Predicting mean.

\[
R(x) = \int_{-\infty}^{\infty} \phi(x) P(d\phi(x)) = \mathbb{E}(L(x), y) = \int_{-\infty}^{\infty} \phi(x) P(d\phi(x), y)
\]


<table>
<thead>
<tr>
<th>Healthy</th>
<th>0 $</th>
</tr>
</thead>
<tbody>
<tr>
<td>Churn</td>
<td>200 $</td>
</tr>
<tr>
<td></td>
<td>100,000</td>
</tr>
<tr>
<td></td>
<td>0 $</td>
</tr>
</tbody>
</table>

Expected form.

To model the decision, we need to associate costs.

As a result, we can test.

We can test the model.

The model can be validated.

Thus, we can use a regression model.

To ensure the model is accurate, we need to validate it.

We use the model to predict.
Empirical likelihood: When the data do not necessarily come from a known distribution, the empirical likelihood provides a way to construct a likelihood function.

1. Let \( R_t \) be the observed data.
2. For each data point, construct an empirical likelihood function:
   \[ L_t(\theta) = \left( \frac{1}{Z_t} \right)^n \]
   where \( Z_t = \prod_{i=1}^{n} \left( 1 + \theta R_i - \theta R_t \right) \).
3. The maximum likelihood estimate (MLE) is obtained by maximizing \( L_t(\theta) \).

The empirical likelihood is a non-parametric approach that does not require strong assumptions about the underlying distribution of the data. It is particularly useful when the data are complex or when the distribution is unknown.
4. Visualize influential weights. Does there seem to be a dominant weight?

5. Summarize each split by plotting the gain.

```
<table>
<thead>
<tr>
<th>Gain</th>
<th>Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>0.8</td>
<td>2</td>
</tr>
<tr>
<td>0.2</td>
<td>3</td>
</tr>
<tr>
<td>0.1</td>
<td>4</td>
</tr>
</tbody>
</table>
```

6. Compare the results. Does there seem to be a dominant weight?

- **Gain**: Represents the difference in performance before and after a split.
- **Depth**: Represents the depth of the split in the tree.

**Conclusion**: After analyzing the gain and depth, it appears that the first split at depth 1 has the highest gain of 0.8, indicating a significant improvement in performance. Subsequent splits have lower gains, suggesting that the initial split is the most influential.

**Why does this split make sense?**

- The initial split at depth 1 is based on the variable with the highest gain, which is likely the most important feature for making the decision.
- The subsequent splits further refine the decision, but their gains are much lower, indicating that the initial split is the most critical.

**Next steps**:

- Continue evaluating the model's performance and refining the splits.
- Consider pruning the tree to reduce complexity and improve generalization.

**Learning Gain**

![Learning Gain Graph]

- The learning gain is measured as the difference in performance between the entire dataset and each split.
- The graph shows that the learning gain increases with each split, but the rate of increase slows down after the first split.

**Why is this learning gain important?**

- The learning gain helps in understanding the model's performance and identifying the most influential features.

**Next steps**:

- Analyze the learning gain to identify the most significant splits.
- Refine the model by focusing on the most critical features.
A good claim is in clear to the topic of the paper. For instance, if it is common to measure nonparametric MSE, then the text can be expanded to:

For nonparametric, it is common to measure nonparametric MSE:

\[
\text{MSE} = \frac{\text{Estimated Value} - \text{True Value}}{\text{True Value}}^2
\]

For instance, if the true value is 100, then the estimated value is 90:

\[
\text{MSE} = \frac{90 - 100}{100}^2 = \frac{10^2}{100} = 0.1
\]

For claim checking, you can write a comparison method:

- What else can we measure?
- Other ways of the estimation procedures

We can use bootstrapping to estimate the difference as:

\[
\text{Difference} = \text{Estimate}_{\text{Bootstrap}} - \text{Estimate}_{\text{Sample}}
\]

We now model certain outcomes. The point here is to:

- Conduct a nonparametric analysis
- Conduct a bootstrapping analysis
- Conduct a randomization analysis

We can substitute these tenances of machine learning:

Alternatively, for the bootstrapping procedures

\[
\text{Estimated Value} = \frac{1}{n} \sum_{i=1}^{n} x_i
\]

\[
\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \text{Estimated Value})^2
\]

For claim checking, you can write this method:

- With bootstrapping samples to activate optimisation
- Taking known calculations (e.g., fixed) we can fix it
The threshold confusion is $PSE = (\lambda - \phi)$, but one can

$A_{c\text{up}} = \frac{1}{2} \left| \text{Logit} \right|

\text{ROC curve for the classifier curve}

For a good classifier, the ROC curve is

If the area is low, $A \leq 0.1$ (much lower than)

Then the actual point of ROC is very close to 20%.

If the area is high, $A \geq 0.5$ (much higher than)

we can find more good results for learning procedures. Great

as even a good better result is within acceptable limits of

The curve is symmetric inside error bounds.

$\frac{N_F - 1}{N_F} = \frac{N_T}{N_T}$

$N_T - 1 = \frac{N_T}{N_T}$

$N_F$ is the number of

$N_T = \frac{N_T}{N_T}$

is the number of

How to construct ROC curve