M13Mwongo / dsc-phase4-project

Code Issues Pull requests Actions Projects Security Insights Settings

0 stars 0 forks 1 watching 9 Branches 0 Tags Activity

Public repository

main 9 Branches 0 Tags Go to file t Go to file + Add file Code

M13Mwongo Merge pull request #1 from M13Mwongo/mwiti 2 minutes ago

.ipynb_checkpoints	done with readme	4 minutes ago
data	Updated file	17 hours ago
images	included adan readme doc	15 hours ago
presentation	done with readme	4 minutes ago
README.md	done with readme	4 minutes ago
Zipcode_predict.csv	Updated file	17 hours ago
house_value_predictor.pkl	Updated file	17 hours ago
index.ipynb	done with readme	4 minutes ago
output_2.xlsx	Updated file	17 hours ago
requirements.txt	syncing files with isaack	3 minutes ago
save_model.py	syncing files with isaack	3 minutes ago
zipCode_value_predictor.pkl	Updated file	17 hours ago

README

UNLOCKING REAL ESTATE SUCCESS WITH TIME SERIES MODELLING

In a bustling cityscape where opportunities arise and markets evolve, a cutting-edge solution is on the horizon for a visionary real estate investment firm – a solution that promises to transcend conventional boundaries and redefine the landscape of strategic decision-making. Welcome to the world of Time Series Modeling, a realm where data becomes the compass for astute investments.

Our journey begins with a challenge posed by a forward-thinking real estate investment firm - BarnaBee & Co.. Faced with the seemingly straightforward question – "What are the top 5 best zip codes for us to invest in?" – the stakes are high, and the road to success is paved with data-driven insights.

Introduction

The seemingly simple query concealed intricate nuances - profits or risks, short-term gains or long-term stability? These were all factors to consider when trying to answer the question laid out by BarnaBee and Co. - ***what are the 5 best zip codes to invest in?*** Furthermore, in the pursuit of excellence, the forged metrics aligned with the investment firm's vision. It wasn't merely about minimizing mean squared error; it was about orchestrating a delicate balance between risk and profitability. The ROI yield emerged as the guiding star, steering investment decisions with unwavering precision.

These complexities were adeptly navigated, transforming ambiguity into actionable intelligence and truly unearthing the value of real estate.

Business Understanding

Problem Statement

Real estate represents a significant portion of most people's wealth, and this is especially true for many homeowners in the United States. A number of factors drive the real estate market including government policies, demographics of the potential buyers,affordability, disparity in housing access, location, cash flows and liquidity as well as the current economic climate. The many variables can make the process tedious for the buyers. Naruto consultants hopes to create a predictive time series model that can help to determine the top five zipcodes in which to invest in.

This forecasting model will help adress the anticipation of any other financial crisis with apt predictive capabilities. The best model shuld capitalize more on the **R2 score** which will be our optmization metric.

Objectives

a) Main Objectives

Investigate and establish the 5 best ZipCodes(Regions) that are the best to invest in that guarantee a good Return on investment.

b) Specific Objective

- Build a time series regression modell to predict the average house prices in the US and top 5 best performing zipcodes.
- To conduct exploratory data analysis to understand the temporal patterns and identify any underlying trends, seasonality, or irregularities in the data. Missing Values Analysis
- Investigate impact of the 2008 recession on the housing market per ZipCode(Region).

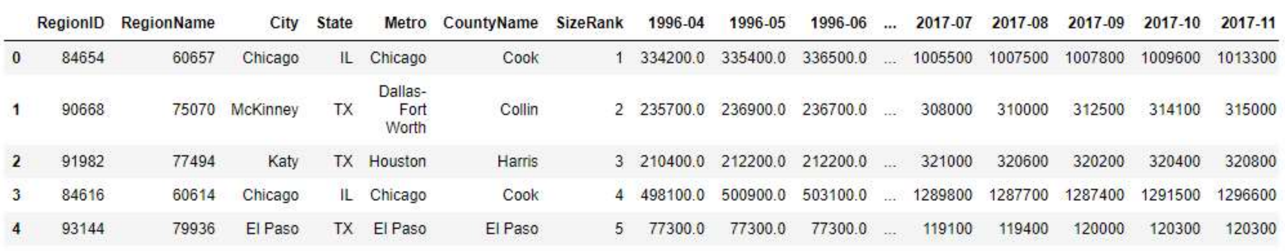
- To utilize univariate time series models, such as ARIMA, to capture and forecast the temporal patterns of individual zip codes. Multivariate Time Series Modeling:
- To explore multivariate time series models, considering factors like seasonality, economic indicators, and other relevant features to improve forecasting accuracy. Model Evaluation
- To evaluate the performance of time series models using appropriate metrics, ensuring reliability and effectiveness in predicting real estate prices.
- To determine the top 5 zip codes for investment based on the forecasted real estate prices, considering profit margins, risks, and other relevant factors. Documentation and Reporting:

Potential Challenges

- Potentially large amounts of missing data with no conclusive reason, forcing assumptions to be made.
- Some trends may be hard to explain due to differences in locale and perceptions of some matters.
- High seasonality caused by the 2008 market crash.

Data Understanding

The Zillow real estate dataset, available in the file zillow_data.csv, is a comprehensive collection of median housing sales values for various zip codes. The dataset follows the Wide Format, where columns represent median sales values for specific months and years. While this format is intuitive, it presents challenges for effective machine learning due to its reliance on metadata-dependent column names.

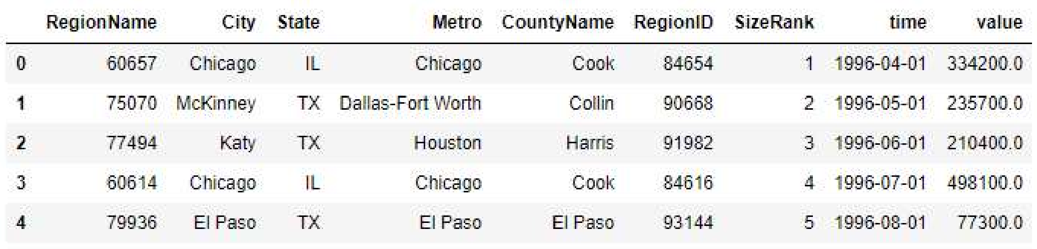


	RegionID	RegionName	City	State	Metro	CountyName	SizeRank	1996-04	1996-05	1996-06	...	2017-07	2017-08	2017-09	2017-10	2017-11
0	84654	60657	Chicago	IL	Chicago	Cook	1	334200.0	335400.0	336500.0	...	1005500	1007500	1007800	1009600	1013300
1	90668	75070	McKinney	TX	Dallas-Fort Worth	Collin	2	235700.0	236900.0	236700.0	...	308000	310000	312500	314100	315000
2	91982	77494	Katy	TX	Houston	Harris	3	210400.0	212200.0	212200.0	...	321000	320600	320200	320400	320800
3	84616	60614	Chicago	IL	Chicago	Cook	4	498100.0	500900.0	503100.0	...	1289800	1287700	1287400	1291500	1296600
4	93144	79936	El Paso	TX	El Paso	El Paso	5	77300.0	77300.0	77300.0	...	119100	119400	120000	120300	120300

To address these challenges, the data undergoes a transformation from Wide to Long Format using the pd.melt() method. In the Long Format, each row corresponds to a unique time and zipcode combination, enhancing compatibility with machine learning algorithms.

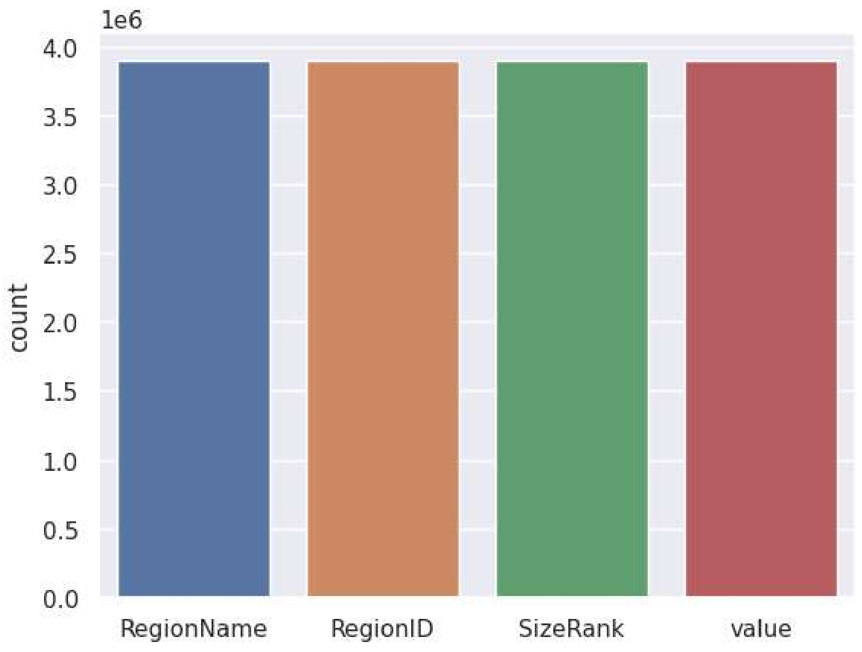
The transition to a Long Format is pivotal for unleashing the potential of the Zillow real estate dataset. By reshaping the data, we create a structure that enhances machine learning compatibility, making it suitable for models like ARIMA. This transformation ensures effective utilization of temporal patterns, providing valuable insights and accurate forecasting capabilities.

You'll notice that the first seven columns look like any other dataset you're used to working with. However, column 8 refers to the median housing sales values for April 1996, column 9 for May 1996, and so on. This is called Wide Format, and it makes the dataframe intuitive and easy to read. However, there are problems with this format when it comes to actually learning from the data because the data only makes sense if you know the name of the column where the data can be found. Since column names are metadata, our algorithms will miss out on what dates each value is for. This means that before we pass this data to our ARIMA model, we'll need to reshape our dataset to Long Format. Reshaped into the long format, the data frame above would now look like this:

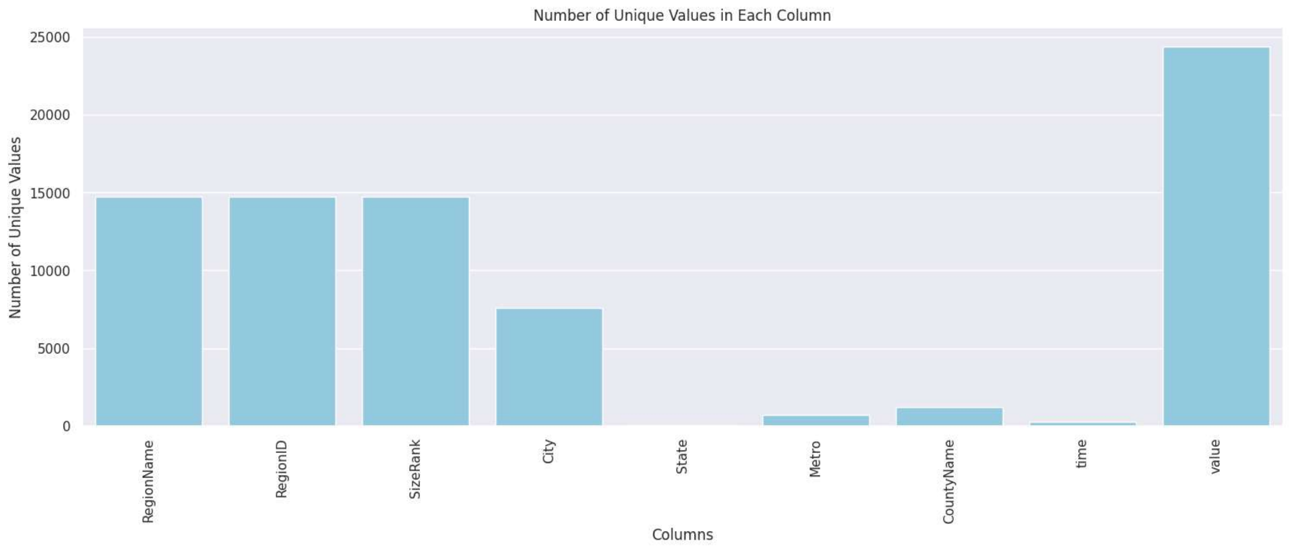


	RegionName	City	State	Metro	CountyName	RegionID	SizeRank	time	value
0	60657	Chicago	IL	Chicago	Cook	84654	1	1996-04-01	334200.0
1	75070	McKinney	TX	Dallas-Fort Worth	Collin	90668	2	1996-05-01	235700.0
2	77494	Katy	TX	Houston	Harris	91982	3	1996-06-01	210400.0
3	60614	Chicago	IL	Chicago	Cook	84616	4	1996-07-01	498100.0
4	79936	El Paso	TX	El Paso	El Paso	93144	5	1996-08-01	77300.0

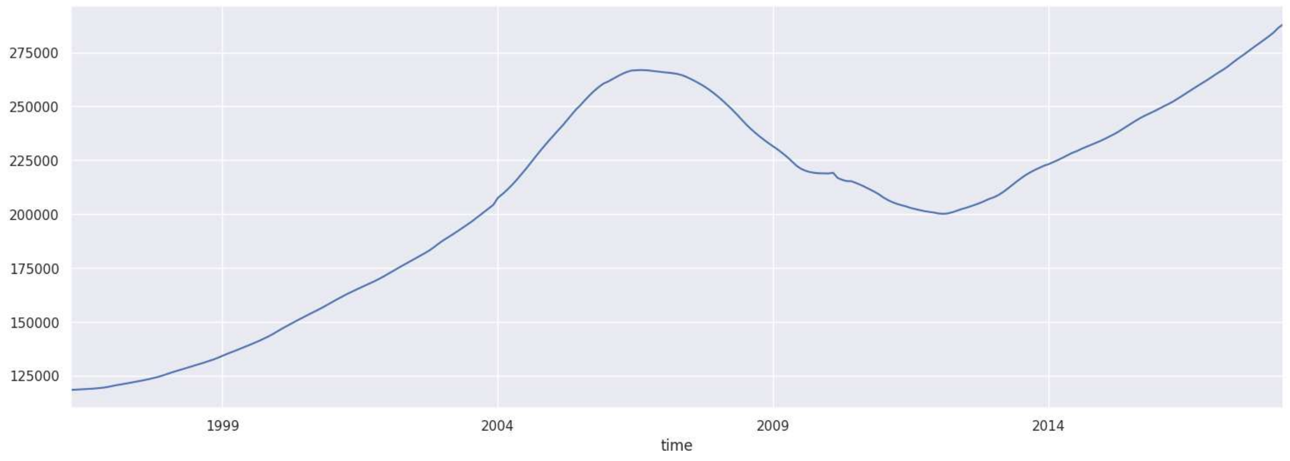
After reshaping the data, we perform a comprehensive analysis to detect and evaluate any missing values in the dataset. For this examination, we utilize Seaborn's countplot to visualize the distribution of missing values.



We have also investigated the number of unique values in the dataframe, focusing on key columns such as RegionID, RegionName, City, State, and Metro, time, value, and also county name



We have also generated a plot illustrating the connection between time and the mean values i



Exploratory Data analysis

- To analyze the real estate market trends is viable only through creation of pivot tables and grouping data into various frequencies as well as filtering down to specific regions.
- Create visuals like bar charts, line plots and graphs to inspect the after-effects of the crisis.

Modelling

Data for modelling should be in a long format and the column containing the datetime values set as index. This is necessary because it ensures easier resampling adn data aggregation on various frequencies. This stage involves using time series forecasting models to analyze the forecasted values and trends. The various time series models include ARIMA, SARIMAX and Prophet models. The SARIMAX handles the seasonality well but is usually prone to overfitting.

As good modelling approach dictates, it is often good practic eto start out with a baseline model with pdq value combinations obtained from the auto-correlation plots. This model at most times has no tuning applied. The subsequent models shouls however have to undergo some tuning so as to avoid the problem of overfitting which is pretty common in most time series models.

The subsequent model employs the use of the **auto-arima** model to find the best pdq value combinations with the aim of minimizing the **AIC** csores. The lower the **AIC** scores, the better the **ARIMA** model. It is however important to use the **TimeSeriesSplit** as it splits the data into various folds and seeks to find the optimum **AIC** scores within each fold.

Challenge: The values for the best possible combinations may be overwhelming because it requires iterating through every possible combination for each fold and finding the conspicuous values with the lowest scores.

After training the **Auto Regressive models**, we moved on to our third model which is the **Prophet** model which has exhibited great results in the recent past. The Prophet model is the ideal model because of the concept of adjusting the parameters to capture the seasonality in the time series data.

The zillow data had a major problem regarding home prices when the great 2008 financial crisis happened fueled by the risky mortages given to low income earners. This piled up debt an the bubble burst when th edebt was no longer fiscal and bearable. In future anticipation of such an event, occurring, the Prophet model is ideal and this is due to the easier adjustments of the seasonal component specifically **Seasonality trend, holiday trend** and **Regressors** addition to capture every other external factors driving the target higher or downwards.

Use of an **ETL** pipeline like method came in handy because of the simplification of feeding any data for prediction provided it has the datetime columns and the target columns.

Challenge:

- Using the class without splitting the data using the **TimeSeriesSplit** results to a mismatch between the real and forecasted values. This results to a difficulty of evaluating the whole pipleine. It is therefore necessary to go through the documentation [https://facebook.github.io/prophet/docs/non-daily_data.html] which comes in handy and clearly explains how to do forecatsing on various seasons.
- **Time** taken to get the average values of the metrics to evaluate our model is long. This is because of the use ot the **TimeSeriesSplit** which uses a cross-validation kind of approach to get the optimal results.

The first ETL pipeline chains everything from data ingestion, transforming, training and evaluating the model. It takes in resampled data obtained after transforming the data from wide to long formart. The data is resampled on a monthly basis and contains values of the average house prices for every month. Should the investor like to know the average house values in the near future, the model comes in handy.

The second **TimeSeriesPipeline** however takes in data filtered from the specific ZipCodes and there forecasted values plotted using the plot function in Prophet.

Challenge: Both pipelines take in quite a lot of time in evaluation of the models. The solution here is training the models on free **GPUs** avilable in the cloud so as to minimize the computational power.

Future Steps

1. Monitor the models house value predictions and scores at production.
2. Deploy both models(**house_value_predictor.pkl** & **Zipcode_predict.pkl**) for easier forecasting.

Deployment(Streamlit)

Testing the model at a production level, the pickled file **house_value_predictor.pkl** should be loaded either using the joblib or loaded into a streamlit environment. The model versions needed to run the model are contained in the **requirements.txt** file. The model runs on python version 3.10. This pickled file takes in datetime inputs and outputs the average predicted house values factoring in the seasonality the model learned during the **2008 market crash**.

To forecast house values on the top 30 best performing zipcodes, the pickled file **Zipcode_predict.pkl** should be loaded in a streamlit environment using the **save_model.py** file. It also takes in datetime input as well as specifying zip codes to get the forecasted value.

Deploying the model to other environments like **Heroku** will however be documented later.

Interpreting Results

Modelling

Data for modelling should be in a long format and the column containing the datetime values set as index. This is necessary because it ensures easier resampling adn data aggregation on various frequencies. This stage involves using time series forecasting models to analyze the forecasted values and trends. The various time series models include ARIMA, SARIMAX and Prophet models. The SARIMAX handles the seasonlaity well but is usually prone to overfitting.

As good modelling approach dictates, it is often good practic eto start out with a baseline model with pdq value combinations obtained from the auto-correlation plots. This model at most times has no tuning applied. The subsequent models shouls however have to undergo some tuning so as to avoid the problem of overfitting which is pretty common in most time series models.

The subsequent model employs the use of the **auto-arima** model to find the best pdq value combinations with the aim of minimizing the **AIC** csores. The lower the **AIC** scores, the better the **ARIMA** model. It is however important to use the **TimeSeriesSplit** as it splits the data into various folds and seeks to find the optimum **AIC** scores within each fold.

Challenge: The values for the best possible combinations may be overwhelming because it requires iterating through every possible combination for each fold and finding the conspicuous values with the lowest scores.

After training the **Auto Regressive models**, we moved on to our third model which is the **Prophet** model which has exhibited great results in the recent past. The Prophet model is the ideal model because of the concept of adjusting the parameters to capture the seasonality in the time series data.

The zillow data had a major problem regarding home prices when the great 2008 financial crisis happened fueled by the risky mortages given to low income earners. This piled up debt an the bubble burst when th edebt was no longer fiscal and bearable. In future anticipation of such an event, occurring, the Prophet model is ideal and this is due to the easier adjustments of the seasonal component specifically **Seasonality trend, holiday trend** and **Regressors** addition to capture every other external factors driving the target higher or downwards.

Use of an **ETL** pipeline like method came in handy because of the simplification of feeding any data for prediction provided it has the datetime columns and the target columns.

Challenge:

- Using the class without splitting the data using the **TimeSeriesSplit** results to a mismatch between the real and forecasted values. This results to a difficulty of evaluating the whole pipleine. It is therefore necessary to go through the documentation [https://facebook.github.io/prophet/docs/non-daily_data.html] which comes in handy and clearly explains how to do forecatsing on various seasons.
- **Time** taken to get the average values of the metrics to evaluate our model is long. This is because of the use ot the **TimeSeriesSplit** which uses a cross-validation kind of approach to get the optimal results.

The first ETL pipeline chains everything from data ingestion, transforming, training and evaluating the model. It takes in resampled data obtained after transforming the data from wide to long formart. The data is resampled on a monthly basis and contains values of the average house prices for every month. Should the investor like to know the average house values in the near future, the model comes in handy.

The second **TimeSeriesPipeline** however takes in data filtered from the specific ZipCodes and there forecasted values

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● **Jupyter Notebook** 100.0%

Suggested workflows

Based on your tech stack

Python Package using Anaconda Configure

Create and test a Python package on multiple Python versions using Anaconda for package management.

Pylint Configure

Lint a Python application with pylint.

SLSA Generic generator Configure

Generate SLSA3 provenance for your existing release workflows

More workflows Dismiss suggestions