Task1 : Estimate **expected** win probability for a given bid price

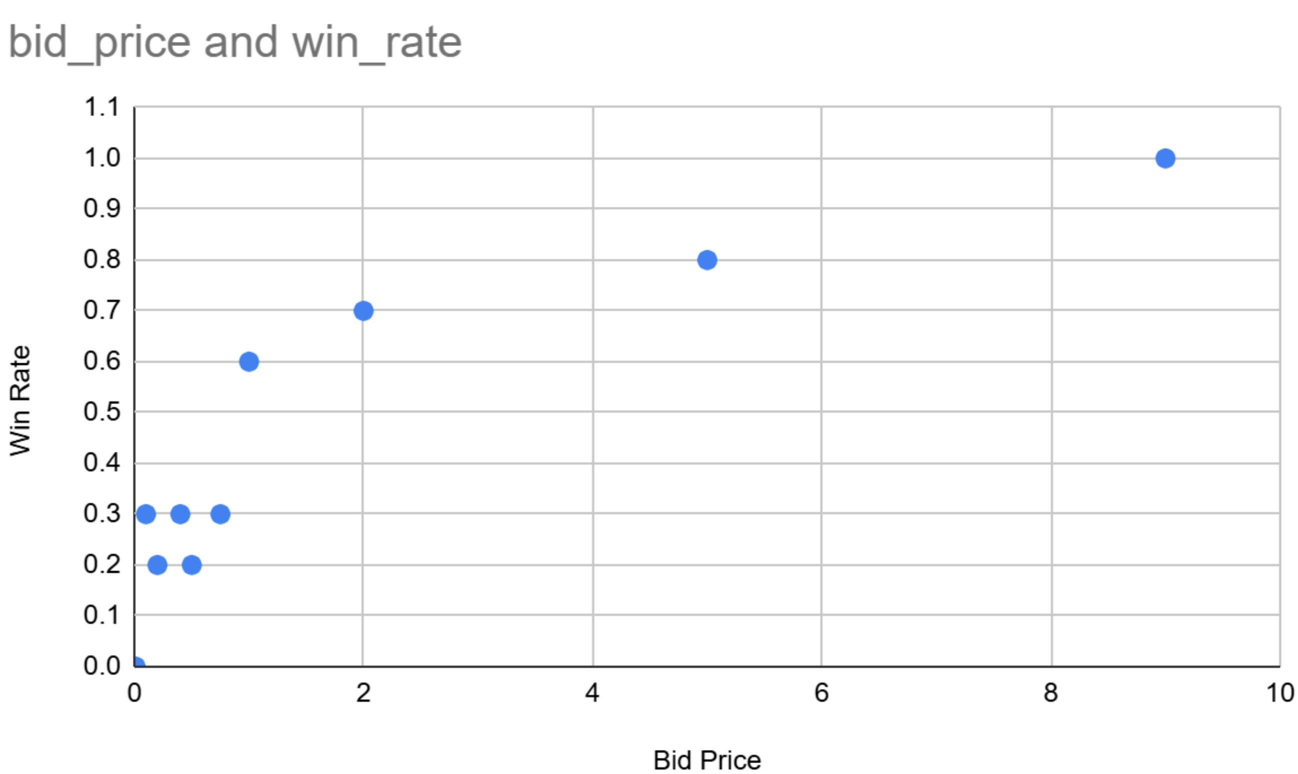
Code :
Verve_1.ipynb

Observations :
1) Few bid prices are sparse and do not have enough data for reliable estimation
2) More the sample size, the more confident we can be about the win rate estimate.
3) Expected win rate is approximately directly proportional to bid prices and we can model win rate as a function of bid price

To be more precise,

F(p) = alpha*f(p) + beta + epsilon

Where,
F(p) = estimated win rate
f(p) = function of p (to be learnt)
Alpha,beta are constants (to be learnt)
epsilon is noise in the data (irreducible noise and is due to external factors)



Solution

Assumptions:

1. **Stationarity:** We assume the underlying auction dynamics (number of bidders, their bidding strategies, etc.) will remain same over the period represented in the data and can serve as reliable ground truth. This allows us to use historical data to predict future win rates.
2. **Independence:** We assume that each auction is independent of the others
3. **Consistent Bidding:** We assume that if we were to bid the same price again in a similar situation, the probability of winning would be similar to what we've observed historically.
4. **Limited Data:** The data provided is sparse, especially for higher bid prices. This will impact the accuracy of our estimates, particularly at the extremes. **We can use bayesian smoothing (eg laplacian) to avoid overfitting on sparse data.**

Methodology:

1. Baseline Model : Linear Interpolation/Extrapolation (Simple, Less Reliable):

2. Linear Regression

3. Logistic Regression (Estimate log odds)

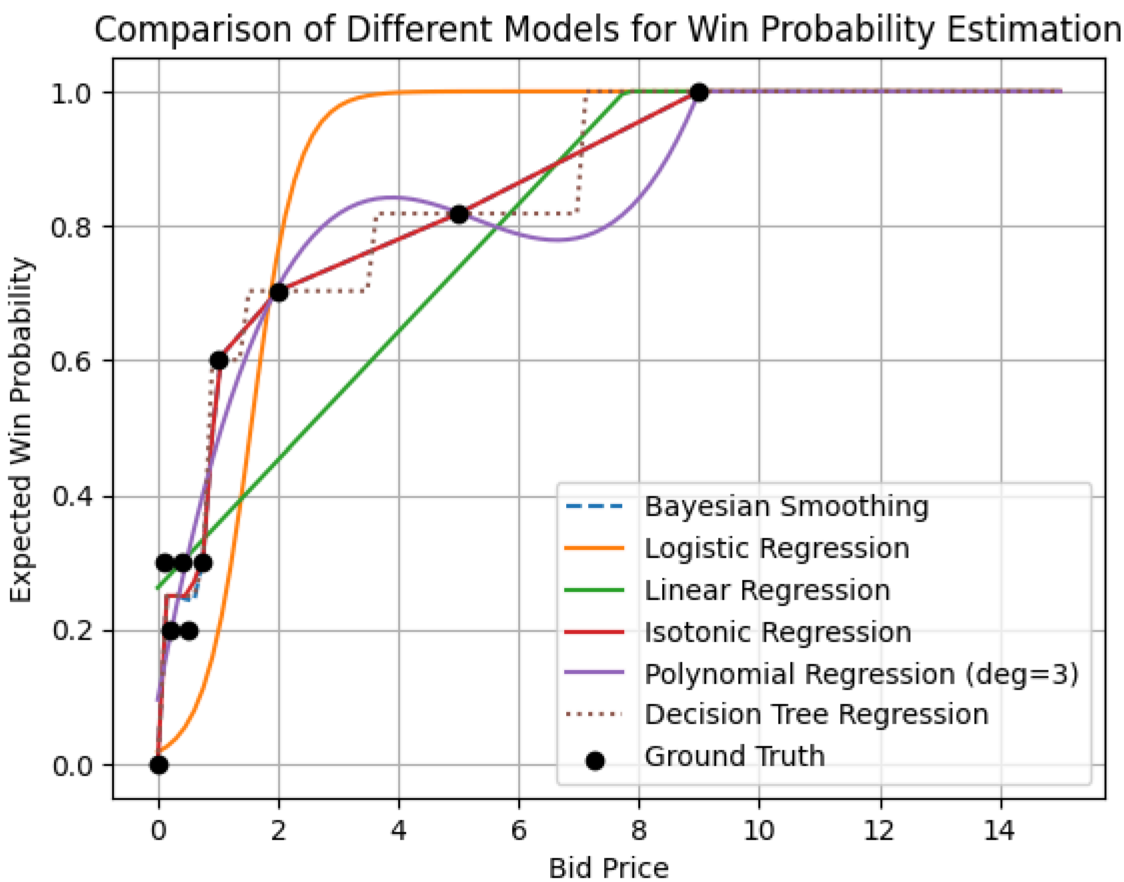
4. Decision Tree Model

5. Isotonic Regression

6. Polynomial Regression (deg = 3)

Implementation Results

Implementation Details here



We observe that Isotonic Regression, DT, Polynomial Regression and Bayesian Model are overfitting as expected as they pay a lot of attention to training data.

Linear Regression models have high bias and underfits

Logistic Regression model can generalize well if we train on more data points.

Recommendation:

Given the limited data, we should do either of the following 2 strategies

1) Use prior information : Bayesian methods if we have good priors. This would require domain expertise.
2) Get more data : If we get more data, logistic regression would be a reasonable next step. It's relatively simple to implement and can capture the non-linear relationship between bid price and win rate better than linear interpolation or bayesian piecewise constant. As we collect more data, we can consider more advanced machine learning models.

Task 2

Task is to maximize **expected net revenue**

Code Here :
Verve_2.ipynb

```
gross_revenue = 0.5
net_revenue = gross_revenue - bid_value if you win else 0
```

```
gross_revenue = 0.5

#Expected Win Rate :
# We assume expected win rate for a bid prices to be equivalent to empirical win rate ( smoothed version to avoid overfitting )

expected_win_rate = {bid_price:smoothed_win_rate for bid_price,smoothed_win_rate in df[['bid_price','smoothed_win_rate']].values}

net_revenue = {bid_price:(gross_revenue-bid_price)*expected_win_rate[bid_price] for bid_price in df['bid_price'].values}
```

```
{0.01: 4.899951000489996e-06,
 0.1: 0.12002799720027997,
 0.2: 0.0600000239999976,
 0.4: 0.030000069999929994,
 0.5: 0.0,
 0.75: -0.07501749825017498,
 1.0: -0.3001998001998002,
 2.0: -1.0544554455445545,
 5.0: -3.681818181818182,
 9.0: -8.5}
```

Bid price of 0.1$ is optimal with net revenue of 0.12$ per bid.

As we can see from the data, revenue increases with win rate and decreases with bid value.

Formally,
R(p) = W(p)*(Gross Revenue - p)

Where
R(p) = net revenue for bid price p
p = bid value
W(p) = expected win rate for given bid price p (can be learnt using a ML model or statistically as described in solution 1)
Gross revenue = 0.5 (given)

Therefore, the empirical answer of 0.12$ is only with the assumption that W(p) is learnt empirically using laplacian smoothing.

The number can change if we assume W(p) to be any other form for example, output of logistic regression, DT model etc.