Leading Indicators

First Report

Keaton Markey

26 October, 2023

Contents

Introduction **1**
Project Aim: . . . 1

Methods: **2**
1. Digest & Prepare Data . . . 2
2. Find Leading Indicators . . . 2
Data Sources . . . 2
Good Predictors . . . 2
3. Create Predictions . . . 3
Linear Model . . . 3
Decision Tree . . . 3
Random Forest . . . 3
ARIMA . . . 4
GRU . . . 4
LSTM . . . 4

Results **4**
Leading Indicators . . . 4

Discussion **5**
COVID-19 . . . 5
Evaluation . . . 6

Indicator Index **6**

Credit **7**

Introduction

Project Aim:

To find external indicators that provide reliable insight into business performance for a local car dealership, and leverage those indicators to predict future business performance.

The findings of this project have been compiled into a web application. A demo is available at the link below.

Demo app

Methods:

1. Digest & Prepare Data

Raw sales data, cleaned and summarized by month. Potential indicators were grouped by data source and summarized by month. They are refreshed each month.

2. Find Leading Indicators

Data Sources

For a sales-oriented business, the most important source of revenue is selling the product. For a consumer, buying a car is different than other purchases (such as buying food at the grocery store) in a few key ways:

- Buying a car is a big financial commitment
- A new car is not always a necessity
- Consumers have unparalleled choice of product

There are also similarities between buying a car and groceries:

- There is increasing competition from online shopping businesses (Carvana, AutoNation)
- A long and traceable pipeline from production to consumer

According to recent research (Cox Automotive Research & Market Intelligence), people spend an average of 89 days to buy a car. And, the average car is owned for 8 years (The Zebra). Buying a car occurs when two worlds meet with perfect conditions– the brand engineering the right product, creating advertising, and distributing, while the consumer is saving, researching options, shopping around, test-driving, and then buying. Buying a car is not just about money, but about timing. Thus, this project captures not just a variety of economic indicators, but also behavioral indicators that may precede buying a car.

To find potential candidates, I consulted sources used by other similar projects, current market research, and suggestions. Leading indicators were selected from a pool of stocks closures, volumes, economic measures, Google Trends information. Finding and extracting good indicators of business performance is also a necessary step to create accurate and reliable predictions.

Good Predictors

A "good" predictor of business performance means a few things in the scope of this project: - On one hand, a good predictor may mean any set of data that is highly correlated with business performance. When the predictor rises, business booms, and when the predictor falls, business slows down. - A good predictor may also be a set of data that is important for a model to predict business performance.

In other words, the leading indicators in this project are identified by both simple and more complex mathematical methods. Some are better interpreted by people, and some are better interpreted by one of the algorithms used to predict business performance. Where applicable, each indicator is given a score based on their importance to a model and correlation to business performance which you can view in the app.

Identifying leading indicators can be translated to a machine learning problem called *feature selection*. The feature selection processes used in this project sometimes combine simple statistics and machine learning methods.

After compiling a list of potential candidates, candidates that met the following criteria were excluded during the feature selection process:

- Updated less frequently than every month
- Not released before the 5th of the month
- Not available for the duration of available data (2010-present)

Candidates were then modified for the appropriate prediction interval (originally six months). Each indicator and it's modified subsidiaries were grouped together, and the single feature with the lowest Variance Inflation Factor (VIF) was passed on as a *feature*.

During the prediction process, machine learning techniques were employed to determine which indicators were important to each model. These methods are further described in the next section about models and feature selection.

3. Create Predictions

After identifying candidates, each of the following models was used in combination with feature selection techniques so that, when interpreted in concert, they may give diverse yet probable insights into future business performance.

a) Linear Model
b) Decision Tree
c) Random Forest
d) ARIMA
e) GRU
f) LSTM

Each model utilizes different methods to establish a relationship between the candidates and business performance. Thus, they must be interpreted differently. Below is a detailed description of each model, some guidelines for interpretation, and additional considerations.

Linear Model

The linear model establishes a linear relationship between each feature and the output. This is the most simple method and typically very effective. The model won't be great at recognizing yearly or quarterly recurring trends, but can be easily quantified mathematically and serves as a great basis of understanding and improvement.

Feature selection was performed with repeated 5-fold cross-validation through Lasso regression (L1 regularization), optimized for mean-squared-error. The leading indicators used in the final version of this model are best described as those with the strongest linear relationship to the target variable. While this model doesn't capture more complex relationships, it discovers each leading indicator with a robust method that can be easily mapped back to the target with a linear function. Feature importance was calculated using the absolute value of the normalized coefficients.

Decision Tree

A decision tree is another simple method of prediction that is slow-moving and resistant to outliers. By writing some binary (yes/no) question for each of the inputs, this model combines the answers into a prediction. For example, if the GM stock closes below 34.00 USD, a decision tree model might predict an decrease of 10 in the target variable. This model works poorly for changes on a small scale, but longer-term changes may be able to be detected early.

Feature selection was performed by trimming the depth of the tree with 5-fold cross-validation to optimize mean-squared-error. Despite its simplicity, this model works well to predict future changes. It can also be understood easily and quantified mathematically. The leading indicators used in this model can be interpreted as those with the greatest impact on minimizing the residual error of the model during cross-validation. Only a few leading indicators are selected by this method, and their importances were calculated using the Gini importance according to existing documentation (sklearn).

Random Forest

A random forest is a collection of hundreds of decision trees, each composed of different features, that each take votes on estimating the target variable. This method is extremely effective and uses the best estimates

from the best predictors.

The architecture of the model was set using 5-fold cross-validation on maximum tree depth and the minimum number of samples per leaf to discourage outlier bias. Feature selection was performed by calculating the permutation importance of each feature on a validation set, and running the model again with only features that had higher than mean feature importance. Features important to this model can be described as those that most often contribute to creating effective decision trees. This model is more complex, and while it can be translated into a series of decision trees, its less clearly able to quantify a direct relationship between each leading indicator and the target.

ARIMA

An Auto-Regressive Integrated Moving Average is a classic model of time-series prediction. Instead of using leading indicators, this makes a prediction based on linear relationships to past observations of the target variable (autoregression) and a calculated moving average. The performance of this model depends greatly on seasonality. This model is a benchmark for the success of this project. If the other regression models cannot outperform it, then it may not be beneficial to consider leading indicators at all.

All features in this model are monthly lagged versions of the target variable, up to 12 months.

GRU

A Generalized Recurrent Unit (GRU) is a basic type of neural network designed to place weight on past observations. Instead of one-shot optimization, neural networks are "trained" incrementally on hundreds of copies of the data and over time, come to some understanding of the relationship between the leading indicators and the target.

After its first training pass, this model selects the top 10-20 most important features and is trained again using only those. Feature importance is calculated with the integrated gradients. Because of the nature of this model and the amount of available data, it was heavily throttled by limiting the amount of training data it sees and the total number of parameters. While they usually achieve great scores, these models are difficult to understand, and thus don't offer a simple 1-1 relationship between input and output. Integrated gradients can give an idea of how important each leading indicator may be to the model, but its difficult to determine exactly what that means.

LSTM

A Long-Short-Term Memory is a deep neural network that is well-suited to the problem of time-series prediction. This is the most complex model used in this project. Being a bit larger than a GRU, this model was also throttled to reduce risks associated with using overfitted models.

This model uses all selected features and does not perform its own feature selection.

Results

Leading Indicators

As discussed, its important to remember that leading indicators are identified by a few methods ranging in interpretability and usability. In general, leading indicators that are identified by simpler, interpretable methods like a correlation coefficient aren't necessarily as important in predicting business performance. The inverse is also true: leading indicators that are highly important to a model such as the LSTM may not be well-understood or useful to people outside of its importance to the model.

Below is a table of all candidates and their importance with respect to each model.

Discussion

One of the greatest concerns when creating machine learning models is called *overfitting*. This is a phenomenon that occurs when the model fits the training data very well, but performs poorly when asked to predict new data. For example, the models are able to establish relationships that exist in data from the past, but don't generalize to make accurate predictions about the future.

Every kind of model is prone to overfitting, and this project is especially at risk of encountering this problem for two reasons.

The first reason is the shape of the data. We are working with information from around 165 months. More longitudinal data would improve both prediction power and the accuracy of the feature selection process. With hundreds of similar candidates to choose from, it's ideal to have a number of observations (months) that exceeds the total number of candidates. This would require more data: - Pre-2010 from the shop - Pre-2010 from all external sources that may not be accessible or useful for a variety of reasons

Fortunately, the general shape of the data (known as n « p where n is the number of months and p is the total number of candidates before selection) conveniently lets us accomplish both goals of the project by requiring a robust feature selection process that: - reduces colinearity (duplicate information) - decreases overfitting - and presents a nice way to quantify true leading indicators

The second reason is the types of models used in the project. As mentioned, a few of the models were heavily simplified to reduce the risk of overfitting. Typically, the more complex the data, the more complex the model. While complicated models can produce extremely accurate predictions on a training set, predictions can be really inaccurate on new data. It's important to balance scores between the training and test set, especially with the more complex models. Cross-validation is a common solution to this problem as well, and has been employed by most of the models.

With so many candidates for leading indicators, more monthly data could dramatically affect the output of feature selection and model performance tests. Often methods like bootstrapping and sampling are used to increase the number of observations, but with time series data there is little that can be done without disrupting the sequential nature of observations. We could also considered other intervals, such as weekly or bi-weekly, but would likely introduce more variance and restrict candidate selection even further.

Additionally, with all the data the models use, it is possible that there is more economic information, stock trades, or other metric that may prove useful. More features may still be necessary to improve prediction of business performance and reduce overfitting. In the future, including more monthly observations and more feature candidates is necessary for improving this project.

COVID-19

COVID-19 put a pause on many businesses. This is a problem for two reasons. First, only two sales occurred in the month of April 2020, which is far removed from the rest of the data–normally, sales hover between 200 and 400 per month. Such a big deviation affects machine learning algorithms greatly, and instead of extracting relationships across the whole time period to improve their score, they would focus on the one or two months that contribute to a high or low score, such as April 2020. Removing this outlier is an option, but it would disrupt the model's usage of sequential data and reduce the chances that we would be able to predict an event like this should it happen in the future.

The second and more serious issue is how the pandemic disrupted the economy. While it had a direct impact on sales for this business, it also disrupted the business relationship between groups across the world. Many organizations underwent a period of rapid change, where few emerged operating exactly how they did before April 2020.

The challenge with trying to predict business performance during this time is that whatever relationships existed between businesses before the pandemic have been changed or dissolved entirely. Currently, the models are primarily trained on data before the pandemic to predict the business for all future months.

Without a reliable way to test relationships that emerged after April 2020, we must make the assumption that the landscape of candidates and their relationship to this business has not changed since then.

Evaluation

Generally, two metrics were used to evaluate the prediction models: mean-squared error (MSE) and coefficient of determination (R^2).

While many candidates were not included in the prediction phase, either due to collection irregularities or feature selection, they still might be useful in interpreting business performance. It is not correct to say that these candidates aren't important at all, but we can say that they are likely less important than the chosen leading indicators

Indicator Index

```
# # years on right, months on left?
# #
# nmonth_metric <- "Back Gross"
#
# estimates_for <- paste0(month(nmonth, abbr = FALSE, label = TRUE), ", ", year(nmonth))
#
# # last 2 months, last 2 years
# predate <- as.character(c(as.character(rev(month((nmonth) - months(1:2), label = TRUE, abbr = FALSE)),
#                           estimates_for,
#                           (year(nmonth) - 1):year((nmonth) - years(2)))) # last 2 years
#
# # group 2 insert for next month
# thismonthinsert <- list(group = 2,
#                         tframe = estimates_for,
#                         "Avg. Cash Price" = NULL,
#                         "Total Gross" = NULL,
#                         "Back Gross" = NULL,
#                         "Front Gross" = NULL,
#                         "New Sales" = NULL,
#                         "Used Sales" = NULL)
#
# # previous months + years for next month
#
# nmonth_df <- bind_rows(KDAc %>% # last 2 months
#                     dplyr::filter(date >= pmonth - months(1), date < pqmonth) %>%
#                     dplyr::mutate(group = 1, tframe = as.character(month(date, label = TRUE, abbr = FA
#                   KDAc %>% # last 2 years
#                     dplyr::filter(month(date) == month(pmonth) + 1, year(date) >= year(pmonth) - 2, da
#                     dplyr::mutate(group = 3, tframe = as.character(year(date)))) %>%
#   group_by(group, tframe) %>%
#   summarise(`Avg. Cash Price` = mean(cash_price, na.rm = TRUE),
#             `Total Gross` = sum(total_gross_profit, na.rm = TRUE),
#             `Back Gross` = sum(back_gross_profit, na.rm = TRUE),
#             `Front Gross` = sum(front_gross_profit, na.rm = TRUE),
#             `New Sales` = sum(ifelse(nu == "NEW", 1, 0)),
#             `Used Sales` = sum(ifelse(nu == "USED", 1, 0))) %>%
#   bind_rows(thismonthinsert) %>%
#   dplyr::mutate(tframe = factor(tframe, levels = predate, order = TRUE)) %>%
```

```
#   pivot_longer(cols = -any_of(c("group", "tframe")), names_to = "class", values_to = "value") %>%
#   group_by(group, class) %>%
#   dplyr::mutate(a = ifelse(group != 2, coef(lm(value ~ as.numeric(tframe)))[2], 0),
#                 b = ifelse(group != 2, coef(lm(value ~ as.numeric(tframe)))[1], 0))
#
# if (!(nmonth_metric %in% nmonth_df$class)) {
#   warning("Enter a valid class")
# }
#
# # prediction function
# predrectf <- function(facet) {
#   g1 <- predict(lm(value ~ as.numeric(tframe), data = nmonth_df, subset = (group == 1 & class == nmon
#   g3 <- predict(lm(value ~ as.numeric(tframe), data = nmonth_df, subset = (group == 3 & class == nmon
#   return(c(group1 = g1, group3 = g3))
# }
#
# # get bounding box
# predrect <- predrectf()
#
# nmonth_df %>%
#   filter(class == nmonth_metric) %>%
#   ggplot() +
#   geom_point(aes(x = tframe, y = value, fill = factor(group), shape = factor(group)),
#              size = 5,
#              color = "black",
#              inherit.aes = FALSE) +
#   scale_shape_manual(values = c(25, 2, 22)) +
#   geom_abline(aes(slope = a, intercept = b, color = factor(group)),
#              lty = 2, size = 2) +
#   theme(axis.text.x = element_text(angle = 15),
#         legend.position = "none") +
#   geom_rect(aes(xmin = 2.8, xmax = 3.2, ymin = min(predrect), ymax = max(predrect)),  # should always
#             stat = "identity",
#             data = nmonth_df %>%
#              filter(group == 2),
#             inherit.aes = FALSE,
#             alpha = 0.1,
#             color = "black") +
#   scale_y_continuous(labels = comma) +
#   coord_cartesian(ylim = c(min(predrect) * 0.7, max(predrect) * 1.3)) +
#   labs(title = paste0(nmonth_metric, " Estimate for This Month"),
#        subtitle = paste0(month(repon, abbr = FALSE, label = TRUE), ", ", year(repon)),
#        x = "",
#        y = "Value")
```

Credit

Author: Keaton Markey Version: 1.0 Date Revised: 10/11/2023