changes in rent. This showed an interesting aspect of rent predictions that are difficult to emulate. Obtaining such information in bulk for a dataset is a difficult task that would be extremely time consuming. Additionally, the hybrid algorithm that was created for the research paper may be more complicated than necessary. While the algorithm had the lowest mean absolute error, it was only by 0.14% when compared to the CNN. In practical applications, a simpler algorithm might be the preferable option when it loses less than one percent of accuracy.

Methodology

To organize time and efforts most efficiently, this project was divided into three major components, each to be completed at a corresponding milestone. These components consist of completing background research, creating the deep learning model, and creating the final report. The methodologies section of this report will go into detail explaining the tasks involved in each component, as well as any obstacles that occurred. For a visual representation of the time spent on each component and its tasks, a Gantt Chart is shown in Appendix A.

Component 1: Background Research

While a project begins with an idea of how to solve a problem, background research formalizes that idea into actualized steps to create the solution. The background research conducted in this project was crucial for determining the model and dataset to be used. This component encompassed the time spent reading related articles; researching how to build the chosen deep learning model; and exploring the different possible databases to be used.

Some of the results of the background research are shown in the previous section of Related Works. The time spent reading related research papers was monumental in shaping how this project would be conducted. For example, the related research papers presented their findings on the most efficient model for predicting rent prices. This gave a list of models to then research in deciding whether they would be ideal for this project.

Researching the list of possible models comprised of identifying how a model would be built and how it processes a datapoint. One example of the results from this research was the information learned about how to apply CNN to make predictions. Previous experience with CNN revolved around using models for classification purposes not predictions. Research helped to find examples showing that a single neuron output with a linear activation function would allow the CNN to make predictions.

During the background research several databases were discovered that could be used in the project. Considerable time was spent authenticating each database due to the issues with one of the related research papers [1]. As mentioned before, this research paper used a database constructed from Craig's List home listings, which is severely unreliable. Not wishing to repeat this error, this project determined that the selected database needed to be government authenticated for greater accuracy. More details about the chosen database will be discussed in the next section.

Component 2: Deep Learning Model Creation

The second component of the project revolved around the creation and optimization of the deep learning model. This component of the project was the largest and hardest component to achieve due to unforeseen complications that arose when trying to create the deep learning model.

Based on the findings from background research, this project initially decided to create a convoluted neural network (CNN) for predicting rent prices. During the construction of the model, type errors kept occurring that could not be resolved and were difficult to track the cause of. After several days it was identified that the reason for these errors was because the model was not processing an image. Due to the nature of the data being processed, this error could not be resolved so a customized multilayer perceptron was created. This new model was able to process the dataset and function without errors. The perceptron consisted of three types of layers (input, hidden, and output) with different hyperparameters to adjust the performance. The input layer of the model utilized Relu activation with four neurons. Four neurons were selected for the input layer due to background research suggesting to use a number of neurons equal to the number of features.

Creating the model's hidden layers required defining several hyperparameters. The first hyperparameter was the number of hidden layers to be used. This value was calculated by using the equation ($n_i = \log_2 n$) that was mentioned in the background research [3]. In the equation, n represents the number of input features, while n_i represents the number of hidden layers, calculated to be two. Thus, two hidden layers were created that utilize the Relu activation function and three neurons. Each hidden layer was followed by a Normalization layer. The summary of the model is shown below in Figure 1, while the architecture is shown in Figure 2. Once the model was created and tested for errors, it was then trained under the hyper parameters of 32 batches over 100 epochs.

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_27 (Dense)            (None, 4)                 20

 dense_28 (Dense)            (None, 3)                 15

 normalization_15 (Normaliz  (None, 3)                 7
 ation)

 dense_29 (Dense)            (None, 3)                 12

 normalization_16 (Normaliz  (None, 3)                 7
 ation)

 dense_30 (Dense)            (None, 1)                 4

=================================================================
Total params: 65 (268.00 Byte)
Trainable params: 51 (204.00 Byte)
Non-trainable params: 14 (64.00 Byte)
_________________________________________________________________
```

Figure 1: Summary of Multilayer Perceptron

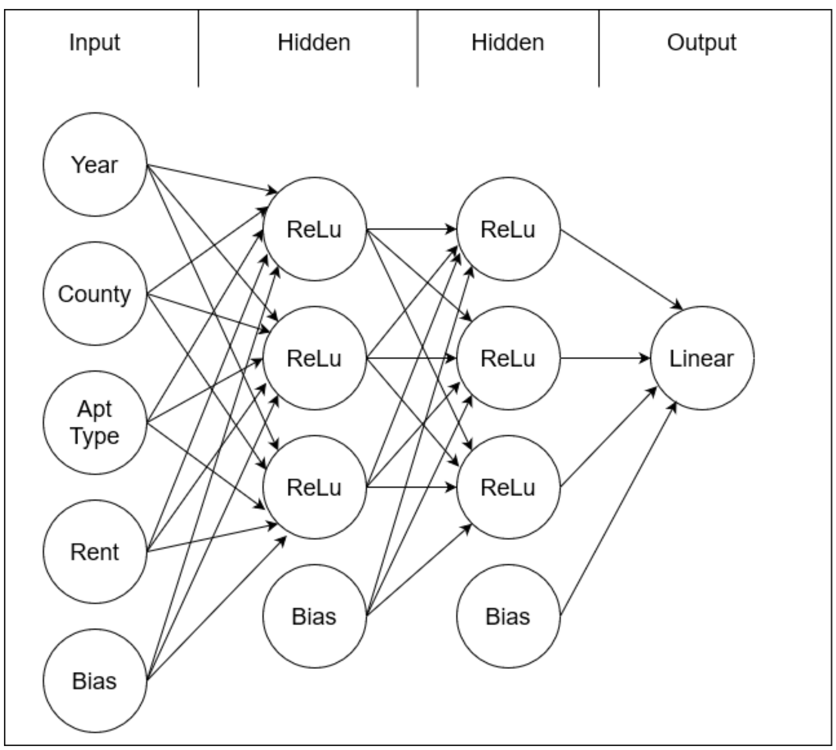


Figure 2: Architecture of Multilayer Perceptron

Component 3: Final Report

The third and final component of the project was the preparation and writing of the final report. This component of the project featured the further optimization of the multilayered perceptron and comparing its results with other models and datasets to determine its effectiveness.

It is not enough for the project to just make a prediction of future rent prices but to also compare those predictions to other attempts to better understand its accuracy. To fulfill this requirement, the project's results were compared to one of the related research papers [1] due the shared geography. Additionally, a second model was created so that the effectiveness of the model could be compared. This new model was Long-Short Term Memory (LSTM) model. This model is comprised of four LSTM layers each with fifty neurons. After each LSTM layer, a normalization layer is inserted to standardize the layer's output, the structure of the model is shown below in Figure 3. The LSTM layers are mandatory with an LSTM model because they allow the model to learn and remember long term dependencies. Once the LSTM model was model was built it was then trained under the same hyper parameters of 32 batches over 100 epochs.

```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_8 (LSTM)                (None, 3, 50)             10400

normalization_16 (Normaliz  (None, 3, 50)             101
ation)

lstm_9 (LSTM)                (None, 3, 50)             20200

normalization_17 (Normaliz  (None, 3, 50)             101
ation)

lstm_10 (LSTM)               (None, 3, 50)             20200

normalization_18 (Normaliz  (None, 3, 50)             101
ation)

lstm_11 (LSTM)               (None, 3, 50)             20200

normalization_19 (Normaliz  (None, 3, 50)             101
ation)

dense_18 (Dense)             (None, 3, 1)              51

=================================================================
Total params: 71455 (279.14 KB)
Trainable params: 71051 (277.54 KB)
Non-trainable params: 404 (1.59 KB)
_________________________________________________________________
```

Figure 3: Summary of LSTM Model

Experimental Setup

Dataset Description

During the beginning phases of this project two separate databases were being considered for use in the project. The first dataset was produced by the Research & Analytics Group at the Atlanta Regional Commission using information published by the US Census Bureau. This dataset was incredibly detailed providing the exact number of apartments available for a rent range. This data set was originally going to be selected for this project, but the Research & Analytics Group only posted information for the years 2019, 2020, and 2021. This data set would not be broad enough to accurately capture the trends of rent changes, thus making it unfit for the project.

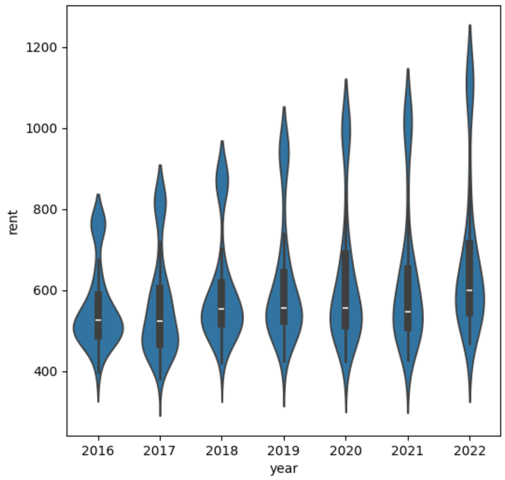
The second dataset, which was selected for the project, was created by compiling the average fair market rent prices posted on the website RentDate.org, an independent organization that is not affiliated with any governmental agency. This website produces interactive maps and rent market analysis for public use based on the Fair Market Rent rate published by the Department of Housing and Urban Development. The Fair Market Rent rate is calculated through a nationwide survey and is used by landlords for deciding rent and government programs for determining housing assistance amounts.

The fair market rents presented on the website were divided based on counties. Each county had a separate rate based on the type of housing and county. These rates were compiled from the years 2016 – 2023 and organized into an excel spreadsheet. The format for an entry is shown below in Figure 4.

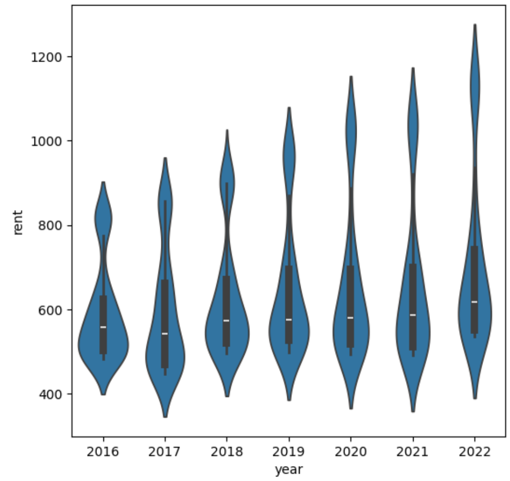
year	county	studio	1-bedroom	2-bedroom	3-bedroom	4-bedroom
2023	Appling	602	606	772	1097	1123

Figure 4: Example of Excel Data Entry

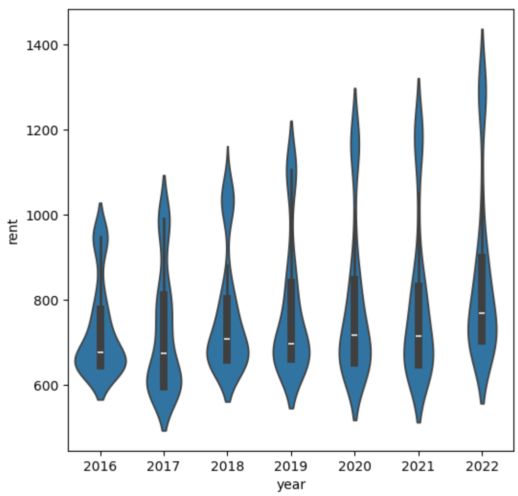
There are a total of one hundred fifty-nine counties in Georgia. If the average rent prices for five different types of apartments were recorded over the span of eight years, that would result in a database of 5,565 datapoints. Each datapoint would consist of the year, county, apartment type, and rent price. The distributions of these datapoints are shown in violin graphs below and are grouped by the type of apartment. From observing changes in distribution as the years change, it can be noticed that the maximum rent prices steadily increase each year. Another observation from these graphs is that the average rent price drops in the year 2017 but then begins to steadily increase.



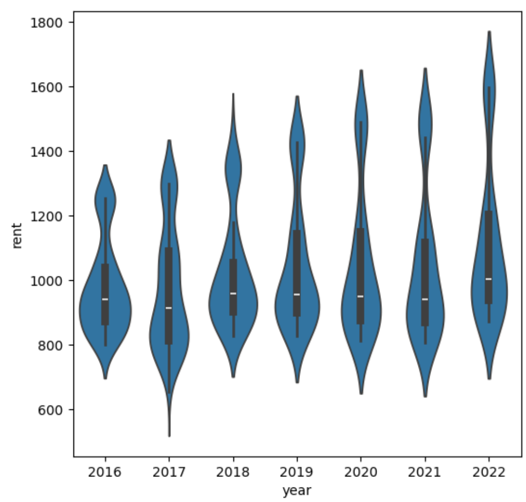
Distribution of Studio Apartments



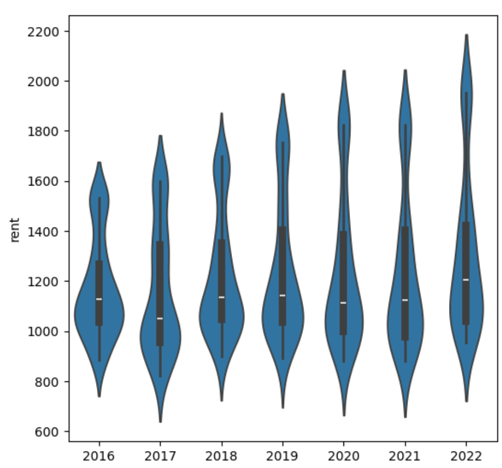
Distribution of 1-Bedroom Apartments



Distribution of 2-Bedroom Apartments



Distribution of 3-Bedroom Apartments



Distribution of 4-Bedroom Apartments

An important argument that needs to be addressed is whether the dataset is viable because of its small input. When an entry is used, it only utilizes four inputs regarding the county, year, apartment type, and current rent. While it may be perceived as being insufficient, it is important to realize these variables are not being used to predict the rent price. This project is seeking to predict rent prices based on previous trends. Due to this, the model is training on the changes in rent prices for different instances. While a larger number of input variables could possibly help improve the accuracy of the model, it is not necessary. What is critical for training the model is including as changes to the rent as possible.

Evaluation Metrics

Deep learning models learn and improve by using an accuracy metric. These models are programmed to minimize the metric and will make decisions to achieve this goal. Due to the nature of the project's deep learning model predicting a numerical value, a regression-based learning metric would have to be used. One such regression metric is the mean absolute error (MAE). The MAE measures the difference between a predicted value and an actual value. This difference is always positive, so larger or smaller predictions don't negatively affect the metric. The error then computes the average of these differences producing the mean absoluter error.

The MAE is the best metric to use for the model due to its resistance to outliers. The violin graphs from the previous section indicate extreme values of rent prices. These values would skew a model when training from it, but the values cannot be removed due to their representation of actual values. Thus, the mean absolute error helps to ensure it still affects the model but does not skew it.

A second accuracy metric was also chosen to help analyze the perceptron's results. This metric was the Mean Absolute Percentage Error (MAPE) which is the percentage equivalent of the MAE. This accuracy metric helps to give a better understanding of how well the results perform by providing the percentage difference rather than a numerical difference.

Implementation Details

As stated, this project utilizes a multilayered perceptron to predict future rent prices. The perceptron was coded in python through Google Collab and utilizes the Pandas and Keras APIs.

The datasets, described previously, were saved as CSV files, one for each year, and imported into the python program. The program then populated Pandas' data frames, converting the counties and apartment type into numerical values. This allowed for a single data frame to contain datapoints for different types of apartments, allowing the model to train without separating the data. The data frames were then split into 75% training data and 25% validation data. The prediction for the year 2023 would serve as the final test once the data was trained and validated.

Results

Once the deep learning model was constructed it subsequently trained using the dataset from 2016-2022. The training consisted of utilizing batches of 32 samples with 100 epochs. This caused the MAE to start around $900 but quickly converged to $45 to $46. The MAPE also underwent similar changes, starting with a percentage error of 97.12% but quickly dropping to the absolute minima around 5%. The final MAE of the validation data was $46.13 and the MAPE was 5.32%. The subsequent figures below show the change in the error rate as the model was being trained.

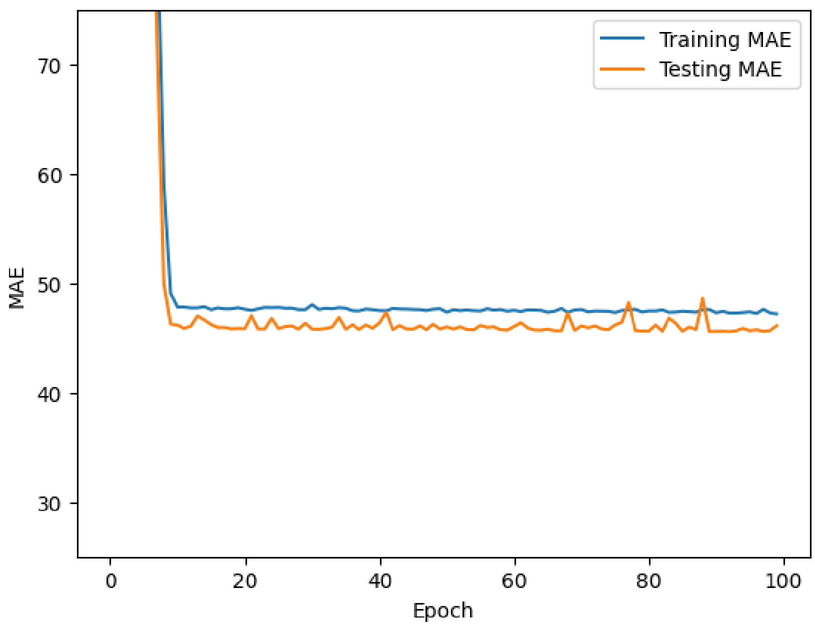


Figure 5: Change in Mean Absolute Error

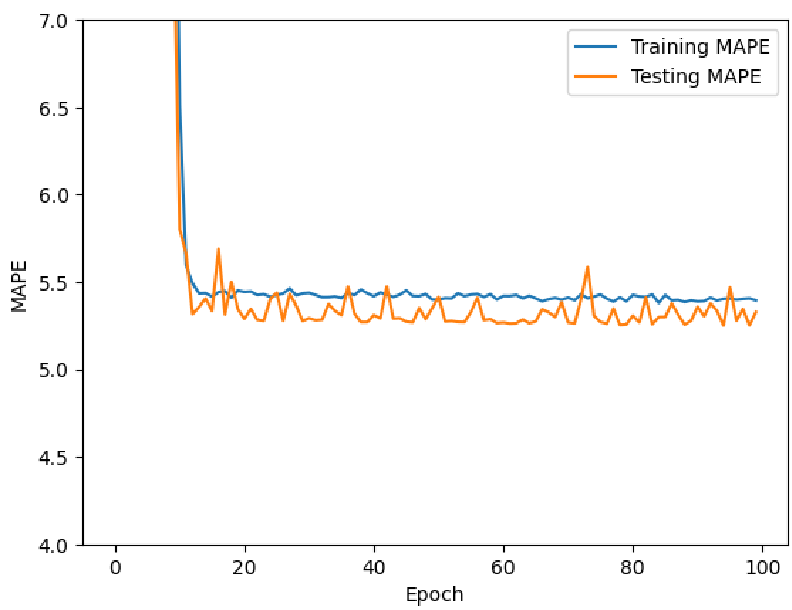
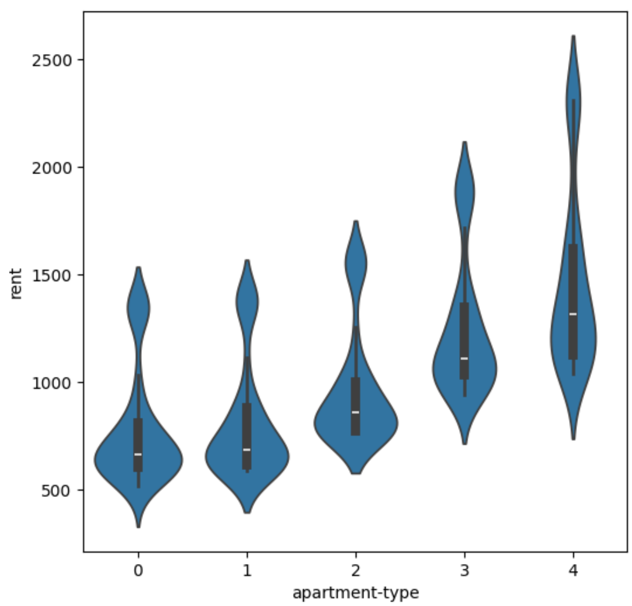
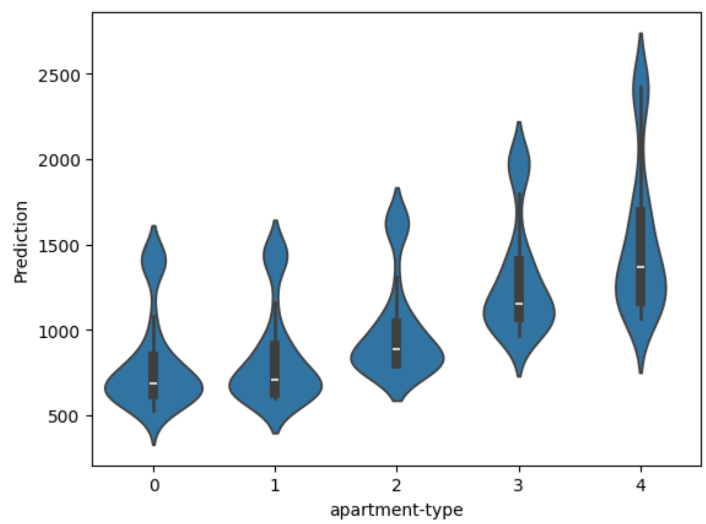


Figure 6: Change in Mean Absolute Percentage Error

Once the model was trained it was then used to predict the future rent prices for the counties in Georgia. The results of this prediction are contained in Appendix B, though a violin graph of predictions and 2023 prices are shown in the figures below. Comparing the violin graphs reveals that there is no significant increase in the rent prices for 2024. There is still an increase in the average rent, but the average rate of change is around $41.



Violin Graph of 2023 Rent Prices



Violin Graph of 2024 Rent Predictions

Comparison

While the perceptron model performs well in this specific test, it is important to compare this project's model to other attempts to identify how effective it is. One of the research papers [1] in the related work section utilized three different deep learning models (LSTM, CNN, and LSA) to predict future rent prices in the Atlanta area. These models were provided textual information as the inputs for their predictions. The performance of the models in this paper are shown below in Figure 7.

	MAE	MAPE (%)	RMSE
LSTM	196.760	15.452	288.370
CNN	208.886	17.030	300.103
LSA	211.701	15.655	311.688

Figure 7: Performance of Research Paper [1] Models

When analyzing the performance of the research paper's models it is important to note that their lowest MAPE was 15.45% for the LSTM model. This percentage error is significantly higher than the perceptron indicating that this project's model is more efficient.

Aside from the previously mentioned research paper, the models used in the other related papers provide little detail about the construction of their models. Additionally, the databases used by these papers were often only accessible to their college or required secondary software to recreate their datasets. Due to these obstacles, it was unfeasible to apply any other related models to this project's dataset or to apply the perceptron to their databases.

To compensate, an LSTM model was created for comparison with the project's perceptron. For this model to work, the dataset was subsequently modified so that a data entry included the average rent prices of the past three years. This resulted in 3,180 datapoints being created with twenty five percent of them serving as validation data. After the model was tested for errors, it was trained and validated, producing a final MAE of $35.31 and MAPE of 4.04%.

When comparing the results of the LSTM model to the perceptron, it is clear to see that the LSTM model performed better. The mean absoluter error of the LSTM model was lower by about $10, and the percentage error was smaller by about 1%. The subsequent figures compare the changes in MAE and MAPE for the perceptron and LSTM models.

While one might quickly write off the perceptron model as being inefficient, the difference between the results is extremely small. Having such a small difference of percentage error indicates the models are comparable in performance, especially when considering the amount of time it takes to train the models. Due to the number of parameters in the models, the perceptron only took 83 seconds to train while the LSTM model took 129 seconds.

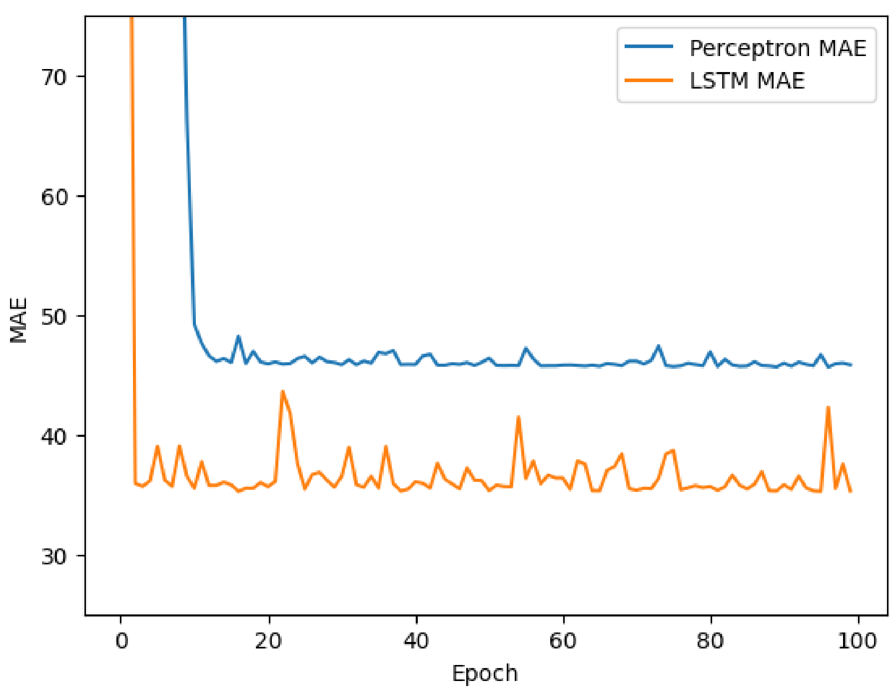


Figure 8: Comparison of MAE between the models

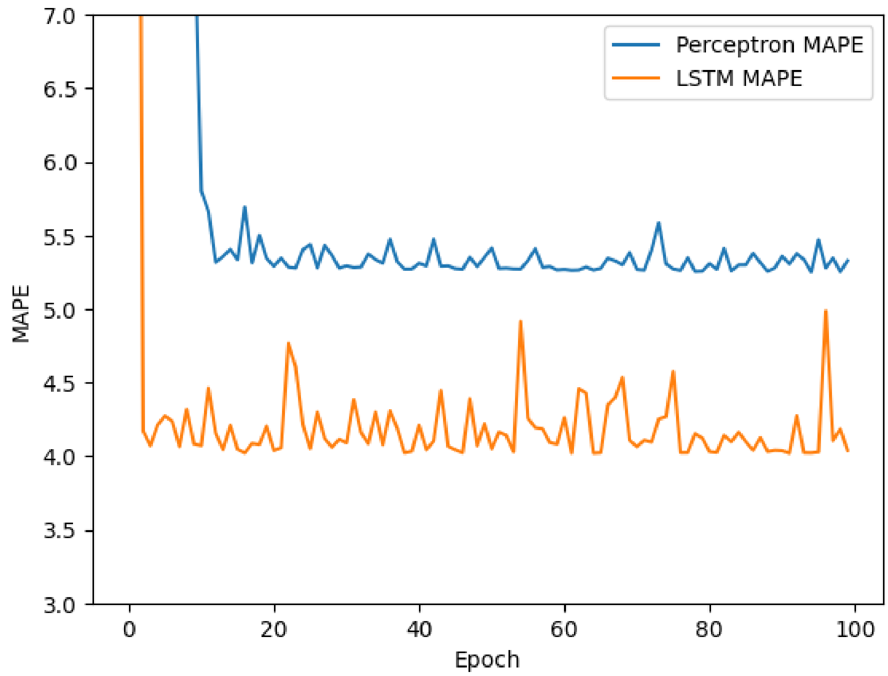


Figure 9: Comparison of MAPE between the models

Conclusion

This project sought to create a deep learning model that would learn from prior rent prices and make predictions on their future value. This model's creation was important due to the author's struggle with changing rent prices. Research was conducted into similar attempts for achieving the same goal, revealing different methods of predicting rent prices. Aspects from these prior attempts were taken into consideration and greatly shaped how the deep learning model was created.

While this project initially tried to create a convoluted neural network, it instead successfully created a multilayered perceptron. This project created a dataset from the independent website Rentdata.org, with an entry including the year, county, type of apartment, and current rent average. The perceptron was trained on this dataset, utilizing a batch size of 32 over a series of 100 epochs and resulted in a mean absolute error of $46.13 and mean absolute percentage error of 5.32%. The results of this model were then compared to the results of a related research paper, which indicated that the created perceptron was more efficient. Additionally, comparing the model to a created Long-Short Term Memory model revealed comparable efficiencies while the perceptron was able to train faster. The perceptron model was then used to predict the future rent prices for the counties in Georgia, predicting an average increase of $41.

References

[1] X. Zhou, W. Tong, and D. Li, “Modeling housing rent in the Atlanta metropolitan area using textual information and deep learning,” *ISPRS International Journal of Geo-Information*, vol. 8, no. 8, p. 349, Aug. 2019. doi:10.3390/ijgi8080349

[2] H. Shen, L. Li, H. Zhu, Y. Liu, and Z. Luo, “Exploring a pricing model for urban rental houses from a geographical perspective,” *Land*, vol. 11, no. 1, p. 4, Dec. 2021. doi:10.3390/land11010004

[3] Z. Jiang and G. Shen, “Prediction of house price based on the back propagation neural network in the Keras Deep Learning Framework,” *2019 6th International Conference on Systems and Informatics (ICSAI)*, Nov. 2019. doi:10.1109/icsai48974.2019.9010071

[4] T. Yoshida, D. Murakami, and H. Seya, “Spatial prediction of apartment rent using regression-based and machine learning-based approaches with a large dataset,” *The Journal of Real Estate Finance and Economics*, Nov. 2022. doi:10.1007/s11146-022-09929-6

[5] J. Valente, S. Wu, A. Gelfand, and C. F. Sirmans, “Apartment rent prediction using spatial modeling,” *Journal of Real Estate Research*, vol. 27, no. 1, pp. 105–136, Jan. 2005. doi:10.1080/10835547.2005.12091148

[6] K. S. Rupesh and S. K. R, “Better Accuracy for House or Apartment Rent Prediction in Metro Cities Using CNN Compared to Support Vector Machine,” *Journal of Survey in Fisheries Sciences*, vol. 10, no. 1, pp. 2631–2641, 2023.

[7] P. Mody, M. Motiramani, and A. Singh, “Enhancing real estate market insights through Machine Learning: Predicting property prices with advanced data analytics,” *2023 4th IEEE Global Conference for Advancement in Technology (GCAT)*, Oct. 2023. doi:10.1109/gcat59970.2023.10353243

[8] W. K. O. Ho, B.-S. Tang, and S. W. Wong, “Predicting property prices with machine learning algorithms,” *Journal of Property Research*, vol. 38, no. 1, pp. 48–70, Oct. 2020. doi:10.1080/09599916.2020.1832558

[9] F. Wang, Y. Zou, H. Zhang, and H. Shi, “House price prediction approach based on Deep Learning and Arima model,” *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, pp. 303–307, Oct. 2019. doi:10.1109/iccsnt47585.2019.8962443

[10] K. Team, “Keras Documentation: Model training apis,” Keras, https://keras.io/api/models/model_training_apis/

[11] D. Mwiti, “How to build a deep learning model to predict employee retention using Keras and tensorflow,” DigitalOcean, https://www.digitalocean.com/community/tutorials/how-to-build-a-deep-learning-model-to-predict-employee-retention-using-keras-and-tensorflow

[12] R. Callis, P. Holley, and D. Truver, “QUARTERLY RESIDENTIAL VACANCIES AND HOMEOWNERSHIP, FIRST QUARTER 2020,” *CB20-58*, Apr. 2020.

[13] R. Callis, P. Holley, and D. Truver, “QUARTERLY RESIDENTIAL VACANCIES AND HOMEOWNERSHIP, FOURTH QUARTER 2020,” *CB21-15*, Feb. 2021.

Appendix

Appendix A

				Milestone #1: Project Proposal						
Component	**Tasks**	**Complete%**	**Current Status Memo**	**01/08**	**01/15**	**01/22**	**01/29**	**02/05**	**02/12**	**02/19**
Background Research	Select a Project	100%								
	Review Previous Work	100%								
	Develop Proposal	100%								
Deep Learning Model Creation	Develop Database	100%								
	Integrate Database into Model	100%								
	Data Preprocessing	100%								
	Create Prediction Algorithm	100%								
	Optimize Algorithm	75%								
	Develop Mid Term Report	100%								
Final Report	Compare Deep Learning Model to Other Models	0%								
	Predictions in Different Scenarios	0%								
	Produce Final Report	0%								

				Milestone #2: Mid Term Report						*Milestone #3: Final Report*			
Component	**Tasks**	**Complete%**	**Current Status Memo**	**02/26**	**03/04**	**03/11**	**03/18**	**03/25**	**04/01**	**04/08**	**04/15**	**04/22**	**04/29**
Background Research	Select a Project	100%											
	Review Previous Work	100%											
	Develop Proposal	100%											
Deep Learning Model Creation	Develop Database	100%											
	Integrate Database into Model	100%											
	Data Preprocessing	100%											
	Create Prediction Algorithm	100%											
	Optimize Algorithm	75%											
	Develop Mid Term Report	100%											
Final Report	Compare Deep Learning Model to Other Models	0%											
	Predictions in Different Scenarios	0%											
	Produce Final Report	0%											

Appendix B

county	2023 Studio	2024 Studio	2023 1-Bed	2024 1-Bed	2023 2-Bed	2024 2-Bed	2023 3-Bed	2024 3-Bed	2023 4-Bed	2024 4-Bed
Appling	602	625.43	606	627.16	772	800.39	1097	1141.94	1123	1166.97
Atkinson	609	632.79	612	633.47	772	800.35	1060	1102.72	1123	1166.92
Bacon	516	534.29	684	709.64	772	800.30	969	1006.35	1123	1166.87
Baker	602	625.28	606	627.02	772	800.25	1034	1075.11	1123	1166.83
Baldwin	658	684.52	719	746.60	812	842.55	997	1035.89	1315	1370.03
Banks	681	708.82	700	726.44	811	841.44	987	1025.26	1240	1290.59
Barrow	1345	1411.68	1375	1440.94	1553	1626.87	1890	1981.12	2308	2421.11
Bartow	1345	1411.63	1375	1440.89	1553	1626.82	1890	1981.07	2308	2421.07
Ben Hill	648	673.74	684	709.36	772	800.02	1027	1067.46	1315	1369.84
Berrien	583	604.89	586	605.57	772	799.97	1097	1141.52	1169	1215.24
Bibb	719	748.81	838	872.29	969	1008.47	1207	1257.91	1302	1355.98
Bleckley	583	604.80	586	605.48	772	799.88	1082	1125.54	1172	1218.32
Brantley	810	845.05	815	847.85	998	1039.07	1348	1407.08	1633	1706.28
Brooks	680	707.38	685	710.18	895	929.99	1219	1270.48	1411	1471.23
Bryan	1031	1078.90	1112	1162.15	1256	1312.09	1715	1795.49	2008	2103.16
Bulloch	706	734.81	711	737.61	887	921.43	1206	1256.62	1313	1367.39
Burke	750	781.35	890	927.05	1017	1059.00	1369	1429.12	1675	1750.56
Butts	831	867.04	836	869.84	1094	1140.46	1433	1496.83	1620	1692.29
Calhoun	644	669.04	648	670.78	772	799.55	982	1019.36	1231	1280.45
Camden	781	814.02	786	816.82	972	1011.22	1380	1440.63	1603	1674.20
Candler	602	624.49	606	626.22	772	799.45	1095	1138.88	1114	1156.50
Carroll	1345	1410.97	1375	1440.23	1553	1626.17	1890	1980.41	2308	2420.41
Catoosa	910	950.44	919	957.47	1067	1111.64	1372	1432.02	1619	1690.99
Charlton	648	673.04	684	708.65	772	799.31	1097	1140.86	1140	1183.88
Chatham	1031	1078.43	1112	1161.68	1256	1311.62	1715	1795.02	2008	2102.69
Chattahoochee	759	790.45	819	851.47	945	982.36	1276	1330.25	1610	1681.33
Chattooga	529	546.93	586	604.77	772	799.17	1061	1102.61	1240	1289.60

Cherokee	1345	1410.69	1375	1439.95	1553	1625.88	1890	1980.13	2308	2420.13
Clarke	825	860.18	877	912.73	995	1035.14	1335	1392.57	1517	1582.74
Clay	602	624.06	606	625.80	772	799.03	1097	1140.58	1123	1165.60
Clayton	1345	1410.55	1375	1439.81	1553	1625.74	1890	1979.99	2308	2419.99
Clinch	602	623.97	606	625.71	772	798.94	1097	1140.48	1123	1165.51
Cobb	1345	1410.45	1375	1439.72	1553	1625.65	1890	1979.90	2308	2419.89
Coffee	606	628.11	610	629.85	772	798.84	940	974.19	1037	1074.38
Colquitt	585	605.83	589	607.57	772	798.80	963	998.49	1135	1178.07
Columbia	750	780.45	890	926.16	1017	1058.10	1369	1428.23	1675	1749.66
Cook	602	623.73	606	625.47	772	798.70	1006	1043.92	1169	1213.97
Coweta	1345	1410.22	1375	1439.48	1553	1625.41	1890	1979.66	2308	2419.66
Crawford	719	747.50	838	870.97	969	1007.15	1207	1256.60	1302	1354.67
Crisp	663	688.17	673	696.26	789	816.56	960	995.08	1060	1098.44
Dade	910	949.59	919	956.62	1067	1110.80	1372	1431.17	1619	1690.15
Dawson	1345	1410.03	1375	1439.29	1553	1625.23	1890	1979.48	2308	2419.47
Decatur	620	642.51	624	644.25	772	798.42	984	1020.35	1037	1073.96
DeKalb	1345	1409.94	1375	1439.20	1553	1625.13	1890	1979.38	2308	2419.38
Dodge	583	603.25	586	603.93	772	798.33	940	973.67	1048	1085.51
Dooly	648	672.01	649	670.57	772	798.28	940	973.63	1037	1073.81
Dougherty	735	764.06	748	775.32	880	912.56	1153	1199.06	1402	1460.15
Douglas	1345	1409.75	1375	1439.01	1553	1624.94	1890	1979.19	2308	2419.19
Early	602	623.17	606	624.91	772	798.14	1000	1037.00	1110	1150.95
Echols	680	705.69	685	708.49	895	928.30	1219	1268.79	1411	1469.54
Effingham	1031	1077.21	1112	1160.46	1256	1310.40	1715	1793.80	2008	2101.47
Elbert	594	614.56	598	616.30	787	813.88	973	1008.28	1057	1094.70
Emanuel	516	531.94	613	632.13	772	797.95	980	1015.64	1315	1367.77
Evans	631	653.64	635	655.37	772	797.90	953	987.01	1037	1073.44
Fannin	662	686.40	666	688.14	788	814.79	1120	1163.75	1146	1188.78
Fayette	1345	1409.37	1375	1438.64	1553	1624.57	1890	1978.82	2308	2418.81

Floyd	726	754.06	731	756.86	962	998.90	1254	1305.51	1500	1563.42
Forsyth	1345	1409.28	1375	1438.54	1553	1624.48	1890	1978.72	2308	2418.72
Franklin	583	602.59	586	603.27	772	797.67	1030	1068.29	1123	1164.24
Fulton	1345	1409.19	1375	1438.45	1553	1624.38	1890	1978.63	2308	2418.62
Gilmer	575	594.03	653	674.10	860	890.73	1130	1174.05	1465	1526.19
Glascock	648	671.26	684	706.87	772	797.53	1044	1082.97	1048	1084.71
Glynn	810	842.70	815	845.50	998	1036.72	1348	1404.73	1633	1703.94
Gordon	582	601.30	655	676.08	790	816.49	1107	1149.57	1216	1262.46
Grady	606	626.65	610	628.39	803	830.20	1082	1123.05	1105	1144.91
Greene	639	661.54	643	663.28	819	847.09	1027	1064.79	1144	1186.14
Gwinnett	1345	1408.86	1375	1438.12	1553	1624.05	1890	1978.30	2308	2418.30
Habersham	516	531.24	652	672.71	772	797.25	1049	1087.98	1053	1089.72
Hall	967	1008.62	1062	1106.69	1200	1250.28	1521	1587.59	1728	1804.22
Hancock	648	670.88	684	706.49	772	797.15	1048	1086.83	1315	1366.98
Haralson	750	778.81	755	781.61	913	946.37	1298	1351.43	1555	1620.99
Harris	759	788.29	819	849.31	945	980.20	1276	1328.09	1610	1679.17
Hart	661	684.50	665	686.24	876	907.11	1066	1105.74	1264	1312.85
Heard	1345	1408.53	1375	1437.79	1553	1623.72	1890	1977.97	2308	2417.97
Henry	1345	1408.48	1375	1437.74	1553	1623.68	1890	1977.93	2308	2417.92
Houston	848	882.32	934	970.86	1055	1096.45	1432	1493.04	1692	1765.78
Irwin	602	621.86	606	623.59	772	796.82	940	972.17	1123	1163.40
Jackson	686	710.73	691	713.53	909	941.80	1238	1287.58	1351	1404.71
Jasper	1345	1408.29	1375	1437.56	1553	1623.49	1890	1977.74	2308	2417.73
Jeff Davis	602	621.72	606	623.45	772	796.68	940	972.03	1113	1152.67
Jefferson	583	601.56	586	602.24	772	796.64	940	971.98	1315	1366.46
Jenkins	602	621.62	606	623.36	772	796.59	1097	1138.14	1123	1163.16
Johnson	648	670.27	684	705.88	772	796.54	1062	1101.04	1123	1163.12
Jones	719	745.38	838	868.86	969	1005.04	1207	1254.49	1302	1352.56
Lamar	717	743.22	722	746.02	950	984.88	1256	1306.31	1276	1324.99

Lanier	680	704.00	685	706.80	895	926.61	1219	1267.10	1411	1467.85
Laurens	603	622.45	607	624.18	772	796.35	1064	1102.97	1315	1366.18
Lee	735	762.13	748	773.40	880	910.64	1153	1197.13	1402	1458.23
Liberty	928	966.39	949	986.13	1072	1113.84	1524	1589.82	1826	1907.02
Lincoln	584	602.19	693	715.08	790	815.27	1057	1095.42	1061	1097.15
Long	705	730.23	721	744.68	814	840.63	1150	1193.82	1387	1442.21
Lowndes	680	703.72	685	706.52	895	926.33	1219	1266.81	1411	1467.57
Lumpkin	768	796.83	773	799.63	985	1021.55	1236	1284.76	1678	1750.16
Macon	587	605.18	591	606.92	772	796.03	949	980.90	1315	1365.85
Madison	825	857.08	877	909.63	995	1032.04	1335	1389.47	1517	1579.64
Marion	759	787.16	819	848.18	945	979.07	1276	1326.97	1610	1678.04
McDuffie	750	777.59	890	923.30	1017	1055.24	1369	1425.37	1675	1746.80
McIntosh	810	841.06	815	843.85	998	1035.08	1348	1403.09	1633	1702.29
Meriwether	702	726.68	707	729.48	930	963.05	1206	1252.72	1281	1329.62
Miller	648	669.47	684	705.08	772	795.74	1011	1046.25	1141	1181.37
Mitchell	648	669.42	684	705.04	772	795.70	940	971.04	1037	1071.23
Monroe	649	670.44	761	786.50	859	887.75	1045	1082.15	1463	1522.14
Montgomery	619	638.63	623	640.37	772	795.60	1097	1137.15	1290	1338.96
Morgan	814	845.01	819	847.81	1078	1119.49	1328	1381.64	1836	1916.90
Murray	561	577.14	614	630.75	808	833.62	1088	1127.53	1344	1396.03
Muscogee	759	786.69	819	847.71	945	978.60	1276	1326.50	1610	1677.57
Newton	1345	1406.98	1375	1436.24	1553	1622.17	1890	1976.42	2308	2416.42
Oconee	825	856.47	877	909.02	995	1031.43	1335	1388.86	1517	1579.03
Oglethorpe	825	856.42	877	908.97	995	1031.39	1335	1388.81	1517	1578.98
Paulding	1345	1406.84	1375	1436.10	1553	1622.03	1890	1976.28	2308	2416.28
Peach	590	607.56	670	689.75	882	911.67	1121	1162.18	1234	1279.30
Pickens	1345	1406.75	1375	1436.01	1553	1621.94	1890	1976.19	2308	2416.18
Pierce	639	659.33	643	661.07	819	844.89	997	1030.82	1119	1157.47
Pike	1345	1406.65	1375	1435.91	1553	1621.85	1890	1976.09	2308	2416.09

Polk	623	642.30	628	645.10	826	852.20	1052	1088.95	1110	1147.85
Pulaski	623	642.26	627	643.99	772	794.99	1078	1116.43	1123	1161.57
Putnam	715	739.60	719	741.34	916	947.38	1115	1155.55	1560	1624.12
Quitman	602	619.93	606	621.67	772	794.90	940	970.25	1123	1161.47
Rabun	641	661.17	646	663.97	850	877.42	1060	1097.23	1376	1429.25
Randolph	602	619.84	606	621.58	772	794.81	982	1014.61	1037	1070.34
Richmond	750	776.46	890	922.17	1017	1054.11	1369	1424.24	1675	1745.67
Rockdale	1345	1406.28	1375	1435.54	1553	1621.47	1890	1975.72	2308	2415.71
Schley	602	619.70	606	621.43	772	794.66	1087	1125.62	1123	1161.24
Screven	583	599.54	586	600.22	772	794.62	1091	1129.81	1123	1161.19
Seminole	643	663.00	646	663.68	823	848.56	1066	1103.30	1329	1379.21
Spalding	1345	1406.09	1375	1435.35	1553	1621.28	1890	1975.53	2308	2415.53
Stephens	620	638.56	624	640.30	822	847.41	1071	1108.50	1315	1364.30
Stewart	583	599.35	586	600.03	772	794.43	940	969.78	1109	1146.18
Sumter	615	633.18	619	634.91	815	839.90	992	1024.78	1188	1229.76
Talbot	621	639.48	669	687.80	772	794.34	1015	1049.08	1315	1364.16
Taliaferro	602	619.32	606	621.06	772	794.29	940	969.64	1123	1160.86
Tattnall	648	667.97	676	695.11	772	794.24	1097	1135.79	1315	1364.06
Taylor	516	528.19	606	620.97	772	794.19	964	994.95	1127	1165.00
Telfair	516	528.14	606	620.92	772	794.15	940	969.49	1037	1069.68
Terrell	735	759.93	748	771.19	880	908.43	1153	1194.93	1402	1456.02
Thomas	765	791.64	769	793.37	943	975.07	1215	1260.51	1373	1425.27
Tift	648	667.73	655	672.65	772	794.01	1016	1049.81	1194	1235.74
Toombs	583	598.88	586	599.56	772	793.96	1084	1121.74	1266	1311.91
Towns	673	694.10	687	706.43	801	824.61	1138	1178.86	1364	1415.61
Treutlen	583	598.79	586	599.46	772	793.87	1059	1095.19	1065	1099.04
Troup	703	725.77	708	728.57	932	963.19	1216	1261.34	1323	1372.11
Turner	602	618.80	606	620.54	772	793.77	940	969.12	1037	1069.31
Twiggs	719	742.61	838	866.09	969	1002.27	1207	1251.72	1302	1349.79

Union	602	618.71	606	620.45	772	793.68	1097	1135.22	1315	1363.50
Upson	516	527.62	674	692.39	772	793.63	1054	1089.66	1093	1128.45
Walker	910	944.66	919	951.69	1067	1105.87	1372	1426.24	1619	1685.22
Walton	1345	1405.10	1375	1434.36	1553	1620.30	1890	1974.54	2308	2414.54
Ware	615	632.28	619	634.02	815	839.01	992	1023.88	1239	1282.86
Warren	583	598.36	586	599.04	772	793.44	1005	1037.60	1315	1363.27
Washington	648	667.12	684	702.74	772	793.40	1046	1080.95	1204	1245.72
Wayne	583	598.27	586	598.95	772	793.35	970	1000.45	1241	1284.84
Webster	602	618.33	606	620.07	772	793.30	988	1019.46	1123	1159.88
Wheeler	583	598.17	685	703.65	872	899.11	1061	1096.69	1171	1210.64
White	737	761.15	777	801.00	878	905.42	1093	1130.52	1222	1264.58
Whitfield	590	605.49	655	671.80	862	888.43	1138	1178.11	1159	1197.84
Wilcox	602	618.15	606	619.89	772	793.11	1097	1134.66	1123	1159.69
Wilkes	601	617.04	605	618.78	797	819.53	1045	1079.57	1071	1104.59
Wilkinson	648	666.75	864	892.91	772	793.02	943	971.54	1037	1068.55
Worth	735	758.80	748	770.06	880	907.30	1153	1193.80	1402	1454.89