Brian Scott
(307) 690-4799
bscottwyoming@gmail.com
[redacted] homework
Submitted [redacted] 2021

Business Summary

As shown in Table 1 below we choose an ensemble of models to predict total delivery duration and select a multi-layer perceptron or artificial neural network (ANN) with data columns reduced to 13 from 90. ANN is non-linear and shows improved prediction errors over linear models. As shown in Figure 1 below we note the most important features are order-related (size, price), followed by some store details (how it received the order, estimated time from store to delivery location), followed by tertiary order and store data; [redacted] capacity, time of day, day of week, and food-type appear to be least important. We used a relatively scaled down ANN model because of time, simplicity, and computation power. Also for simplicity, we decline to enforce an asymmetric loss function to training. Presumably computation power and complexity are increased/tolerated in a commercial setting and we expect model performance to improve accordingly.

	Lasso Regression	Ridge Regression	Support Vector Regression (sigmoid kernel)	Neural Network (full features)	Neural Network (lasso culled)	Neural Network (PCA culled)	Neural Network (PCA culled w k-fold cross validation)
Mean RMSE (mins)	17.4	17.5	16.7	13.4	13.4	13.4	13.5
Mean RMSE (secs)	1044.6	1047.7	1001.6	801.1	801.1	801.1	808.2
Sigma RMSE (secs)	32.8	31.8	69.8	HM	HM	HM	8.6
Features	35	50	45	90	52	13	13

Table 1: We first test lasso and ridge linear regression models then support vector regression with a sigmoid kernel. All three models were evaluated with 5x2 nested cross validation. That is, we selected the optimal hyperparameter and PCA feature extraction via GridSearchCV(cv=2) outer loop then validate with cross_val_scores(cv=5) inner loop, collecting mean and sigma of RMSE. These predictions were greatly improved with ANN models. We used a small 20x20x20 ANN model for three different feature sets using the holdout method. The smallest feature set appeared to be as accurate at the largest. We slightly expanded ANN model to 50x50x50 and deployed 5-fold cross validation again noting mean and sigma RMSE. We select this model to predict an unlabeled data set.

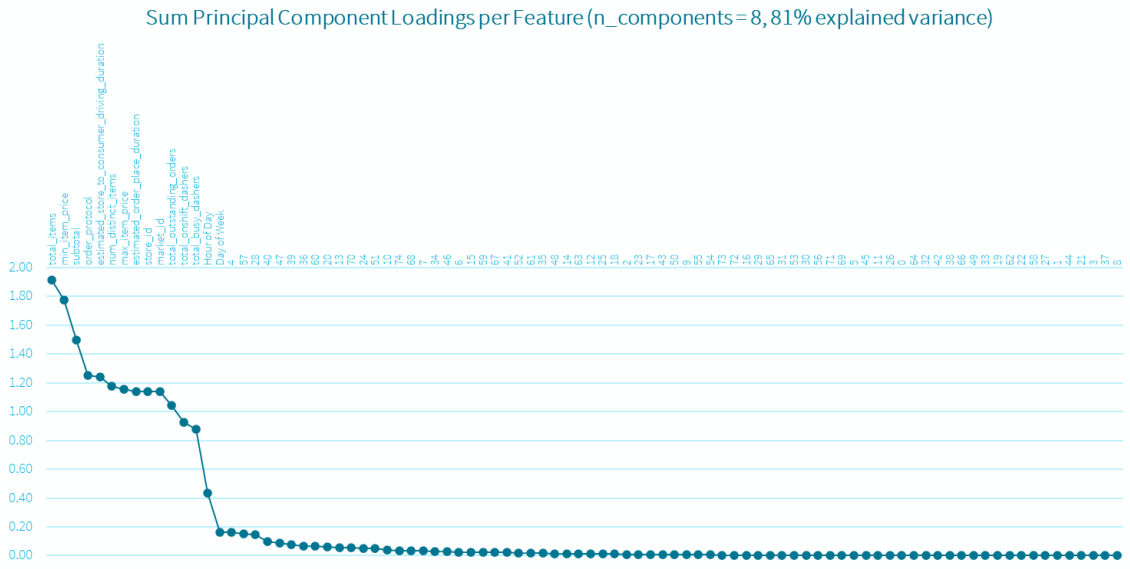

Figure 1: Noting of course PCA-space columns are the PCs and the rows are the features, we take an inelegant but systematic approach to reducing features by eigenvector loadings. We have determined 8 PCs explain 81% of the standardized feature matrix variance, we sum the absolute values of the loadings across each feature rows, than rank those sums. Above we can see multiple kinks implying tiers. For the final 13 features in our NN model we impose a cutoff below 0.20 which eliminated time of day, day of week, and all food types (indicated above by numbers since we deployed one-hot encoding during preprocessing). Note the time/day features were engineered from the original dataset but ultimately appear unnecessary.

Brian Scott
(307) 690-4799
bscottwyoming@gmail.com
[redacted] homework
Submitted [redacted] 2021

Feature Analysis

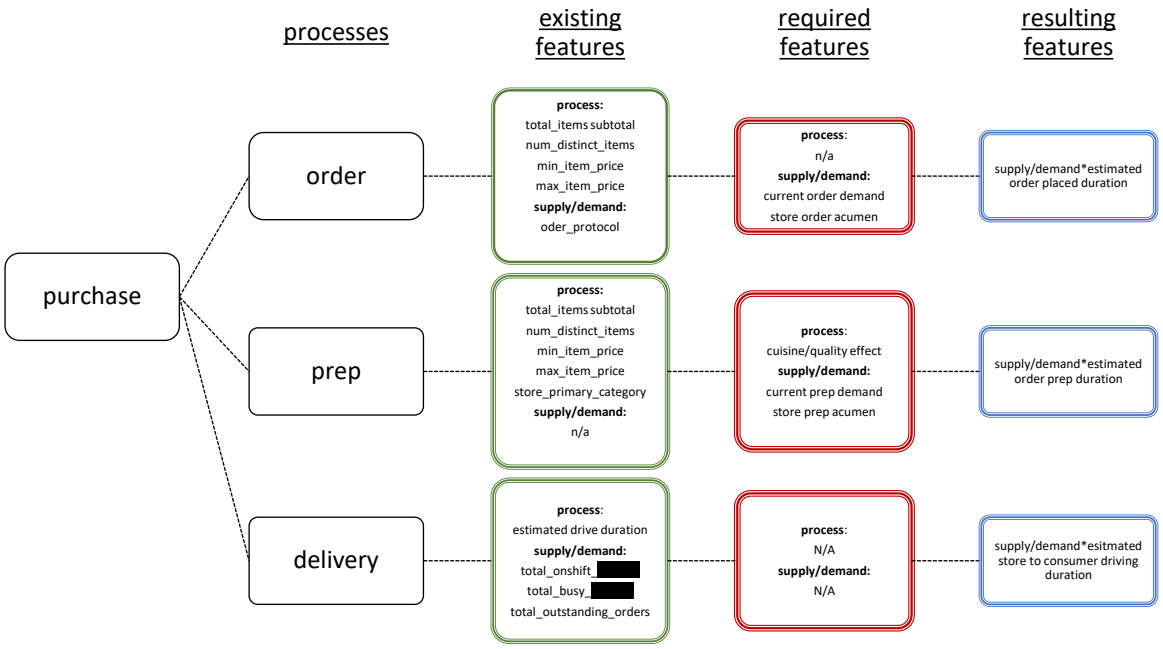

Figure 2: summary of existing, required, and possible resulting feature set

A single customer purchase appears to cross four fields of influence. Order processes, prep processes, delivery processes, and all three are set in some kind of supply/demand context.

- Order processes: time a customer sees a purchase confirmation to the moment store begins food prep. This process appears to be captured by existing order features, order protocol feature, and estimated order place duration feature.
  - Supply/demand context: there likely exists data indicating how many delivery orders are being sent to a store at any given moment and a history of the store's ability to process these orders.
- Prep processes: time food prep begins to the moment it is available for pickup. Highly relevant data is missing from store features and an estimated prep duration is missing entirely. Enhanced store features could include a ranking for which cuisines take longest to prepare plus some kind of food quality rating i.e. prep delta between the same items of different quality.
  - Supply/demand context: general popularity of store as ascertained from sources like Yelp, current busyness of store as ascertained from sources like Open Table, store history that could estimate store's general prep acumen.
- Delivery process: time the order is available for delivery to the moment it is delivered to customer. This appears to be captured in existing [redacted] capacity and outstanding order features. Presumably drive duration data is being optimized by navigation sources like Google Maps. Date for this process appears to be developed an easily obtained.
  - Supply/demand context: appears to be a subset of existing [redacted] and outstanding order features.

Store-level data appears to be the most underdeveloped. Obviously stores can be characterized across many dimensions. It seems likely that between third-party data and a store's track record it's possible to illuminate the space between when a store receives an order until a [redacted] picks it up. This area requires the most development it was not in the exiting data set. Once developed we might be able to simplify a total delivery duration model with the following highly generalized pseudo formula, where β is a supply/demand coefficient and X is a process time estimate:

$$Total\ Delivery\ Duration = \beta_{order}X_{order} + \beta_{prep}X_{prep} + \beta_{delivery}X_{delivery} \quad [1]$$

Brian Scott
(307) 690-4799
bscottwyoming@gmail.com
[redacted] homework
Submitted [redacted] 2021

Selecting/Deploying the Production Model

Per Ashmore et al. [1] when comparing new models to existing production models assessing performance comes down to two key decision chokepoints: model verification and model deployment.

- Model verification
  - Somewhat easy for analysts to build an error function alongside output and rank order RMSE. Even more so when headcount provides for barely mitigated model proliferation. Certainly we need a framework to prevent metastasizing workloads and drifting from business purpose. Ultimate performance metrics must tie to business value via informed KPIs. This likely will require input from non-data teams (e.g. finance, marketing) and should define model verification context from the outset. Model verification can also be affected by regulatory requirements e.g. Basel and CCAR for bank holding companies.
  - Alas, quants are well versed in test-based verification. Here we ensure trained models generalize well to new data. In this writeup, we minimized RMSE across validation and test sets. Moving forward we need to obtain out-of-sample results. Out-of-sample results can come from simulation or perhaps "paper trading" live data. If results are consistent and superior to existing models we run our new models alongside existing models for some kind of live trial period. Again if results are consistent and superior perhaps we have arrived at the final accuracy/complexity tradeoff: if simplicity only costs us seconds, and that simplicity provides better accessibility and lower expenses, then we should choose simplicity; conversely if complexity subtracts tens of minutes from prediction errors and deployment is feasible, then certainly complexity adds superior value.
- Model deployment
  - We again cite Ashmore then Paleyes et al. [2] as it relates to model deployment concerns.
    - Integration. This idea includes infrastructure required to run models in a way that can be consumed and supported. This includes systems engineering but also ideas like code reuse, ML-specific engineering issues that may contrast with convention, and the required mixing of data and engineering teams.
    - Monitoring. Technical items in this realm can include prevention/identification of feedback loops as the model continually updates; outlier detection early in the data pipeline process, especially for live analysis; and the concept of custom design tooling, where each deployment is inherently unique and requires newly created solutions.
    - Updating. We are required to prevent drift and adhere to accuracy standards. Here we could be retraining models and obtaining better fits with near-live or highly recent lookbacks. We could also be blending (ensemble) recent fits with more robust fits that span the entire length of datasets.
  - Model deployment concerns again lead us to a tradeoff decision point. Per Palyeyes, Holder et al. [3] details an ABB search team being overwhelmed by a multi-layer ANN project ultimately abandoning such a lofty foray and scaling down. This led to earlier deployment which provided an effective development setting to incrementally add complexity. Also per Paleyes, when referencing Ashmore's workflow spanning data management, model learning, model verification, model deployment, and exogenous concerns—model deployment is the most fertile ground for obstacles and setbacks.

References

[1] Rob Ashmore, Radu Calinescu, and Colin Paterson. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *arXiv preprint arXiv*:1905.04223, 2019.

[2] Paleyes, Andrei & Urma, Raoul-Gabriel & Lawrence, Neil. (2020). Challenges in Deploying Machine Learning: a Survey of Case Studies.

[3] Malay Haldar, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C. Turnbull, Brendan M. Collins, and et al. Applying deep learning to AirBnB search. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jul 2019.