MGT 6203 Group Project Final Report

Bet on Better Forecasts
Leveraging Customer-Level Insights for Enhanced Financial Prediction for Racing and Wagering Western Australia

Team 14

Reiko Takizuka

Duc Nguyen Hong

Chandana Mudunuri

Dayoon Sug

Michael Sullivan

Abstract

Customer Value is a key metric for businesses to gain insights into the effectiveness of their products, services and overall customer experience. Predicting customer value helps businesses estimate the total revenue they can expect from a customer. It allows for prioritization of valuable customer segments, tailor marketing efforts, and allocating resources effectively. This project is predicting the Customer Value of the TAB's individual wagering customers using machine learning techniques.

Company background

Racing and Wagering Western Australia (RWWA) is the controlling authority for horse and greyhound in the state of Western Australia. It is a governmental body charged with maintaining the long-term sustainability of the racing industry in the region.

Part of the RWWA's remit is to control off-course betting activities for the racing industry, which they do through the **Western Australia TAB, a state-owned company that runs over 300 betting retail outlets and an online** betting platform known as "TABtouch". WA TAB allows customers to wager on racing (horse or greyhound) and other sports (both Australian and international). Each year customers place bets with WA TAB of around AUS $2bn, and WA TAB pays out around AUS $1.7bn to winning bets. Though large, these amounts make WA TAB a relatively small player in the context of the gambling industry.

TABtouch logo

Business Justification and Objective

The data pattern allows accurate turnover forecasting for the whole business using time series models, which is fantastic for financial planning and budgeting or risk management. However, we recognize the potential benefits of customer-level prediction that can be achieved through different supervised machine learning models (since we are not able to apply the same time series model type to predict future customer spending as they are much more inconsistent compared to the aggregated turnover). In the context of this project, we define Customer Value as the total turnover spent by each individual customer. Predicting Customer Value will allow TABTouch to:

- Focus on retaining the customers with the highest long-term value.
- Make better decisions around marketing spending.
- Improve segmentation of customers based on long-term values.
- Improve financial forecasting by understanding the value of existing Customers.
- Help the company with the development of new wagering products.

Research Question

Can the customers' historical wagering transaction data forecast their future turnover?

Initial Hypothesis

We believe customers' past behaviors and betting outcomes have significant impacts on their future spending. Behaviors include how often they play, how much they have spent, how much they have won, or how many tickets they have bought could all be potential factors.

We also believe demographic factors have minimal effects on their future betting expense. To verify this hypothesis, we intend to examine it using regression models.

Data Source

TAB Betting Data : Our primary data source is (anonymized) internal wagering data from TAB: we have data that aggregates daily betting totals for each individual customer across 2021 and 2022.

Each row of this dataset represents the bets made by one customer on a given day. There is a unique customer identifying number, customer attributes and then a breakdown of the customers' betting pattern for that day. Because of the sheer number of bets made on the platform, we are using data that has been aggregated daily rather than including one separate row for each bet made by each customer.

Column	Type	Description
DATE_DIM	DateTime	Date of the transaction
DAY_OF_WEEK	String	Day name of the week of the transaction
BET_ACCOUNT_NUM_HASH	Integer	Customer unique identifier
AGE	Integer	Customer's age as of Wager date
AGE_BAND	String	Customer's age band as of Wager date
GENDER	String	Customer's gender (M, F, U)
TENURE_IN_DAYS	Integer	Number of days since Customer opened the account as of Wager date
RESIDENTIAL_STATE	String	Residential state where the customer resides
FOB_RACING_TURNOVER	Float	Total Bet amount spent on Fixed-odds Racing events
FOB_SPORT_TURNOVER	Float	Total Bet amount spent on Fixed-odds Sports events
PARI_RACING_TURNOVER	Float	Total Bet amount spent on Pari-mutuel Racing betting
PARI_SPORT_TURNOVER	Float	Total Bet amount spent on Pari-mutuel Sports betting
TOTAL_TURNOVER	Float	Total Bet Amount spent on the day
DIVIDENDS_PAID	Float	Total Dividend Amount won by the customer
GROSS_MARGIN	Float	Gross Margin for the Wagering provider
TICKETS	Integer	Total tickets bought by the customer on the day

Methodologies

After defining the business process question, we follow a typical process pipeline of a data science project, including Data Cleaning -> Data Exploration -> Data Modeling -> Model Evaluation.

Data Cleaning

Given that the data was sourced directly from the company, minimal data cleaning was required. Nevertheless, various measures were taken to prepare the data for analysis:

- *Missing data*: There were NAs in 5 columns: AGE and 4 columns related to TURNOVER. The presence of null values in those 4 columns signifies zero transactions. We decided to impute AGE column by filling in the average age of the all customers which is 44.

  Missing value on each column

```
DATE_DIM                       0
BET_ACCOUNT_NUM_HASH           0
AGE                         2669
AGE_BAND                       0
GENDER                         0
TENURE_IN_DAYS                 0
RESIDENTIAL_STATE              0
FOB_RACING_TURNOVER      3773146
FOB_SPORT_TURNOVER      10549527
PARI_RACING_TURNOVER     3800973
PARI_SPORT_TURNOVER     12305088
TOTAL_TURNOVER                 0
DIVIDENDS_PAID                 0
GROSS_MARGIN                   0
TICKETS                        0
```

- *Redundant columns and duplicate records*: DAY_OF_WEEK can be derived from DIM_DATE column, same with AGE and AGE_BAND. These columns are created for the purpose of reporting; therefore, they can be removed. There are no duplicate records in the datasets.

- *Transactions with TOTAL_TURNOVER equaling 0*: These transactions were complemented to other transactions in some regards. They can be deleted to increase prediction model accuracy.

Data Exploration

Regarding our primary research question, we emphasize on daily total turnover value and individual customer turnover analysis (click here for the full analysis).

1. Daily total turnover value:

Figure 1 : Daily Turnover

- *Seasonality:* The data demonstrate seasonality, with the wagering product showing variation throughout the week and differing slightly from month to month. The highest conversion rates happen on Saturdays and dip at the beginning of the week. This pattern stays consistent throughout both years.
- *Trend*: From the 7-day moving average line, there is no significant upward or downward trend throughout the year, except for two contextual outlier points.
- *Outliers*: There are two noteworthy outliers:
    - Melbourne Cup Day: an enormous racing event that takes place on the first Tuesday of November every year, garners significant attention. It attracts a substantial number of customers to place a wager on the event.
    - Christmas Day: Wagering options are limited on this day. Therefore, a significant drop is expected.

2. Individual customer turnover analysis:

- *High variance of total turnover:* In two years, 90% of customers spent under $5000 wagering. We spotted an exponential decay in the distribution. This could greatly affect the accuracy when it comes to a large value. A log transformation could normalize the distribution.

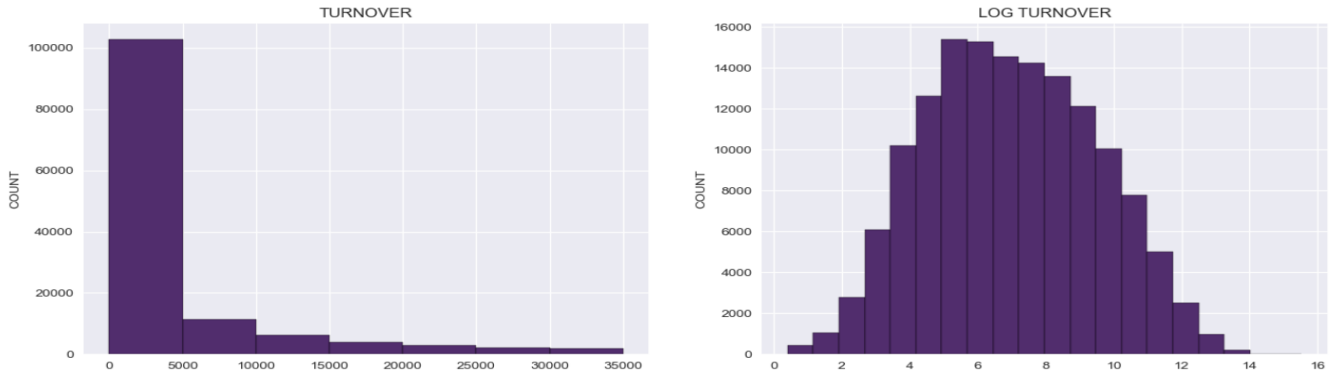


Figure 2 : Turnover Distribution

- *Categorical factors*: Demographic factors can be significant to the prediction models for each categorical group has a different level of turnover.

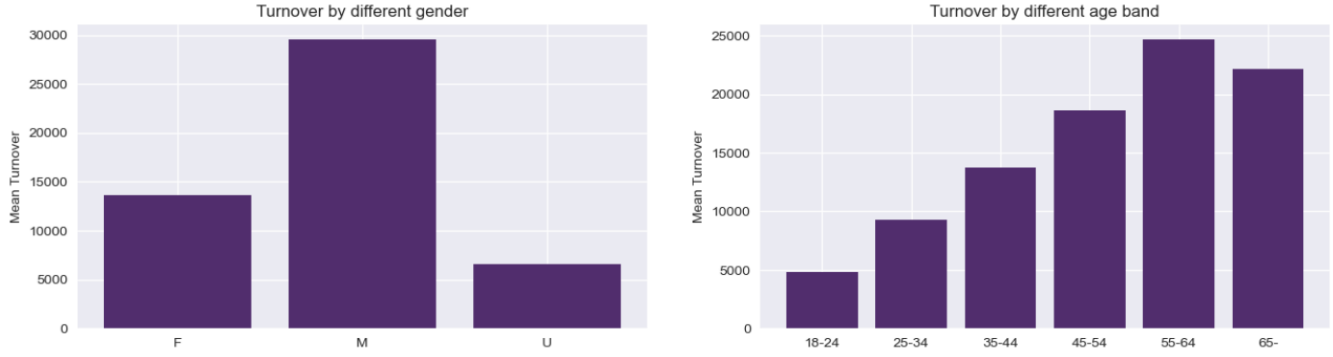


Figure 3: Average Turnover by each group

- *Correlated numerical factors*: Since we are going to build regression models, it is important to collect linear correlated factors to use.

Figure 4 : Correlation heatmap

Data Modeling Processes and Insights

We have implemented these model types to explore our research questions:

- Regression models (Linear & Non-linear Models) (click here for the full analysis)
- Tree models (Decision Tree Regressor, Random Forest Regressor) (click here for the full analysis)
- Time series model (Exponential Smoothing) (click here for the full analysis)
- Probabilistic Customer Lifetime Value (CLV) (click here for the full analysis)

1. Regression models

- **Background**: Regression models are powerful statistical techniques used for predictive analysis. They are widely used for prediction because they offer several advantages. First, they provide a systematic and quantitative approach to understanding and analyzing complex relationships between variables. Second, regression models can be used to make accurate predictions, allowing us to forecast future outcomes based on historical data and patterns. Third, regression models enable us to identify the most significant factors that influence the dependent variable, helping us gain insights into the underlying drivers of the phenomenon we are studying.

  We used the regression model to predict a unique customer's daily average turnover for the next 28 days (about 4 weeks). Regression models can be implemented via 'statsmodels' package in Python.

- **Modeling**:

  a. Data preprocessing and feature engineering:

  We applied the same data-cleaning steps above then created extra columns of past wagering behaviors. The bold one is our dependent variable; the rest are our independent variables.

Column	Description
AGE_BAND	Customer's age band as of Wager date
GENDER	Customer's gender (M, F, U)
RESIDENTIAL_STATE	Residential state where the customer resides
AVG_FREQ_12	Betting frequency of the last 12 weeks [0-1]
RACING_RATIO_12	Racing spending ratio of the last 12 weeks [0-1]
AVG_TURNOVER_12	Average turnover per day for the last 12 weeks
DIVIDENDS_RATIO_12	Dividends paid of the last 12 weeks
AVG_TICKETS_12	Average tickets purchased per day for the last 12 weeks
AVG_FREQ_4	Betting frequency of the last 4 weeks [0-1]
RACING_RATIO_4	Racing spending ratio of the last 4 weeks [0-1]
AVG_TURNOVER_4	Average turnover per day for the last 4 weeks
DIVIDENDS_RATIO_4	Dividends paid of the last 4 weeks
AVG_TICKETS_4	Average tickets purchased per day for the last 4 weeks
AVG_FREQ_1	Betting frequency of the last week [0-1]
RACING_RATIO_1	Racing spending ratio of the last week [0-1]

AVG_TURNOVER_1	Average turnover per day for the last week
DIVIDENDS_RATIO_1	Dividends paid of the last week
AVG_TICKETS_1	Average tickets purchased per day for the last week
AVG_TURNOVER	**Average turnover per day for the next 4 weeks**

We chose the cutoff date to be '2021-05-10' (a Monday) to avoid all the contextual outliers so the training period would be from '2021-02-15' to '2021-05-09' (12 weeks) and the validation period would be from '2021-05-10' to '2021-06-06' (4 weeks).

b. Fit model: We transformed categorical data into dummy variables and fitted a regression model using 'statsmodel' package here's the result:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.615
Model:	OLS	**Adj. R-squared:**	0.614
Method:	Least Squares	**F-statistic:**	3327.
Date:	Tue, 18 Jul 2023	**Prob (F-statistic):**	0.00
Time:	14:33:27	**Log-Likelihood:**	-3.0132e+05
No. Observations:	50106	**AIC:**	6.027e+05
Df Residuals:	50081	**BIC:**	6.029e+05
Df Model:	24		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.5494	2.521	0.615	0.539	-3.392	6.490
x1	3.5341	4.734	0.747	0.455	-5.744	12.813
x2	1.7155	4.489	0.382	0.702	-7.084	10.515
x3	0.4563	4.474	0.102	0.919	-8.314	9.226
x4	1.5038	4.477	0.336	0.737	-7.271	10.278
x5	2.7802	4.483	0.620	0.535	-6.006	11.567
x6	-0.1611	4.489	-0.036	0.971	-8.960	8.637
x7	-8.2794	28.631	-0.289	0.772	-64.397	47.838
x8	0.4634	1.290	0.359	0.719	-2.065	2.992
x9	3.5613	1.040	3.424	0.001	1.522	5.600
x10	-2.4753	1.060	-2.336	0.019	-4.552	-0.398
x11	3.5520	1.546	2.297	0.022	0.521	6.583
x12	-2.0025	1.409	-1.421	0.155	-4.764	0.759
x13	-18.6393	5.360	-3.478	0.001	-29.145	-8.134
x14	-0.7406	1.832	-0.404	0.686	-4.332	2.850
x15	0.3050	0.007	41.668	0.000	0.291	0.319
x16	-0.4793	0.717	-0.668	0.504	-1.885	0.926
x17	0.6850	0.082	8.349	0.000	0.524	0.846
x18	22.5120	6.347	3.547	0.000	10.072	34.952
x19	0.6774	1.556	0.435	0.663	-2.373	3.728
x20	0.2094	0.008	26.290	0.000	0.194	0.225
x21	-0.0179	0.617	-0.029	0.977	-1.227	1.191
x22	-0.0043	0.092	-0.047	0.963	-0.184	0.176
x23	3.6585	3.832	0.955	0.340	-3.852	11.169
x24	1.3990	1.518	0.922	0.357	-1.576	4.374
x25	0.3037	0.005	61.681	0.000	0.294	0.313
x26	0.4471	0.477	0.937	0.349	-0.488	1.382
x27	-0.4922	0.046	-10.608	0.000	-0.583	-0.401

Positive points of the model:

- Good fit: R-squared and Adj. R-squared are relatively high (0.615 and 0.614 respectively) along with high F-statistic score signifying that the model is significant.
- No autoregression (Durbin-Watson close to 2)

Problems with the model:

- Multicollinearity (high Cond. No.): From the correlation matrix, there are a few heavily correlated columns. This could weaken the accuracy of our prediction.
- Insignificant variables: Based on the p-value, we could weed out the statistically insignificant variables. However, we should fix the multicollinearity problem beforehand for the metric to be meaningful.
- Heteroskedasticity: As can be seen in the scatter plot, the variance of residuals gets larger as the value gets bigger. We can attempt non-linear models to lessen this effect.
- Residuals significantly deviate from normality.

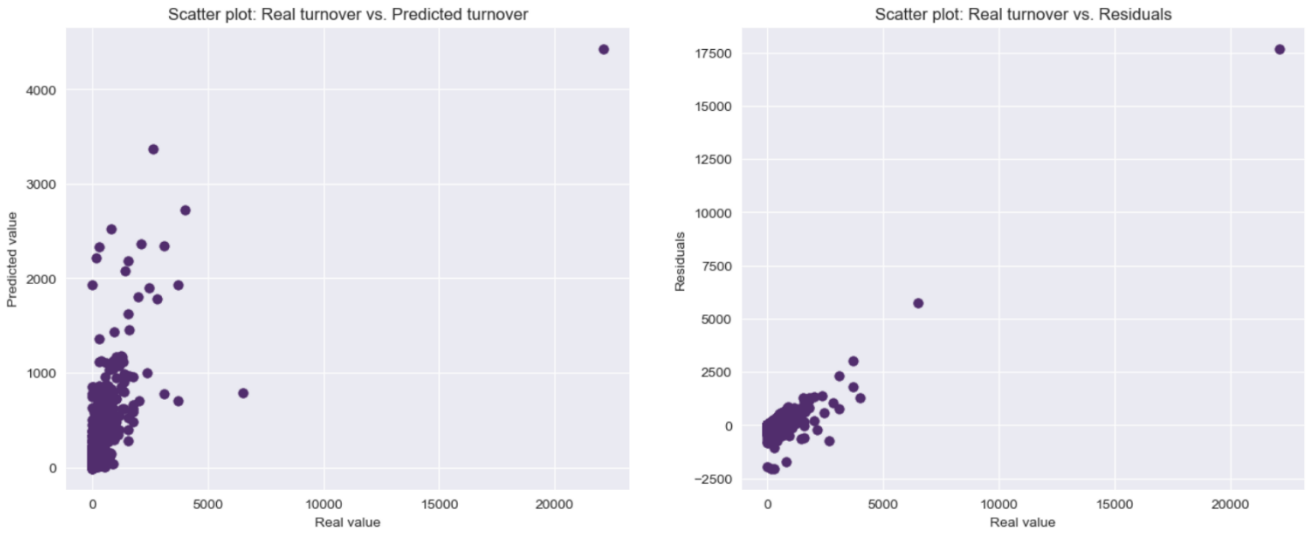


Figure 5 : Scatter Plots of Predictions

c. Model tuning: We applied a variety of tuning methods to fine-tune the model then evaluated each method. We also applied K-fold Cross Validation method to remove selection bias and obtained the average of each evaluation metric.

- Remove categorical data: To test hypothesis one which assumes no relation between categorical factors and future turnover spending, we omitted all demographic variables from the model then fitted it again. The result was very similar.

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.614
Model:	OLS	**Adj. R-squared:**	0.614
Method:	Least Squares	**F-statistic:**	5316.
Date:	Wed, 19 Jul 2023	**Prob (F-statistic):**	0.00
Time:	10:09:46	**Log-Likelihood:**	-3.0135e+05
No. Observations:	50106	**AIC:**	6.027e+05
Df Residuals:	50090	**BIC:**	6.029e+05
Df Model:	15		
Covariance Type:	nonrobust		

- Regularization with LASSO Regression

  Lasso regression introduces a penalty term to the linear regression objective function, which is based on the absolute values of the regression coefficients. This penalty term encourages sparsity in the coefficient values, effectively shrinking some coefficients to zero. Therefore, the least significant variables will be eradicated.

$$\min_w \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha ||w||_1$$

  We used GridSearch to find the optimal alpha which appeared to be 1000. We fitted the model and calculated evaluation metrics.

- Regularization with Ridge Regression

  In Ridge Regression, the goal is to minimize the sum of squared residuals, similar to OLS regression. However, an additional term, known as the ridge penalty or L2 penalty, is introduced to the loss function. This penalty term is proportional to the squared magnitude of the regression coefficients, effectively shrinking them toward zero. Unlike LASSO, the coefficients will not reach zero, but the model can still mitigate the multicollinearity problem effectively.

$$\min_w ||Xw - y||_2^2 + \alpha ||w||_2^2$$

  We used GridSearch to find the optimal alpha which appeared to be 1000. We fitted the model and calculated evaluation metrics.

- Linear Regression with PCA

  Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction and data exploration. It allows us to transform a high-dimensional dataset into a lower-dimensional representation while preserving as much of the original variability as possible. PCA will help to eliminate correlation in the variables and remove insignificant variables.

  We scaled the data then applied PCA transformation. We chose the number of components to be 7 as they retain 90% of the original data then applied a linear regression model and evaluated.

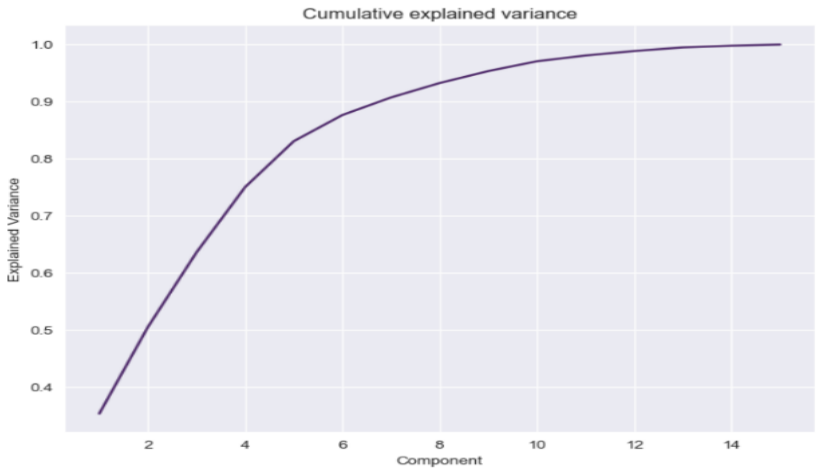


Figure 6 : Cumulative explained variance

- Log-Log Regression:

  We decided to utilize log transformation method to mitigate heteroscedasticity but only with 'AVG_TURNOVER_12','AVG_TURNOVER_4','AVG_TURNOVER_1' columns as they were considered the most significant variables. Then we fitted a regression model and evaluated it.

- **Model Evaluation**: Here is the evaluation data frame. It appears linear regression with PCA returned the best result, but they did not vary that much. The R squared can be considered good. However, the biggest problem is that the variance of the residuals still gets larger as the value gets bigger (figure 5).

Model	**MSE**	**MAE**	**R2**
Linear Regression	15524.48	24.27	0.56
LASSO regression	15552.24	24.86	0.56
Ridge regression	15518.78	23.93	0.56
Linear Regression with PCA	15351.31	24.04	0.57
Log-Log Regression	17486.93	22.64	0.53

2. Tree model:

- **Background**: The Decision Tree Regressor works by constructing a tree-like model where each internal node represents a decision based on a specific feature and a corresponding threshold value. The leaf nodes of the tree contain the predicted output values.

  The Random Forest Regressor is a machine-learning algorithm that belongs to the ensemble learning family. It is designed for regression tasks and is based on the concept of decision trees. Random Forest combines predictions from multiple decision trees to make more accurate and robust predictions.

- **Modeling**: With Decision Tree Regressor model, we extracted the same features as regression models with the same cutoff date ('2021-05-10') and used GridSearch to find the optimal hyperparameters like max depth (12), weight faction leaf (0.03) ...

  With Random Forest, we planted 500 trees with the optimal hyperparameters found above.

- **Model Evaluation**: We used the same metrics as regression models for comparison and here's the result:

Model	MSE	MAE	R2
Decision tree regressor	23492.29	26.91	0.36
Random forest regressor	23389.38	26.41	0.29

The model couldn't cover higher value predictions (need more pruning and a deeper level tree). This is due to the imbalance of turnover distribution. That's where the biggest residuals are.

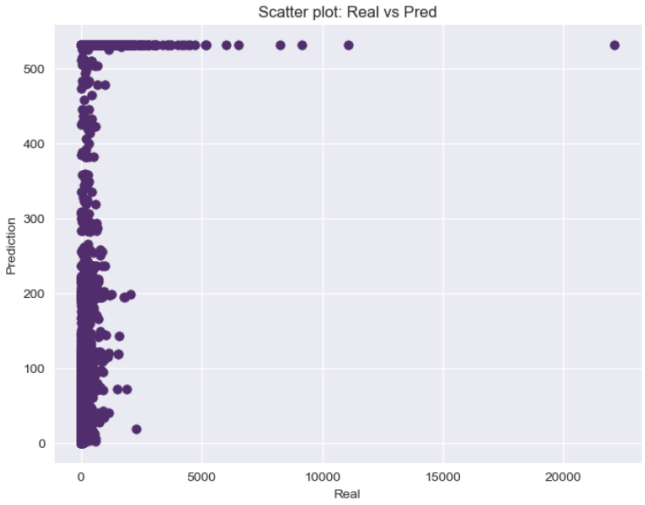


Figure 7 : Tree models real values vs predictions

3. Exponential Smoothing

- **Background**: One of the downsides of the predictive models above is that they do not factor in the time effects. Triple Exponential Smoothing, also known as Holt-Winters' Triple Exponential Smoothing, is a time series forecasting method that extends the concept of exponential smoothing to capture both trend and seasonality in the data. It is a popular technique used in various industries to make accurate predictions for time-dependent data.

  In this case, daily trends can be acting at high variance. Therefore, we chose weekly granularity level which is more stable. We ran a triple exponential smoothing model with a trend factor on all historical data and predicted the next 4-week turnover spending of a customer.

- **Modeling**: We chose the cutoff date to be '2021-05-10' and fitted a distinct model on each customer. The function 'ETSModel' would find the optimal smoothing factor and trend factor from the train set.

- **Model Evaluation**: Here's the result:

Model	MSE	MAE
Exponential Smoothing	29623.28	30.15

4. Probabilistic CLV Modelling

- **Background:** Probabilistic Models offer another way to predict the future betting behavior of TABTouch customers. These models use data on the frequency, recency and value of past transactions and combine them with probability distributions to predict the frequency and monetary value of future transactions (wagers in our case). They offer individualized predictions and are relatively easy to interpret; but they rely on the probability distributions being a good fit to our data, they also do not account for seasonality. In Python, we can use the Lifetimes package to perform this modeling.

- **Modeling**: To model our data with Lifetimes we followed five broad steps[1]:

  (1) Transforming our data into a Recency-Frequency-Monetary (RFM) format and performing a train/test split (by holding out the final period – a month in our case). For each customer this format captures the number of purchases the customer made in the training period (frequency), the age of the customer at their most recent purchase (recency), the average value of their transactions during the period (monetary) and the customer age at the end of the training period. For the monetary value component, we modeled Turnover: the value of bets customers made, not whether they won. We also modeled at a "weekly" frequency to match our modeling elsewhere.

  (2) Fitting a Probabilistic Model on our RFM data to predict the frequency of future transactions (predicting values requires a separate model, covered in part 3). We fit two different types of models offered by the Lifetimes package: (a) a Beta-Geometric / Negative Binomial Distribution (BG/NBD) model - which fits the time between transactions (bets) to a geometric model, assuming it is constant, and (b) a Pareto / Negative Binomial Distribution (Pareto/NBD) model – which fits the time between transactions to a Pareto (exponential) model, assuming it decays over time. The Pareto model is the more complex of the two but can better capture data in some situations.

  (3) Fitting a Probabilistic Model on our RFM data to predict the value of future transactions. We fit a Gamma-Gamma model which assumes transaction values follow a gamma distribution – useful when our customers have heterogeneous transaction values. Note this model is fit separately to the BG/NBD and Pareto/NBD models.

  (4) Combining our two separate probabilistic models to make CLV predictions for each of our customers. The process takes in our RFM data, then takes future frequency predictions from the BG/NBD or Pareto/NBD models and combines them with future transaction value predictions from the Gamma-Gamma model to predict CLV for each individual customer. In our case, we predicted "lifetime" value for the next month.

  (5) Finally, we calculated actual spending by each customer in the next month (our holdout period) and evaluated our predictions against the real values using a range of standard metrics (MAE, MSE, RMSE, MAPE, and R^2).

[1] We created our modelling strategy using info from the Lifetimes Quickstart Documentation with additional information from this Medium post Modelling CLV with Lifetimes.

- **Initial Modelling Outcomes**: For our first prediction, we trained our models on data from January 2021 through November 2022, and made predictions for December 2022. The table below shows our evaluation of each variant's predictions against actual spending.

BG/NBD Model with Gamma-Gamma:	**Pareto/NBD Model with Gamma-Gamma:**
Mean Absolute Error: 603.75	Mean Absolute Error: 602.87
Mean Squared Error: 11967090.00	Mean Squared Error: 11818942.59
Root Mean Squared Error: 3459.35	Root Mean Squared Error: 3437.87
Mean Absolute Percentage Error: inf	Mean Absolute Percentage Error: inf
R-squared: 0.37	R-squared: 0.38

Our metrics show that both our initial models were performing poorly. The MAE scores can be interpreted as our predicted CLVs for the final month being $600 out on average for each customer. Our RMSE scores were significantly higher than the MAE scores, which suggests that we have some very large individual residuals – and which makes sense given the variance we have across customers. The R-squared scores are not terrible for real-world data though less than we achieved with the Linear Regression modeling. Overall, the scores for the two models variant were very similar (see additional chart "CLV-A" in appendix), but we noticed during modeling that the Pareto/NBD was computationally much more expensive so we decided to progress with the BG/NBD model for future refinement.

- **Model Tuning:** We suspected that seasonality might be a primary cause of our poor fit since we are unable to introduce time covariates into our models, but we know from earlier EDA that our data is highly seasonal and that our prediction period (December 2022) follows shortly after some of the largest sports events of the year with large-scale wagering. So we ran our modeling process making predictions for every month from February 2021 through December 2022 and tracked metrics. We let the training period incrementally increase instead of restricting it to only the month prior. However, we didn't notice major seasonal patterns affecting our monthly predictions (see appendix chart CLV-B): in fact, they generally got better over time, particularly MAE, though this is probably due to our training dataset size incrementally growing too.

  Seeing that seasonality might not be the main cause of poor fit we realized the distribution of our real data might be causing our issues. We noticed during model fitting that our modeled distributions weren't good at reproducing frequencies that matched our actual data (see appendix CLV-C): specifically, our models were not capturing a group of high-frequency gamblers very well. The kdeplot below shows the distribution of frequencies in our data: it shows that we have a bimodal distribution with a lot of low-frequency customers and then a second 'cluster' of customers with very high frequencies. This distribution is not a good fit for the BG/NBD or Pareto/NBD models and is causing our models to overestimate values for a lot of the low-frequency customers. We need to deal with this somehow.

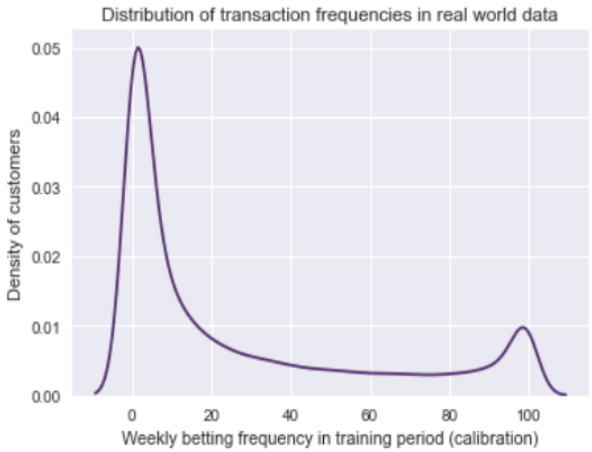


Figure 8 : Distribution of transaction frequencies

To deal with the distribution issue we need to segment our customers so we can model distributions for higher and lower frequency customers separately. We edited our modelling process so that we: split our RFM dataset into high-frequency and low-frequency customers; created a BG/NBD with Gamma-Gamma model combination for each group; predicted CLV values for each group; then aggregated predictions and computed overall metrics. We also needed to decide where to split customers into high/low frequency, so we defined a range of "cutoff" separation values from 10 through to 90 and performed the above process for each threshold. In each case, we trained our model on data for January 2021 through November 2022 and made predictions for December 2022 so we could compare it to our original model.

cutoff_values	mae	mse	rmse	r2
10	216.26	5157104.97	2270.93	-0.11
20	246.42	4965929.95	2228.44	-0.04
30	275.93	4988261.73	2233.44	-0.02
40	307.04	5686479.64	2384.63	0.05
50	334.58	5898738.44	2428.73	0.09
60	361.67	6106255.28	2471.08	0.10
70	399.18	8889858.15	2981.59	0.12
80	436.21	9424508.37	3069.94	0.17
90	484.43	10813159.60	3288.34	0.25

The table to the left shows the metrics we achieved when we introduced segmentation on betting frequency for various segmentation thresholds (see appendix CLV-D for a corresponding graph). We can see that introducing the segmentation has a big impact on our scores. In the best case when the cutoff threshold was 10, we were able to reduce our MAE to $216 from the initial value of over $600. For the same model, our RMSE has decreased to 2271 (from 3459 with the original model) – a significant improvement, though the large difference from the MAE suggests that we still have some large individual residuals. Our R-squared score has decreased from 0.37 in our original model and is negative for our "best" model (when the separation threshold was 10) though in our case we think that R-squared is probably less useful as we have two very different 'groupings' in our data. Overall, we think our best model is the one with the lowest MAE and second lowest RMSE (achieved when the separation threshold was 10).

- **Summary:** Probabilistic models offer a way to make individualized predictions with surprisingly little data (we just need past transactions, not demographic features). We were able to improve our modeling through introducing segmentation (and our optimal threshold to split on was a frequency of 10 weeks) though unfortunately our models still produce predictions quite far from reality for the average consumer.

Overall Model Insights and Comparations

The business can determine the method they will choose, considering their objectives and other relevant factors:

Models	Advantages	Disadvantages
Regression models	- Relatively fast to train and run - Achieve the higher accuracy compared to other models - Able to identify the most significant variables; easy to interpret	- Require the customers to be of a certain seniority - Do not factor in the time series elements - Residual variance gets larger as the value gets bigger.
Tree models	- Easy to train, doesn't need much tuning (especially with Random Forest) - With Decision Tree, it's possible to point out the most significant variables - Able to handle non-linear relationship	- Relatively slow to train - Do not factor in the time series elements - Difficult to interpret with Random Forest - Couldn't cover higher value predictions (need more pruning and a deeper level tree). This is due to the imbalance of turnover distribution.
Exponential Smoothing	- Easy to set up and train, don't need much tuning. - Might be slow to run the model in a loop for all customers in a Python notebook but could be easy to implement under another system. - Able to factor in the time series elements	- Lower accuracy compared to other models - Not able to factor in other variables - Limited long-time forecasting - Residual variance gets larger as the value gets bigger (because of the unpredictability of customer's behaviors)
CLV	- Relatively easy to set up and train, requiring only transaction data. - Can predict additional metrics like the probability a customer is alive. - Can be extended to easily predict true lifetime value (with cash flow discounting).	- Not able to account for seasonality (at least not with Lifetimes package). - Can only model repeat customers. - Relatively slow to train. - Has trouble with customer distributions that don't fit BG/NBD or Pareto/NBD distributions.

Conclusion

Research question: Can the customers' historical wagering transaction data forecast their future turnover?

Throughout our analysis, we explored a diverse range of models: from Regression analysis to greedy algorithms like Decision Tree; from Time-Series models to Probabilistic models. Still, despite the variety of approaches, a consistent issue that arose was heteroskedasticity. This phenomenon indicates that the variance of our error increases with larger values. In essence, it highlights the unpredictability of customer behavior in our data. Nonetheless, it's worth noting that the model remains capable of making predictions within a confident interval. The business can utilize the results for building marketing strategy, customer segmentation or product development. There can be so many possibilities. Overall, customer value forecasts are valuable tools for strategic decision-making, customer-centric planning, and optimizing business operations, enabling companies to focus on maximizing long-term customer value and profitability.

We believe customers' historical wagering transaction data can be used to train a model that forecasts future turnover even more accurately. But it requires collecting additional data, conducting further research on hidden patterns and trends, and exploring alternative approaches.

References

Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong. "Linear Regression" *Mathematics for Machine Learning,* edited by Lauren Cowles, Cambridge University Press, 2020, pp. 289-315.
-- "Dimensionality Reduction with Principal Component Analysis" *Mathematics for Machine Learning,* edited by Lauren Cowles, Cambridge University Press, 2020, pp. 317-343.

Aurélien Géron." Decision Trees". *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, edited by Nicole Butterfield, Nicole Taché, Michele Cronin, Beth Kelly, O'Reilly Media; 3rd edition, 2022, pp. 195-207.
-- "Ensemble Learning and Random Forests". *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, edited by Nicole Butterfield, Nicole Taché, Michele Cronin, Beth Kelly, O'Reilly Media; 3rd edition, 2022, pp. 211 -232.

"Short Time Series Forecasting: Recommended Methods and Techniques"
Cruz-Nájera, Mariel Abigail, et al. "Short Time Series Forecasting: Recommended Methods and Techniques." Symmetry 14.6 (2022): 1231.
https://www.mdpi.com/2073-8994/14/6/1231

"Counting Your Customers" the Easy Way: An Alternative to the Pareto/NBD Model
Peter S. Fader, Bruce G. S. Hardie and Ka Lok Lee *"Marketing Science Vol 24 No. 2 Spring 2005*
http://brucehardie.com/papers/018/fader_et_al_mksc_05.pdf

"A Note on Deriving the Pareto/NBD Model and Related Expressions"
Peter S. Fader and Bruce G. S. Hardie *November 2005*
http://brucehardie.com/notes/009/pareto_nbd_derivations_2005-11-05.pdf

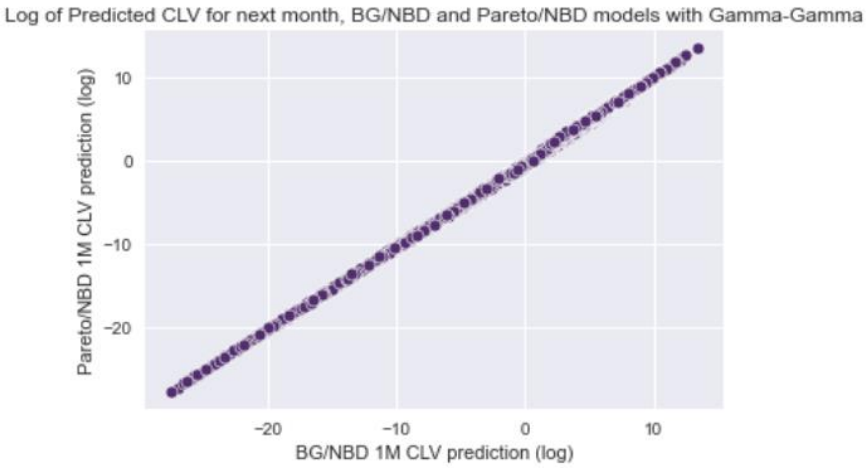
"Modeling Customer Lifetime Values with Lifetimes"
Meraldo Antonio *Towards Data Science March 2022*
https://towardsdatascience.com/modeling-customer-lifetime-value-with-lifetimes-71171a35f654

Appendix

Additional Plots for CLV Modelling

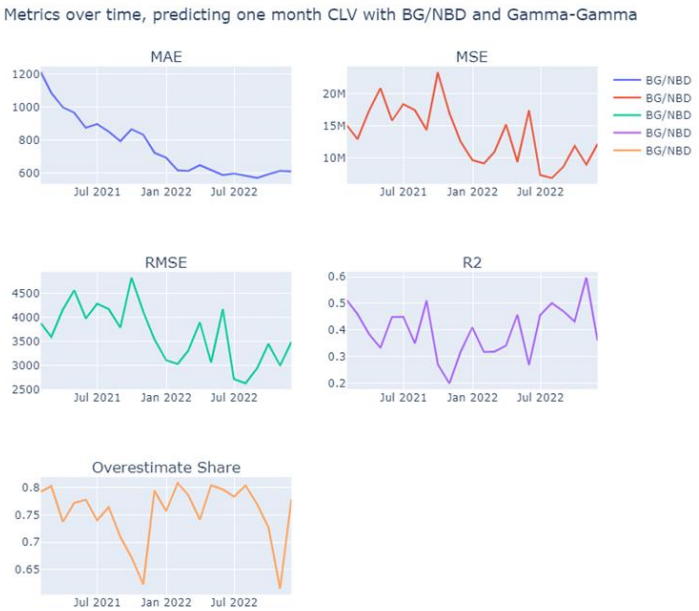
CLV-A

This chart shows the predicted values one month CLV for our initial BG/NBD with Gamma-Gamma and Pareto/NBD with Gamma-Gamma models. Predictions have been converted into a log scale. It shows how the predictions are very similar across the two models.



CLV-B

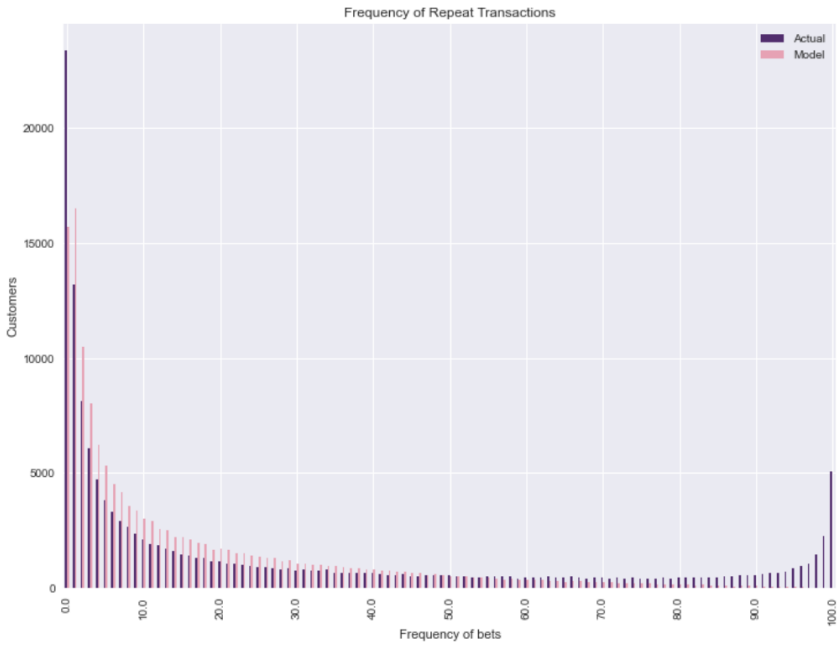
This chart shows the metrics for our original (non-segmented) CLV model when the training and prediction period was changed over time. We can see that most metrics tend to get a bit better over time. There aren't too many obvious seasonal patterns expect for the share of overestimation.



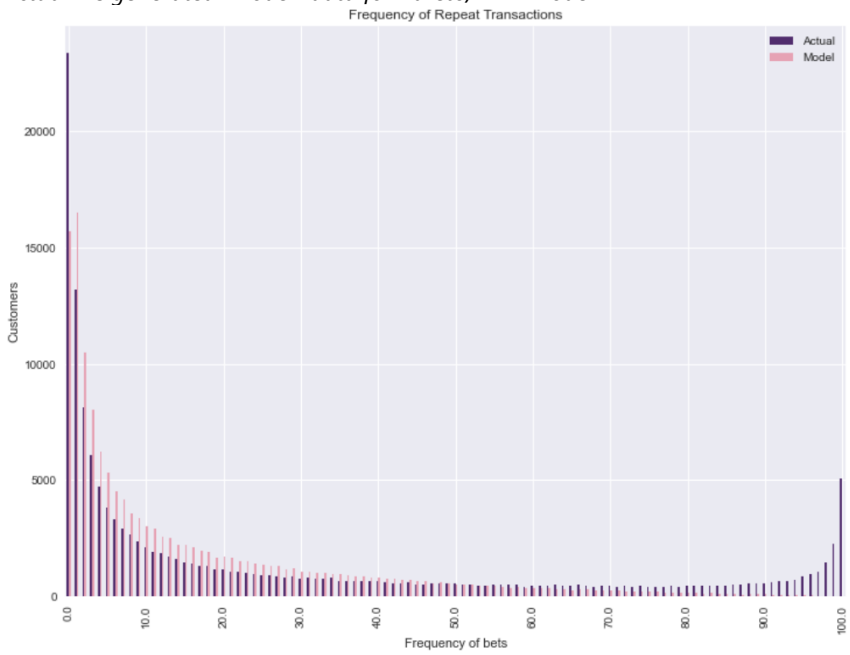
CLV-C

The two charts below show the number of customers for each given frequency of transactions: blue is the real number of customers with that frequency based on our "actual" data, while orange is artificial "model" data on the number of customers we would expect with that frequency generated by the exact probability distributions created when we fit these models to our data. In both cases we can see that there is a mismatch between our modelled distribution and our actual distribution.

Actual" vs generated "Model" data for BG/NBD Model



"Actual" vs generated "Model" data for Pareto/NBD Model



CLV-D

The chart below shows the metrics achieved on predictions for December 2022 from our BG/NBD with Gamma-Gamma model with segmentation on customer frequency. For each chart the x-axis shows different threshold values between 10 and 90, and the y-axis shows the metric score. Can see that most metrics get worse as the threshold increases, apart from R-squared.