Seas The Data

Applied Scientist SX2 Take-home Challenge

An algorithm to predict the future of a time series data set.

Files

Important files are:

- **Analysis.py** - Main analysis engine
- **Context.py** - Helper class for Analysis.py
- **Design.pdf** - additional, optional document (this one)
- **Executive summary.pdf** - 200 word limit Executive Summary
- LICENSE - unchanged
- **Metadata.py** - Helper class for Analysis.py
- Model.py - unchanged (preserved)
- **Model_modified.py** - My modifications with the SARIMAX modelling
- Modelling.ipynb - unchanged (preserved)
- **Modelling_modified.ipynb** - My modifications - (I don't think there actually are any)
- **ParseSplit.py** - Helper class for Analysis.py
- README.md - unchanged
- Technical Summary.pdf - 500 word limit Technical Summary
- **Workbook.ipynb** - Demonstrates use of Analysis Engine
- data/ - data: csv files, figures

Approach

The dataset can be decomposed into more elemental components. Brainstorming on this, These are possible approaches:

- **Population models** - treat the sales and redemptions as coming from a population with a propensity toward that behaviour (travelling on a ferry). Population can be decomposed into age groups, genders, racial identity, political leaning, whatever would make the most sense.
- **Temporal decomposition** - break up the total dataset into models based on trends at different timescales, such as sub-day, weekly, annual, overall, and noise.

Population modelling

For this type of data (ferry ticket sales / redemptions) population modelling seems less useful. But if the subject were more strongly tied to population, such as health-related concepts, the population model would likely be a better approach.

For this project, I will not use population modelling; I will use temporal decomposition.

Temporal decomposition

The total signal can be broken up by models operating at different temporal scales.

- **Sub-day timescale** - 24 hours long, 15 minute resolution - typical variation throughout a full day.
- **Weekly timescale** - 7 day long - daily variations in a typical week, starting on Wednesdays.
- **Annual timescale** - 52 weeks long - weekly variations in a typical year, excluding holidays.
- **(Special) holiday models** - Take special account of holiday periods.
- **Overall trends** - overall trend, based on population sizes, economic activity, historic trends.
- **Noise** - from the dataset derive an expected noise level, and apply it as uncertainty in the prediction.

Method

Holidays

Since holidays occur inconsistently through the year, it would be nice to remove holidays from the dataset and model them separately, as indicated above, but given time constraints, I will omit the holiday model and return to it if I have time.

It turns out I did not have time.

Prediction

The basic governing equation here is:

$$P \approx T \times P_A \times P_W \times P_S$$

where:

- P is the predicted value (sales or redemptions)
- T is the overall trend (normalised)
- P_A is the annual profile (normalised)
- P_W is the weekly profile (normalised)
- P_S is the sub-day profile (normalised)

Uncertainty

Uncertainty can be modelled:

- Using statistical distribution, detected during the creation of each *profile*.
- Breaking the data into two pieces: the training set and the testing set.
    - *Training set* - used to establish profiles and coefficents for the prediction functionality
    - *Testing set* - used to assess the quality of the prediction tool, and assess its uncertainty

User-option - split date

I should have used the `TimeSeriesSplit` functionality. What I did, instead, was manually split the dataset into a 'training' aspect (earlier) and a testing aspect (more recent data). I added some rich user-interface functionality into choosing where the split occurred:

- **split_date** : a timestamp where the split occurs (currently implemented)
- **split_portion** : a fraction 0 .. 1.0, with data[0] .. data[size x split_portion] is training
- **split_test_interval** : a timespan, (seconds / weeks / days, ec) where the test data is the most recent data points, going back in time by the given split_test_interval.

Career Transition

This project marks an exciting transition in my career—from high-performance and scientific computing to data science. I've been studying intensively, building on a foundation of 17 years developing complex models for demanding, real-world applications. The data modelling side of this challenge felt natural to me, and I'm rapidly growing into the statistical and machine learning aspects of the field.

While my solutions may not be fully mature yet, I hope they demonstrate potential. I bring a strong background in engineering-grade software, analytical thinking, and end-to-end problem solving. I would be grateful for the opportunity to present my ideas in person and discuss how my evolving skillset can contribute to your team's goals.

AI Use Disclosure

Nearly every line of code was written exclusively by me. As the deadline neared and I was frustrated by stack overflow, I asked chatgpt for syntax help into the pandas API, but it was minimal. Additionally, I asked chatgpt for help coming up with a funny name for this project. Seas the Data is its fault. I also did not notice the unfortunate acronym before committing.

Conclusion

Thank you for taking the time to consider my application.