Forecasting City Finances

Summary

The budgeting at governmental institutions is as important, if not more important, as at private companies. Evaluations of the necessary upcoming expenditures and potential sources of income for the city are thoroughly studied and compiled every year. Once prepared, "the budget estimates are then submitted to a city council or board for review and modification, often with citizen input from public meetings. The budget is then legally approved and adopted." *(source: National League of Cities)*

To be able to ensure that a city is able to cover all its expected spendings, proper forecasting of the city's income is primordial. **The following study explored this problem and proposed a solution to that end.**

The Dataset

The data for this effort is obtained from the Lincoln Institute of Land Policy. It proposes city-level finances for 150 of the largest U.S. cities. Moreover, the team at the Lincoln Institute of Land Policy have standardized these finances (since they are at different scales for different cities) allowing for easier modeling and comparisons across different cities. This database, developed by Adam Langley, is called "Fiscally Standardized Cities database" and covers the years 1977 to 2016, taking into account the ways in which finances and responsibilities overlap between cities, counties, school districts, and other local governments.

Data Wrangling

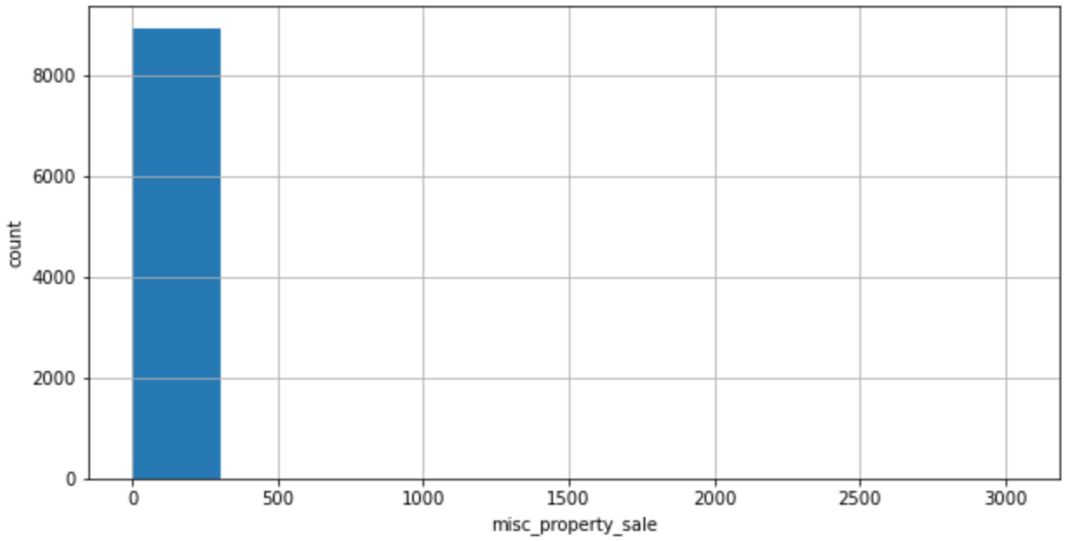
Data Wrangling Notebook

In dealing with the data in its raw form, a few issues were noted and some were treated before further steps were taken. The major issues were as follows:

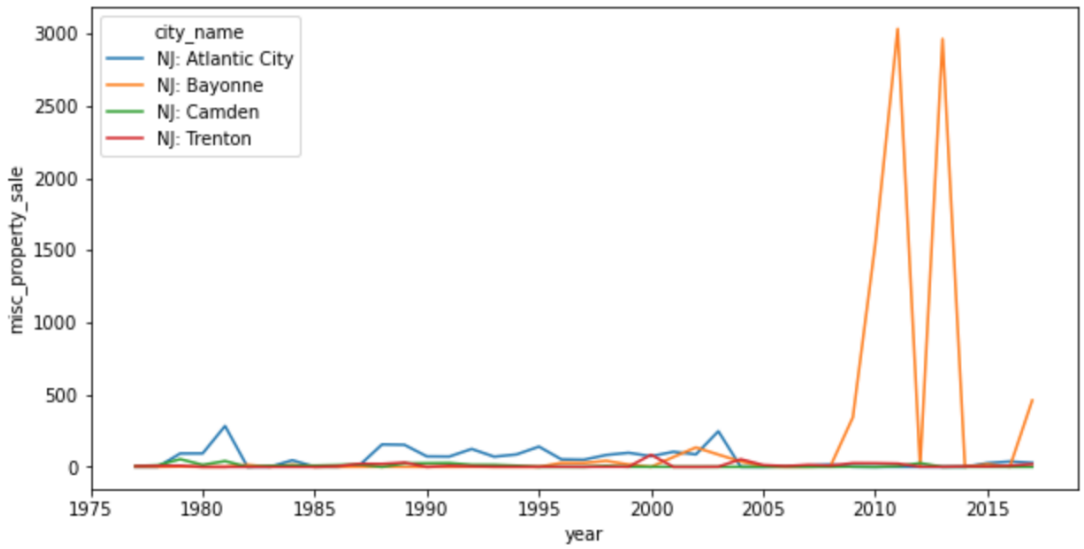
1. "Duplicate" features: The dataset contained a very large number of features (663 columns to be exact). On exploring these features, it was noted that the features were repeated to represent finances at different governmental levels (for eg: *"rev_total"* v/s *"rev_total_city"* v/s *"rev_total_cnty"*). By checking the data source, it was identified that the features of interest (fiscally standardized finance features) were the ones without any suffix (like *"city"* or *"cnty"*). Getting rid of the irrelevant features dropped the count from 663 to 139)

2. Null values: Checking for null values in the remaining dataset showed that there were only 5 features with null entries. By checking the data source, it was found that these exact 5 fields were not listed as intended fields for the report. Therefore it was concluded that these fields should not be present in the database and were likely erroneously entered. For this reason, these fields were also dropped.

3. Outliers: Histograms were plotted for all the fields to check the distribution of the values for all available features. This highlighted that many features seemed to contain outliers. For example, consider the histogram for the field '*misc_property_sale*'



The range for the above histogram is quite wide while all the majority of the values are concentrated towards the lower end of the spectrum. This signals that there are some outliers in this field. By further exploring this field, it was found that the outlier is related to one specific city, **NJ: Bayonne**.



Comparing all data points for this field and this city against neighboring cities, we can clearly identify the outliers. For the years 2010 and 2013, NJ: Bayonne has two very large (~300x higher) values for 'misc_property_sale'. From further external research, it was found that there is also no evidence to justify that significant property sales occurred in Bayonne. One way to fix this outlier would have been to replace it with the average of neighboring non-outlier values. Given that not all fields would be used in the modeling, the necessary changes for outliers were left for the EDA step once the choices of fields were made.

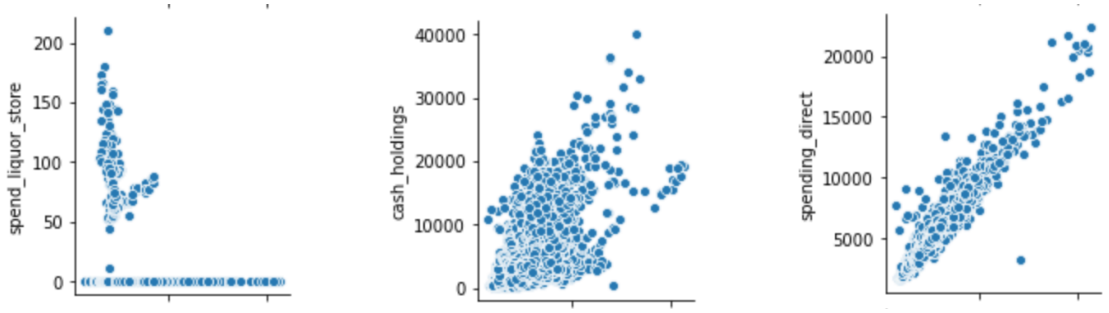
EDA

EDA Notebook

Exploratory Data Analysis was conducted in order to identify the features and methodology that would be best suited for the objective at hand. The overarching goal of the study is to predict a city's income for the upcoming year, using currently available historical data. Therefore, the target variable from our dataset is rather intuitively the **rev_total** field.

The exploratory variables on the other hand are trickier since there are two potential approaches:
Method 1: use historical expenditure data to predict income (**regression models**), or
Method 2: use historical income data to predict income (**auto-regressive moving average models**).

A deeper study of the features available show that there were a lot of sub-categories of the fields which represented too much granularity. For example: *'education', 'educ_higher'* and *'educ_elem_sec'* were fields that all summed up to another field in the dataset '*education_services'.* Since we did not need all levels of granularity but rather just the high level spendings and earnings of the city, careful cleaning was conducted to only keep the highest level of the different categories in the dataset. Checking the correlation of our target variable with the remaining features revealed a lot of interesting information that allowed us to choose a path forward:



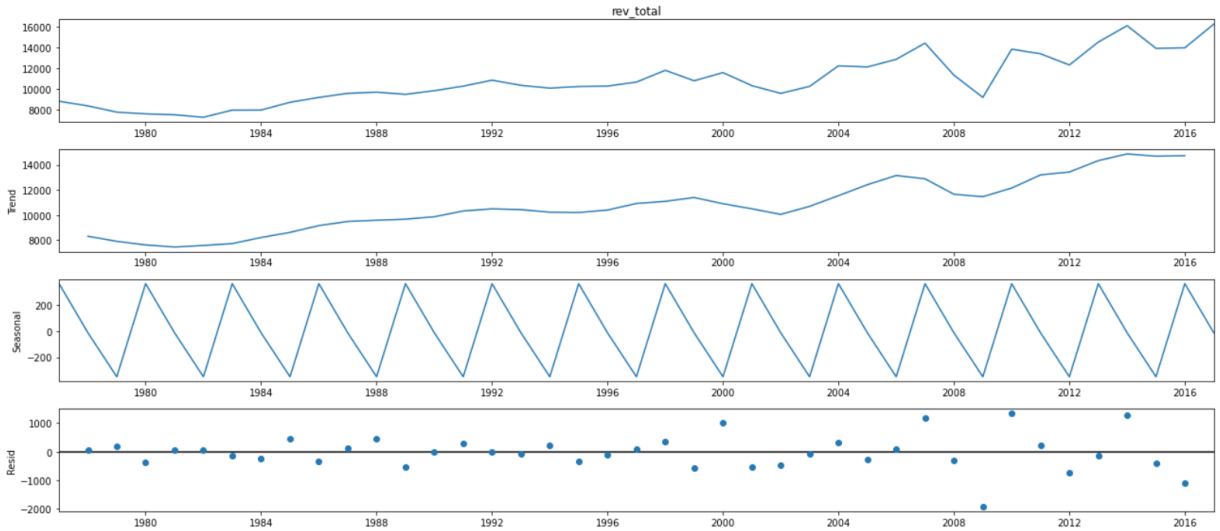
The three examples above are scatter plots of three features against the target variable **rev_total.** As visible there were some features that were completely unrelated to the city's income, such as the spending on liquor stores, while other features showed more correlation such as cash holdings of the city. However, rather intuitively, the feature that showed the most correlation with the city's income was the city's direct spending. This exercise highlighted that the features at hand were not instructive features that helped inform the city's income, but rather features that were simply resulting from the city's income. Therefore, trying to use these features to forecast the income did not make much sense.

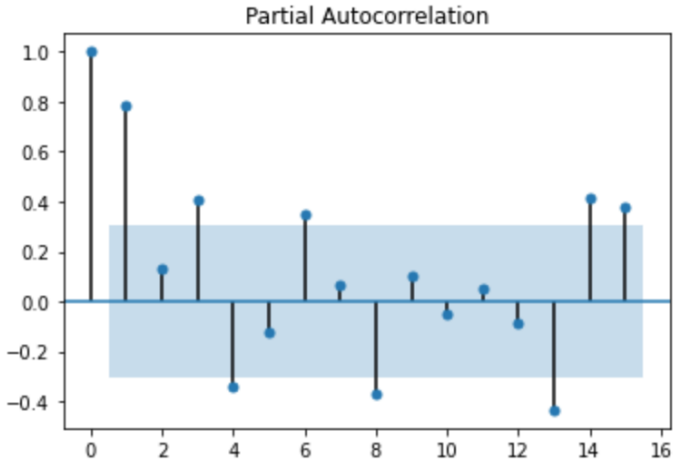
With this finding, Method 2 (auto-regressive moving average models) was chosen as the best way to predict the city's income.

Preprocessing

[Preprocessing Notebook]

Since the chosen methodology was ARIMA, a preprocessing step was taken to explore the relevant characteristics for time-series analyses, such as auto-correlation, seasonality, and stationarity. There is a time-series for each of the cities in the dataset, and therefore, so as to not be repetitive, the preprocessing was done only using **NY: New York** as a subset of the dataset.





The above plots are obtained using the relevant methods from the statsmodels library. The first one shows the trend and seasonality and the second partial auto-correlation for up to 15 lags. From these plots, we are able to find that there was no strong seasonality, and that the time series is mostly correlated to 1 or 2 lags. We can also find that the time-series was not stationary but we confirmed this finding by applying the Dickey-Fuller test. In trying several methods of making the data stationary, the one that led to the best result was a log-difference method.

Modeling

Modeling Notebook

Given that each city has its own time-series, it only makes sense that each city would need an ARIMA model specific to its data. Therefore, the methodology for the modeling step was to find the hyperparameters (which for ARIMA models are p, d, and q) that best suit each time-series for each city - essentially treating each city as a separate dataset in itself.

The way that the training was performed was as follows. The first 85% of the dataset for a city is used to train the ARIMA model. Then, the model is used to predict the next data point. This data point is then added to the first training set and then the next data point is predicted. This is repeated until the remaining 15% of the dataset is predicted. To evaluate the performance of the ARIMA model, the mean squared error of the predicted values is calculated. This training methodology is repeated for each p-d-q hyperparameter combination, and the one that leads to the lowest error is chosen as the best ARIMA model for that city. Finally, this process is done for each city in order to get the best model for each city. During the training process, the absolute percent error between the predicted data and the actual data is also calculated and stored.

state_city	best_ARIMA_order	Training Error(%)	train_err_num
AK: Anchorage	(2, 1, 2)	30.58%	0.3058
AK: Fairbanks	(0, 1, 2)	24.60%	0.2460
AL: Birmingham	(2, 1, 1)	9.87%	0.0987
AL: Gadsden	(1, 1, 1)	1.90%	0.0190
AL: Mobile	(0, 1, 0)	8.75%	0.0875

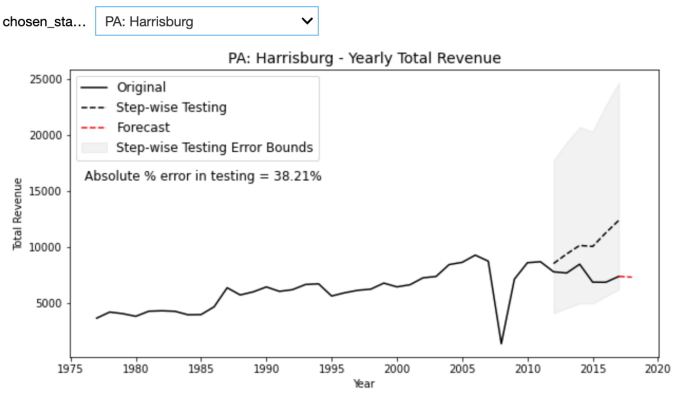
The above snippet of data frame is the result of the best ARIMA models obtained from training every city's data using the described methodology.

Results & Conclusion

The best ARIMA models obtained during training can be used to train the entire dataset for each city in order to forecast an out-of-sample data point, which was the desired purpose of this whole exercise! In order to quickly toggle between the forecasts for different cities, a widget was written with a drop down menu. Below are the cities with the worst and best performing predictive models.

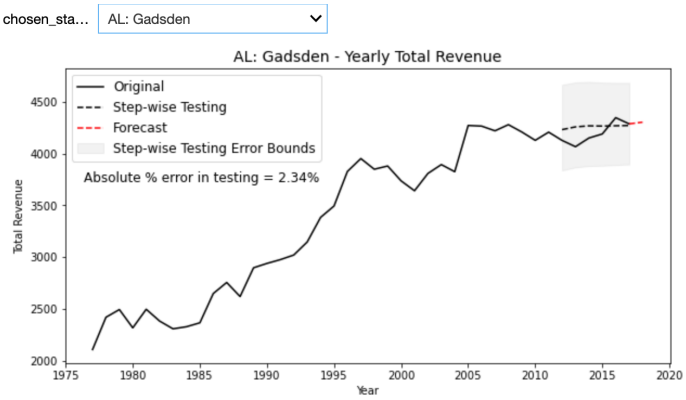
Cities with the worst performing models

state_city	best_ARIMA_order	Training Error(%)
PA: Harrisburg	(2, 0, 0)	923.82%
MO: St. Louis	(0, 1, 2)	184.24%
CA: San Diego	(0, 1, 2)	182.84%
CO: Denver	(2, 1, 2)	135.42%
MA: Boston	(0, 0, 0)	125.77%



Cities with the best performing models

state_city	best_ARIMA_order	Training Error(%)
AL: Gadsden	(1, 1, 1)	1.90%
AR: Little Rock	(0, 2, 2)	2.22%
SC: Charleston	(0, 0, 1)	2.55%
TX: El Paso	(2, 1, 0)	2.64%
MS: Jackson	(2, 0, 2)	2.90%



In the worst performing model, the training error was as high as 924%! However, this error only represents the uncertainty in the model, and the actual deviation from the real data measured as “Absolute % error in testing” is only 38%. In the best model, this metric drops to <2.5%! Comparing the worst models to the better ones, it is found that the reason behind the poor performance was the presence of big/significant fluctuations in the historical data (like the one visible in the plot above). These dips were not removed from the training/modeling process, because they represent real life events such as the 2008 economic crash in this case. Nevertheless, a potential improvement to the models would be to train with some treatment of these big dips.