Analyzing Settlers of Catan

Caleb Johnson, Casey Hackett

April 2025

Abstract

This report explores the use of IBM SPSS Modeler to analyze gameplay data from Settlers of Catan. By applying data mining techniques, we aim to uncover which factors and achievements are most strongly associated with winning, providing insight into effective game strategies and priorities.

1 Introduction to Catan and the Project

Settlers of Catan is a competitive resource-management board game where players build settlements, cities, and roads to earn victory points. The game is played on a hexagonal board where each hex produces a specific resource (brick, lumber, wool, grain, ore) based on dice rolls. Players collect resources, trade with others, and expand their control of the board, aiming to be the first to reach 10 victory points.

Certain actions are critical to success:

- **Building settlements and cities** increases a player's resource income and contributes directly to victory points.
- **Claiming harbors** allows for more favorable trading rates, increasing resource flexibility.
- **Achieving Largest Army and Longest Road** provides valuable bonus points that often decide the winner.
- **Early placement** on high-frequency dice numbers (e.g., hexes marked 6 and 8) is key to resource generation.

Our dataset from Kaggle captures a rich set of features related to these mechanics, including:

- Game metadata (e.g., `game_id`, `date`, `start_time`, `is_extension`)
- Player performance indicators (e.g., `score`, `rank`, `winner`)
- Strategic choices (e.g., `num_of_cities`, `num_of_settlement`, `num_of_roads`)
- Harbor usage (e.g., `first_brick_harbor`, `num_of_3to1_harbor`)

- Resource dynamics (e.g., `which_resources_first`, `initial_dices`)

By analyzing these variables, we aim to uncover which gameplay behaviors and strategies are most associated with winning Catan games. Understanding the importance of factors like early harbor acquisition, city building, and initial settlement placement can provide valuable insights into effective playstyles.

2 Data Selection and Preprocessing

During the preprocessing phase, several data quality issues were identified and addressed to ensure the integrity and usefulness of the dataset. We summarize the key issues and the corresponding actions taken below.

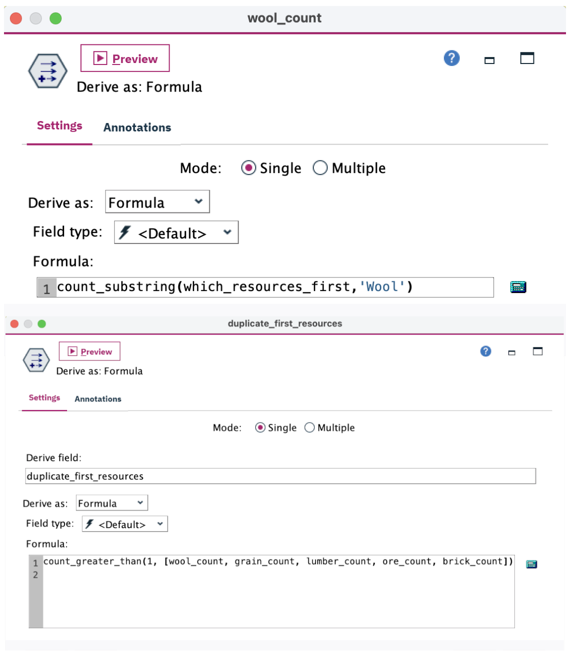
2.1 Missing Feature Data in Early Games

Some features, such as `number of harbors` and `longest road`, were not consistently collected until game 12. As a result, games 1 through 11 contain many missing or incomplete fields. To maintain the quality of analysis and avoid introducing bias from incomplete data, we decided to exclude games 1–11 from the dataset and only consider games 12 onward for modeling and evaluation using a select node.

2.2 Parsing Starting Resources

The `which_resources_first` field was initially stored as a single string, listing the resource types obtained from the player's first settlements. Since this format is not directly usable, we used `Derive` nodes to split this string into separate fields. We engineered new features based on this parsing, including:

- The count of each type of starting resource (brick, lumber, grain, ore, wool).
- The total number of starting resources.
- A binary feature indicating whether the player started with duplicate resources.



Derive as: Formula

Mode: [✓ Single] Multiple

Derive as: Formula

Field type: <Default>

Formula:

```
count_substring(which_resources_first,'Wool')
```

Derive as: Formula

Mode: [✓ Single] Multiple

Derive field: duplicate_first_resources

Derive as: Formula

Field type: <Default>

Formula:

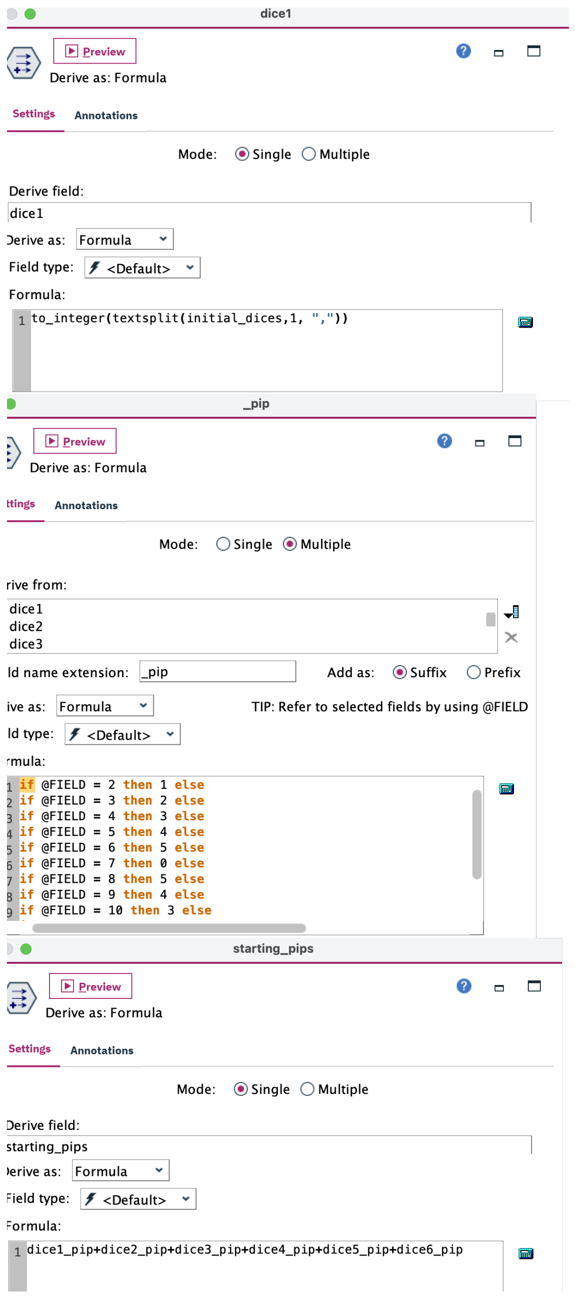
```
count_greater_than(1, [wool_count, grain_count, lumber_count, ore_count, brick_count])
```

Figure 1: Process for splitting and counting starting resources using SPSS Modeler

2.3 Transforming Initial Dice Placement

The `initial_dices` field captured the dice numbers (2–12) associated with the tiles where players placed their initial settlements. However, this data format did not directly reflect what we are truly interested in: the probability that players generate resources during the game. To address this, we:

- Split the list of initial dice values into separate fields.
- Mapped each dice number to the corresponding number of ”pips” (the number of dice combinations that produce that roll; for example, 6 and 8 are much more probable than 2 or 12).
- Summed the pip values across a player’s initial placements to create a new feature called `starting_pips`, reflecting the overall production potential from their starting settlements.



Derive as: Formula

Mode: [✓ Single] Multiple

Derive field: dice1

Derive as: Formula

Field type: <Default>

Formula:

```
to_integer(textsplit(initial_dices,1, ","))
```

Derive as: Formula

Mode: Single [✓ Multiple]

rive from:
dice1
dice2
dice3

ld name extension: _pip   Add as: [✓ Suffix] Prefix

ive as: Formula   TIP: Refer to selected fields by using @FIELD

ld type: <Default>

rmula:

```
if @FIELD = 2 then 1 else
if @FIELD = 3 then 2 else
if @FIELD = 4 then 3 else
if @FIELD = 5 then 4 else
if @FIELD = 6 then 5 else
if @FIELD = 7 then 0 else
if @FIELD = 8 then 5 else
if @FIELD = 9 then 4 else
if @FIELD = 10 then 3 else
```

Derive as: Formula

Mode: [✓ Single] Multiple

Derive field: starting_pips

erive as: Formula

ield type: <Default>

ormula:

```
dice1_pip+dice2_pip+dice3_pip+dice4_pip+dice5_pip+dice6_pip
```

Figure 2: Splicing and mapping initial dice placements to pip values and creating `starting_pips`

3 Methodology

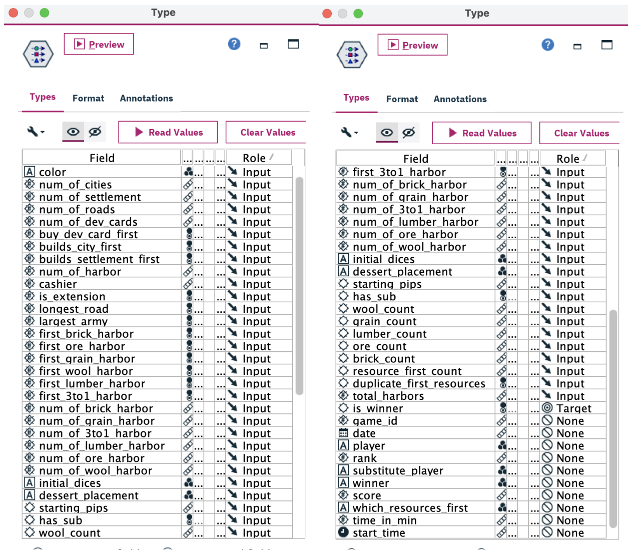
All data preprocessing and feature engineering were completed within IBM SPSS Modeler using a series of data nodes, derive nodes, and lookup operations.

Following preprocessing, we categorized the available features based on the level of strategic control players have during the game:

- **Direct Outcome Features**: Metrics that are directly tied to winning (e.g., `score`, `winner`). These features are excluded from predictive modeling because they inherently determine game outcomes.
- **Semi-Controlled Strategic Features**: Features that players can influence but do not fully control, such as earning `largest army` or `longest road`. These goals can be pursued but are also dependent on game dynamics and opponents' actions.
- **Fully Controlled Early Game Features**: Features entirely within a player's control at the beginning of the game, including `starting resources` and `starting pips`. These reflect choices made during settlement placement and initial strategy planning.

To analyze player strategy across different levels of control, we designed two main modeling groups:

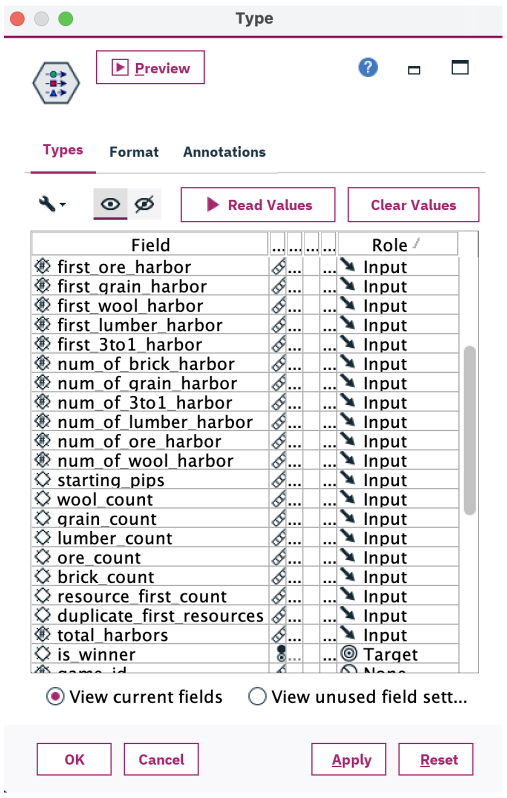
1. **Broad Strategy Analysis**: This group includes all features that are not direct outcomes (excluding score and winner) and captures both broad strategic goals (such as pursuing largest army) and decisions players have full control over. The objective here is to understand how a combination of aspirational goals and controllable choices relate to winning probability.
2. **Short-Term Strategy Analysis**: This group only includes features fully under the player's control at the beginning of the game. By isolating early-game decisions, we aim to evaluate how initial strategy alone affects a player's chances of winning, independent of how the mid- and late-game evolve.



Field	Role
color	Input
num_of_cities	Input
num_of_settlement	Input
num_of_roads	Input
num_of_dev_cards	Input
buy_dev_card_first	Input
builds_city_first	Input
builds_settlement_first	Input
num_of_harbor	Input
cashier	Input
is_extension	Input
longest_road	Input
largest_army	Input
first_brick_harbor	Input
first_ore_harbor	Input
first_grain_harbor	Input
first_wool_harbor	Input
first_lumber_harbor	Input
first_3to1_harbor	Input
num_of_brick_harbor	Input
num_of_grain_harbor	Input
num_of_3to1_harbor	Input
num_of_lumber_harbor	Input
num_of_ore_harbor	Input
num_of_wool_harbor	Input
initial_dices	Input
dessert_placement	Input
starting_pips	Input
has_sub	Input
wool_count	Input

Field	Role
first_3to1_harbor	Input
num_of_brick_harbor	Input
num_of_grain_harbor	Input
num_of_3to1_harbor	Input
num_of_lumber_harbor	Input
num_of_ore_harbor	Input
num_of_wool_harbor	Input
initial_dices	Input
dessert_placement	Input
starting_pips	Input
has_sub	Input
wool_count	Input
grain_count	Input
lumber_count	Input
ore_count	Input
brick_count	Input
resource_first_count	Input
duplicate_first_resources	Input
total_harbors	Input
is_winner	Target
game_id	None
date	None
player	None
rank	None
substitute_player	None
winner	None
score	None
which_resources_first	None
time_in_min	None
start_time	None

Figure 3: Semi-Controlled and Fully controlled inputs



Field	Role
first_ore_harbor	Input
first_grain_harbor	Input
first_wool_harbor	Input
first_lumber_harbor	Input
first_3to1_harbor	Input
num_of_brick_harbor	Input
num_of_grain_harbor	Input
num_of_3to1_harbor	Input
num_of_lumber_harbor	Input
num_of_ore_harbor	Input
num_of_wool_harbor	Input
starting_pips	Input
wool_count	Input
grain_count	Input
lumber_count	Input
ore_count	Input
brick_count	Input
resource_first_count	Input
duplicate_first_resources	Input
total_harbors	Input
is_winner	Target

(● View current fields) ○ View unused field sett...

Figure 4: Fully controlled inputs only

4 Feature Selection

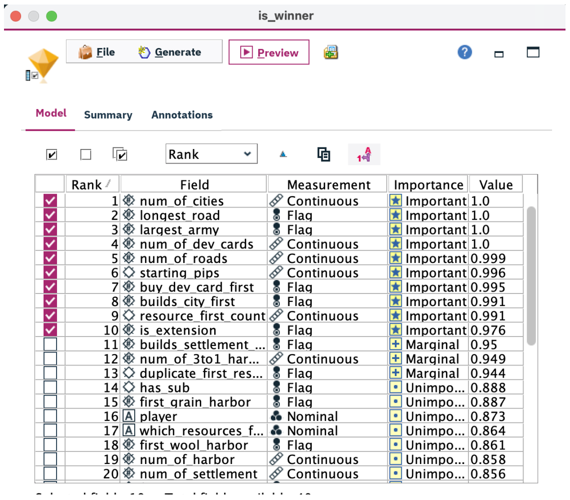
After categorizing the features into broad strategy and short-term strategy groups, we applied SPSS Modeler's `Feature Selection` node to each group separately. The goal of this step is to identify the most relevant predictors for winning, allowing us to simplify the modeling process while maintaining high explanatory power.

Separate feature selection processes were performed for the two groups:

- **Broad Strategy Group:** Includes both partially controlled strategic features and fully controlled early-game features.

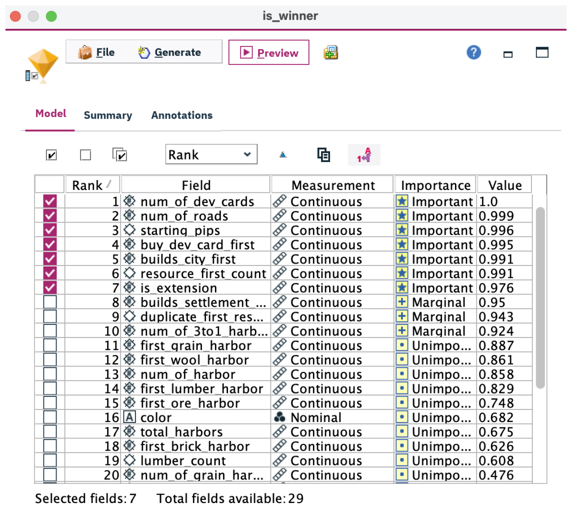
- **Short-Term Strategy Group:** Focuses only on features under the player's full control at the beginning of the game.

The following figures show the feature importance results for each group:



	Rank	Field	Measurement	Importance	Value
✓	1	num_of_cities	Continuous	Important	1.0
✓	2	longest_road	Flag	Important	1.0
✓	3	largest_army	Flag	Important	1.0
✓	4	num_of_dev_cards	Continuous	Important	1.0
✓	5	num_of_roads	Continuous	Important	0.999
✓	6	starting_pips	Continuous	Important	0.996
✓	7	buy_dev_card_first	Flag	Important	0.995
✓	8	builds_city_first	Flag	Important	0.991
✓	9	resource_first_count	Continuous	Important	0.991
✓	10	is_extension	Flag	Important	0.976
☐	11	builds_settlement_...	Flag	Marginal	0.95
☐	12	num_of_3to1_har...	Continuous	Marginal	0.949
☐	13	duplicate_first_res...	Flag	Marginal	0.944
☐	14	has_sub	Flag	Unimpo...	0.888
☐	15	first_grain_harbor	Flag	Unimpo...	0.887
☐	16	player	Nominal	Unimpo...	0.873
☐	17	which_resources_f...	Nominal	Unimpo...	0.864
☐	18	first_wool_harbor	Flag	Unimpo...	0.861
☐	19	num_of_harbor	Continuous	Unimpo...	0.858
☐	20	num_of_settlement	Continuous	Unimpo...	0.856

Figure 5: Feature Selection Results - Broad Strategy Group



	Rank	Field	Measurement	Importance	Value
✓	1	num_of_dev_cards	Continuous	Important	1.0
✓	2	num_of_roads	Continuous	Important	0.999
✓	3	starting_pips	Continuous	Important	0.996
✓	4	buy_dev_card_first	Continuous	Important	0.995
✓	5	builds_city_first	Continuous	Important	0.991
✓	6	resource_first_count	Continuous	Important	0.991
✓	7	is_extension	Continuous	Important	0.976
☐	8	builds_settlement_...	Continuous	Marginal	0.95
☐	9	duplicate_first_res...	Continuous	Marginal	0.943
☐	10	num_of_3to1_harb...	Continuous	Marginal	0.924
☐	11	first_grain_harbor	Continuous	Unimpo...	0.887
☐	12	first_wool_harbor	Continuous	Unimpo...	0.861
☐	13	num_of_harbor	Continuous	Unimpo...	0.858
☐	14	first_lumber_harbor	Continuous	Unimpo...	0.829
☐	15	first_ore_harbor	Continuous	Unimpo...	0.748
☐	16	color	Nominal	Unimpo...	0.682
☐	17	total_harbors	Continuous	Unimpo...	0.675
☐	18	first_brick_harbor	Continuous	Unimpo...	0.626
☐	19	lumber_count	Continuous	Unimpo...	0.608
☐	20	num_of_grain_har...	Continuous	Unimpo...	0.476

Selected fields: 7 Total fields available: 29

Figure 6: Feature Selection Results - Short-Term Strategy Group

5 Model Creation

The goal of the analysis is to evaluate different strategies that can increase the likelihood of winning the game. We separate the analysis into two groups: broad strategies and short-term strategies, using different features for each. The following methodology outlines the steps taken to select features, train models, and evaluate their performance.

5.1 Feature Selection

For each group, we begin by selecting the most important features using the Feature Selection node. We focus on moderately and significantly important features to ensure that the models are only trained on relevant information.

Figures 5 and 6 illustrate how we selected the features for each group.

5.2 Feature Selection for Input Nodes

Once the important features are identified, we use the Type node to select these features as the inputs for the models. This step guarantees that only the relevant features are used to predict the target variable.

The Type node setup is shown in Figure 7 and 8.

5.3 Data Partitioning

The data is partitioned into 70% training and 30% testing sets. This split allows us to train the models on a portion of the data and evaluate their performance on unseen data.

Figure 9 demonstrates how we performed the data split.

5.4 Modeling

We run four models to predict the outcome of each game:

- C5.0 (Decision Tree): A robust decision tree algorithm that efficiently generates interpretable rules.
- CHAID (Decision Tree): A statistical decision tree method that captures nonlinear relationships by focusing on significant splits.
- Neural Network: A machine learning model inspired by the structure of the

human brain that captures complex, non-linear patterns in data by learning from multiple layers of interconnected nodes.

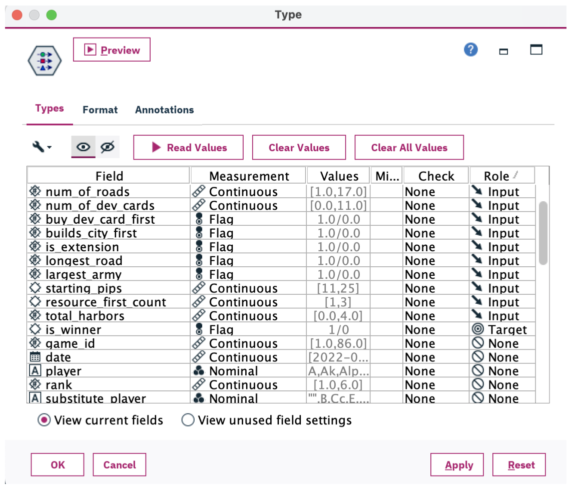
- Logistic Regression: A statistical model that estimates the probability of a binary outcome based on input features, offering a simple and interpretable baseline for classification tasks.

The modeling node setup for the four algorithms is shown in Figure 10.

5.5 Model Evaluation

After training the models, we use the Analysis and Evaluation node to assess their accuracy and gains. This step helps us determine the predictive power of each model and compare their performance. The evaluation results guide our understanding of how different features and strategies affect a player's chances of winning.

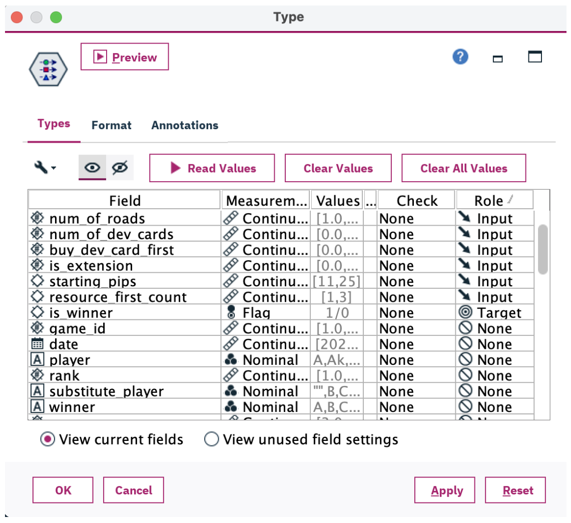
The evaluation node setupd is displayed in Figure 11.



Field	Measurement	Values	Mi...	Check	Role
num_of_roads	Continuous	[1.0,17.0]		None	Input
num_of_dev_cards	Continuous	[0.0,11.0]		None	Input
buy_dev_card_first	Flag	1.0/0.0		None	Input
builds_city_first	Flag	1.0/0.0		None	Input
is_extension	Flag	1.0/0.0		None	Input
longest_road	Flag	1.0/0.0		None	Input
largest_army	Flag	1.0/0.0		None	Input
starting_pips	Continuous	[11,25]		None	Input
resource_first_count	Continuous	[1,3]		None	Input
total_harbors	Continuous	[0.0,4.0]		None	Input
is_winner	Flag	1/0		None	Target
game_id	Continuous	[1.0,86.0]		None	None
date	Continuous	[2022-0...		None	None
player	Nominal	A,Ak,Alp...		None	None
rank	Continuous	[1.0,6.0]		None	None
substitute_player	Nominal	"",B,Cc,E,...		None	None

[✓ View current fields] View unused field settings

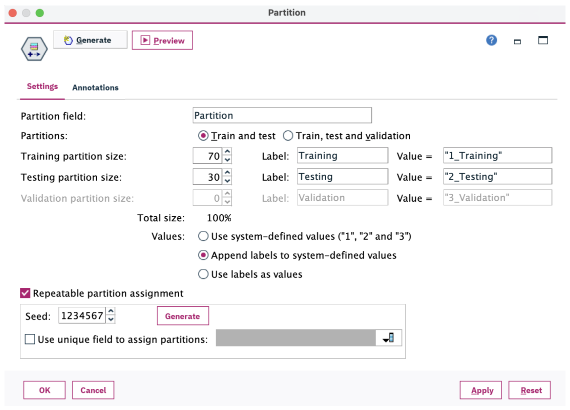
Figure 7: Broad Strategy Selected Features



Field	Measurem...	Values	...	Check	Role
num_of_roads	Continu...	[1.0,...		None	Input
num_of_dev_cards	Continu...	[0.0,...		None	Input
buy_dev_card_first	Continu...	[0.0,...		None	Input
is_extension	Continu...	[0.0,...		None	Input
starting_pips	Continu...	[11,25]		None	Input
resource_first_count	Continu...	[1,3]		None	Input
is_winner	Flag	1/0		None	Target
game_id	Continu...	[1.0,...		None	None
date	Continu...	[202...		None	None
player	Nominal	A,Ak,...		None	None
rank	Continu...	[1.0,...		None	None
substitute_player	Nominal	"",B,C...		None	None
winner	Nominal	A,B,C...		None	None

[✓ View current fields] View unused field settings

Figure 8: Short-Term strategy Features


Partition

Generate | Preview

Settings | Annotations

Partition field: Partition

Partitions: [✓ Train and test] Train, test and validation

Training partition size: 70 Label: Training Value = "1_Training"

Testing partition size: 30 Label: Testing Value = "2_Testing"

Validation partition size: 0 Label: Validation Value = "3_Validation"

Total size: 100%

Values: Use system-defined values ("1", "2" and "3")
[✓ Append labels to system-defined values]
Use labels as values

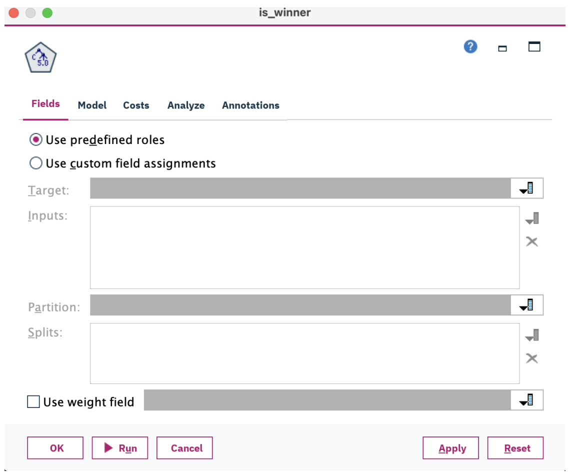
[✓ Repeatable partition assignment]

Seed: 1234567 Generate

Use unique field to assign partitions:

OK | Cancel | Apply | Reset

Figure 9: Partition Node for Data Split (70/30)


is_winner

Fields | Model | Costs | Analyze | Annotations

[✓ Use predefined roles]
Use custom field assignments

Target:

Inputs:

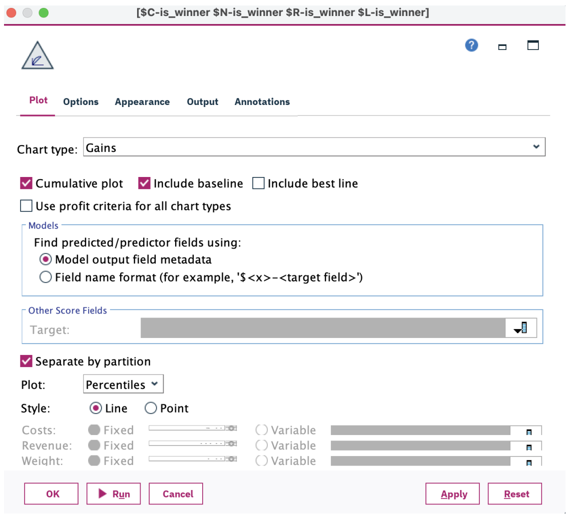
Partition:

Splits:

Use weight field

OK | Run | Cancel | Apply | Reset

Figure 10: Modeling Node Example (Similar for other models)


[$C-is_winner $N-is_winner $R-is_winner $L-is_winner]

Plot | Options | Appearance | Output | Annotations

Chart type: Gains

[✓ Cumulative plot] [✓ Include baseline] Include best line

Use profit criteria for all chart types

Models

Find predicted/predictor fields using:
[✓ Model output field metadata]
Field name format (for example, '$<x>-<target field>')

Other Score Fields

Target:

[✓ Separate by partition]

Plot: Percentiles

Style: [✓ Line] Point

Costs: Fixed Variable

Revenue: Fixed Variable

Weight: Fixed Variable

OK | Run | Cancel | Apply | Reset

Figure 11: Model Evaluation Node

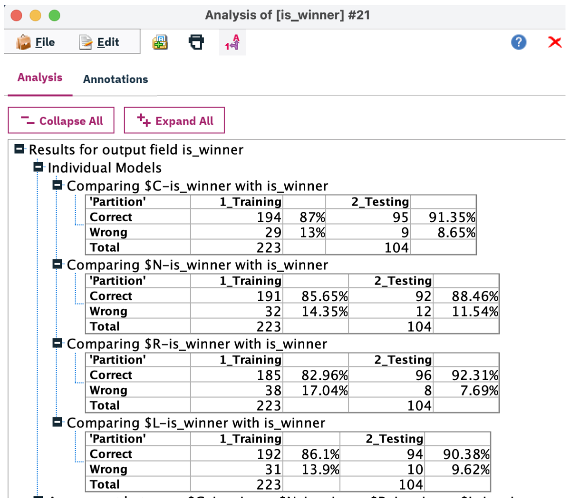
6 Evaluation

6.1 Semi-Controlled Strategy Results

We first analyze the models trained using the semi-controlled strategy features, which include both fully controlled and partially controlled aspects of player decision-making. Model performance is evaluated using the testing set confusion matrices and gain charts.

- **Confusion Matrices**: The semi-controlled models achieved high accuracy across the board. CHAID had the highest testing set accuracy (92.31%), followed closely by C5.0 (91.35%) and Logistic Regression (90.38%). The Neural Net model achieved an accuracy of 88.46%.

- **Gain Charts**: In the training set, all models performed similarly in terms of gains. On the testing set, Logistic Regression slightly outperformed the others, though all models maintained similar performance levels with minimal loss compared to training.



Results for output field is_winner

Individual Models

Comparing $C–is_winner with is_winner

'Partition'	1_Training		2_Testing	
Correct	194	87%	95	91.35%
Wrong	29	13%	9	8.65%
Total	223		104	

Comparing $N–is_winner with is_winner

'Partition'	1_Training		2_Testing	
Correct	191	85.65%	92	88.46%
Wrong	32	14.35%	12	11.54%
Total	223		104	

Comparing $R–is_winner with is_winner

'Partition'	1_Training		2_Testing	
Correct	185	82.96%	96	92.31%
Wrong	38	17.04%	8	7.69%
Total	223		104	

Comparing $L–is_winner with is_winner

'Partition'	1_Training		2_Testing	
Correct	192	86.1%	94	90.38%
Wrong	31	13.9%	10	9.62%
Total	223		104	

Figure 12: Confusion Matrices for Semi-Controlled Features

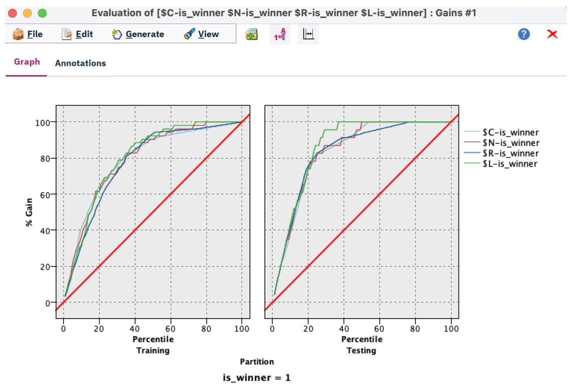


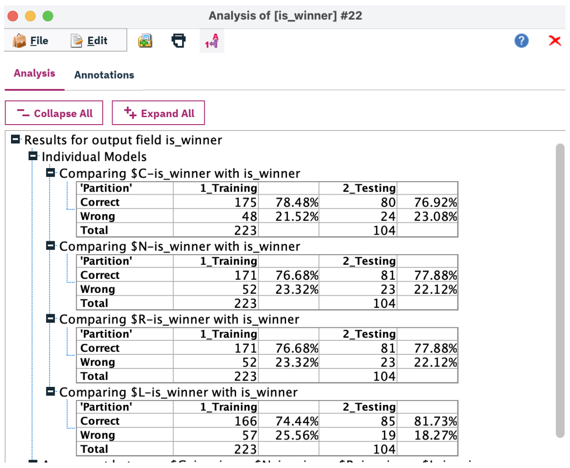
Figure 13: Gain Charts for Semi-Controlled Features

6.2 Fully Controlled Strategy Results

Next, we evaluate the models using only fully controlled features, based purely on player decisions.

- **Confusion Matrices**: Logistic Regression achieved the highest testing set accuracy (81.73%), followed by the Neural Net and CHAID models (both at 77.88%), and C5.0 (76.92%).

- **Gain Charts**: On the training set, CHAID had the highest gains, while Neural Net and Logistic Regression followed closely. On the testing set, Neural Net and Logistic Regression performed best, with CHAID slightly behind, and C5.0 trailing.



Analysis of [is_winner] #22

Results for output field is_winner

Individual Models

Comparing $C–is_winner with is_winner

'Partition'	1_Training		2_Testing	
Correct	175	78.48%	80	76.92%
Wrong	48	21.52%	24	23.08%
Total	223		104	

Comparing $N–is_winner with is_winner

'Partition'	1_Training		2_Testing	
Correct	171	76.68%	81	77.88%
Wrong	52	23.32%	23	22.12%
Total	223		104	

Comparing $R–is_winner with is_winner

'Partition'	1_Training		2_Testing	
Correct	171	76.68%	81	77.88%
Wrong	52	23.32%	23	22.12%
Total	223		104	

Comparing $L–is_winner with is_winner

'Partition'	1_Training		2_Testing	
Correct	166	74.44%	85	81.73%
Wrong	57	25.56%	19	18.27%
Total	223		104	

Figure 14: Confusion Matrices for Fully Controlled Features

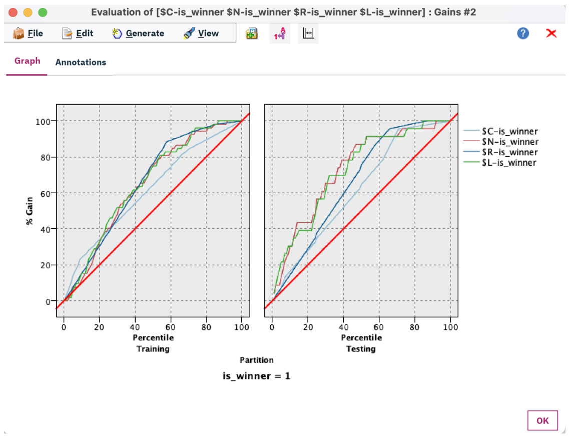


Figure 15: Gain Charts for Fully Controlled Features

6.3 Comparison and Conclusions

Comparing the two strategies provides several important insights:

- **Accuracy**: Models trained on semi-controlled features achieved higher testing set accuracies across all model types compared to those trained only on fully controlled features. This suggests that incorporating broader game state information improves predictive ability.

- **Gains**: Semi-controlled models exhibited more stable gain performance, losing less gain between the training and testing sets. Fully controlled models experienced greater loss in gains when moving to the testing set, suggesting a higher degree of overfitting.

- **Model Performance**: CHAID delivered the best performance among all models when evaluating the more complex Semi-controlled features. In contrast, Logistic Regression outperformed the other models on the Fully-controlled feature subset.

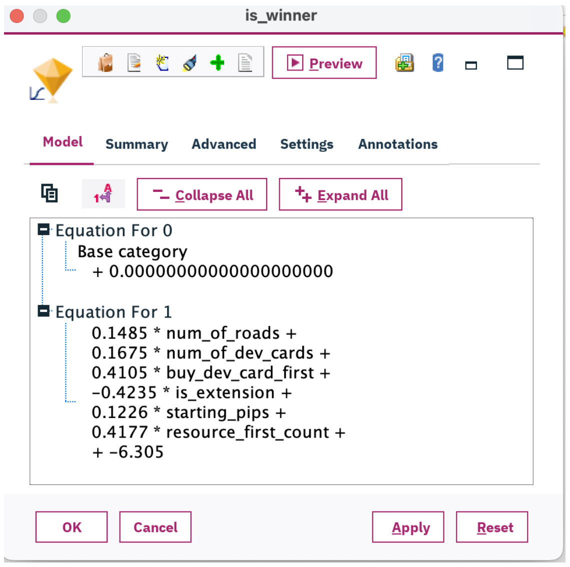
- **General Observations**: While early game decisions (fully controlled features) are important, the results clearly indicate that dynamic factors throughout the game (semi-controlled features) are critical for accurately predicting outcomes. This highlights the importance of adaptability in gameplay.

7 Analysis

7.1 Fully Controlled Strategy: Logistic Regression Analysis

The Logistic Regression model emerged as the best-performing model for the fully controlled feature set, achieving a testing accuracy of 81.73%. Notably, 2 out of 4 models for this feature set defaulted to predicting the majority class (always guessing 0), and one model predicted some 1s but performed worse on testing. Logistic Regression was the only model that successfully outperformed majority-class guessing, improving predictive accuracy by approximately 5%.

To understand the strategies associated with winning, we analyze the coefficients of the logistic regression model. Positive coefficients indicate that an increase in the corresponding feature value raises the likelihood of winning, while negative coefficients indicate the opposite.



```
Equation For 0
  Base category
    + 0.00000000000000000000
Equation For 1
    0.1485 * num_of_roads +
    0.1675 * num_of_dev_cards +
    0.4105 * buy_dev_card_first +
    -0.4235 * is_extension +
    0.1226 * starting_pips +
    0.4177 * resource_first_count +
    + -6.305
```

Figure 16: Logistic Regression Coefficients for Fully Controlled Inputs

The logistic regression model produced the following weights:

Feature	Coefficient
num_of_roads	0.1485
num_of_dev_cards	0.1675
buy_dev_card_first	0.4105
is_expansion	-0.4235
starting_pips	0.1226
resource_first_count	0.4177
Constant (Intercept)	-6.3050

Table 1: Logistic Regression Coefficients for Fully Controlled Inputs

Key findings from the coefficient analysis:

- Building more roads and acquiring more development cards slightly increase the chance of winning.

- Buying a development card early and starting with more varied resources are strongly positive influences.

- Playing in an expansion map format slightly decreases the chance of winning since there are more players.

Thus, fully controlled early-game decisions do influence the probability of winning, but even the best model highlights that the early-game factors alone have limited predictive power.

7.2 Semi-Controlled Strategy: CHAID Analysis

For the semi-controlled feature set, the CHAID decision tree model achieved the highest testing accuracy at 92.31%. This group included broader strategic goals that players can strive toward but not fully control, such as obtaining the Largest Army or Longest Road.

We first examine the variable importance chart to identify which features had the greatest impact on predicting wins.

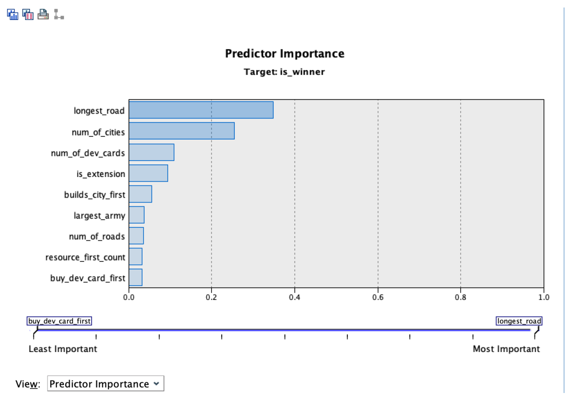


Figure 17: CHAID Model Feature Importance for Semi-Controlled Inputs

The CHAID tree shows that the most important factors for winning in the semi-controlled set were:

- Obtaining the Longest Road
- Building more cities
- Earning bonus points through Development Cards

The flowchart created by the CHAID algorithm further illustrates how combinations of semi-controlled strategies lead to higher probabilities of winning.

Key insights from the CHAID tree:

- Players with more cities have a much higher chance of winning, bringing in both points and resources.

- If you are only able to secure a medium amount of cities the next most important thing is securing Longest Road.

- Finally, if that is not possible, you must secure any points you can via dev cards.

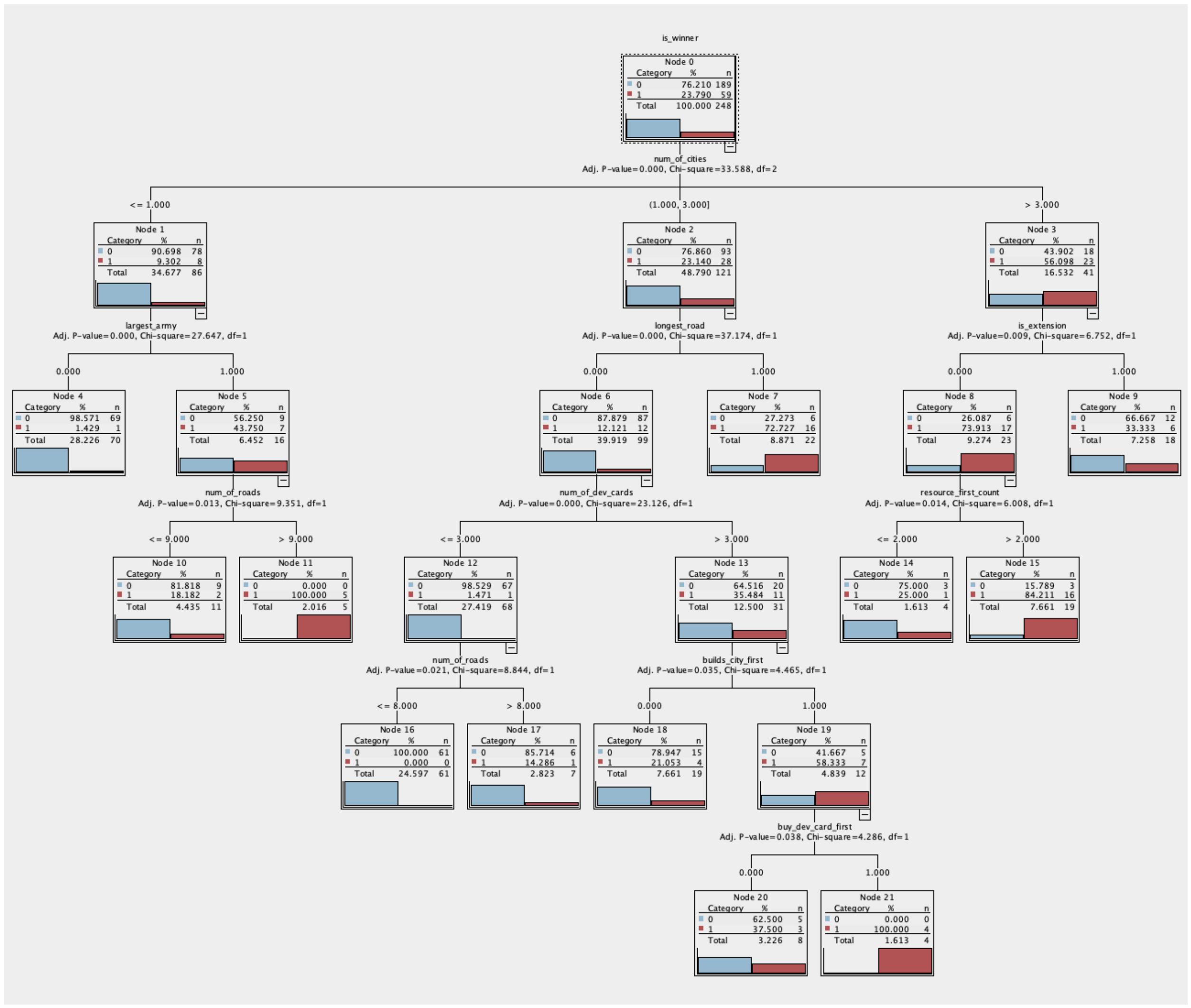


Figure 18: CHAID Decision Tree Flowchart for Semi-Controlled Inputs

7.3 Comparison and Conclusions

Comparing the two approaches:

- The fully controlled analysis, particularly through the Logistic Regression model, highlights several core factors that improve winning chances: selecting starting locations with high resource pips (strong income potential), prioritizing building roads, buying development cards early, and maintaining a diverse initial setup. These early-game decisions create a strong foundational position.

- The semi-controlled CHAID analysis complements this by emphasizing the importance of flexible mid- to late-game objectives. Specifically, securing the Largest Army, maintaining strong resource flow, and leveraging Development Cards for bonus points emerge as critical adaptive strategies to capitalize on the foundation established early.

Rather than viewing the strategies as separate, the two perspectives work together: Establishing a high-quality early position through careful starting choices (e.g., maximizing pips and early development card access) lays the groundwork for success, while dynamic mid-game flexibility—pursuing Largest Army and resource advantages—is essential for maintaining and expanding that lead throughout the match. Winning, therefore, depends both on intelligent initial setup and on responsive strategic adaptation as the game unfolds.

8 Final Strategy Conclusion

Based on the analysis of both fully controlled and semi-controlled strategies, the optimal strategy for future Catan games can be summarized as follows:

8.1 Early Game Strategy

- **Maximize starting pips:** Prioritize settling on high-pip tiles to increase the likelihood of resource production early in the game.

- **Secure broad opening resources:** Ensure access to a balanced set of resources, particularly *ore*, *wheat*, and *sheep*, which are essential for purchasing development cards.

8.2 Mid to Late Game Strategy

- **Pursue Largest Army:** Build cities on your *ore*, *wheat*, and *sheep* starting areas to strengthen your position for purchasing development cards and accumulating points. The Largest Army strategy will increase your chances of winning by giving you control over this key area of the game.

- **Adapt to Longest Road if Largest Army is Contested:** If the Largest Army strategy is too contested by other players, CHAID suggests transitioning to a strategy focused on *Longest Road*.

This pivot can help maintain a competitive edge by shifting your focus to a different route for achieving victory points.

This adaptable approach allows for early resource optimization while remaining flexible in the mid to late game, ensuring you can adjust your strategy based on the game's progression and the competition.

9 Problems and Future Work

9.1 Problems Encountered

- **Limited Dataset Size:** The number of games analyzed was relatively small, which may limit the generalizability of the conclusions.
- **Model Overfitting:** Some models, particularly for the fully controlled set, showed signs of overfitting, performing well on the training data but struggling with testing data.

9.2 Future Work

- **Expand Dataset:** Collect more games to improve the robustness and reliability of the model results.
- **Explore Additional Features:** Incorporate more mid- and late-game variables, such as trade patterns, robber placements, and port usage, to better capture player dynamics.
- **Experiment with Other Models:** Test more advanced machine learning models to see if predictive performance can be improved.

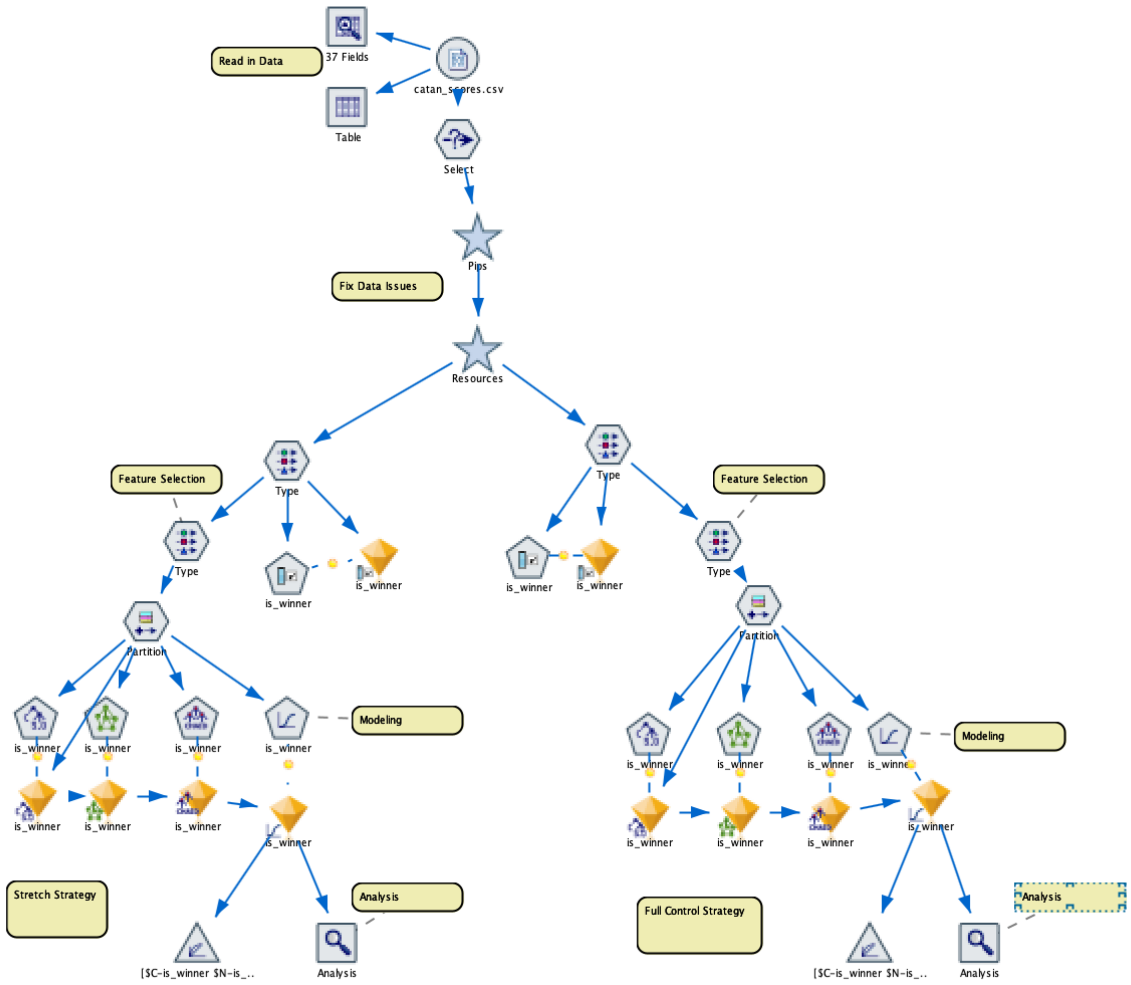


Figure 19: Commented IBM SPSS Modeler Stream

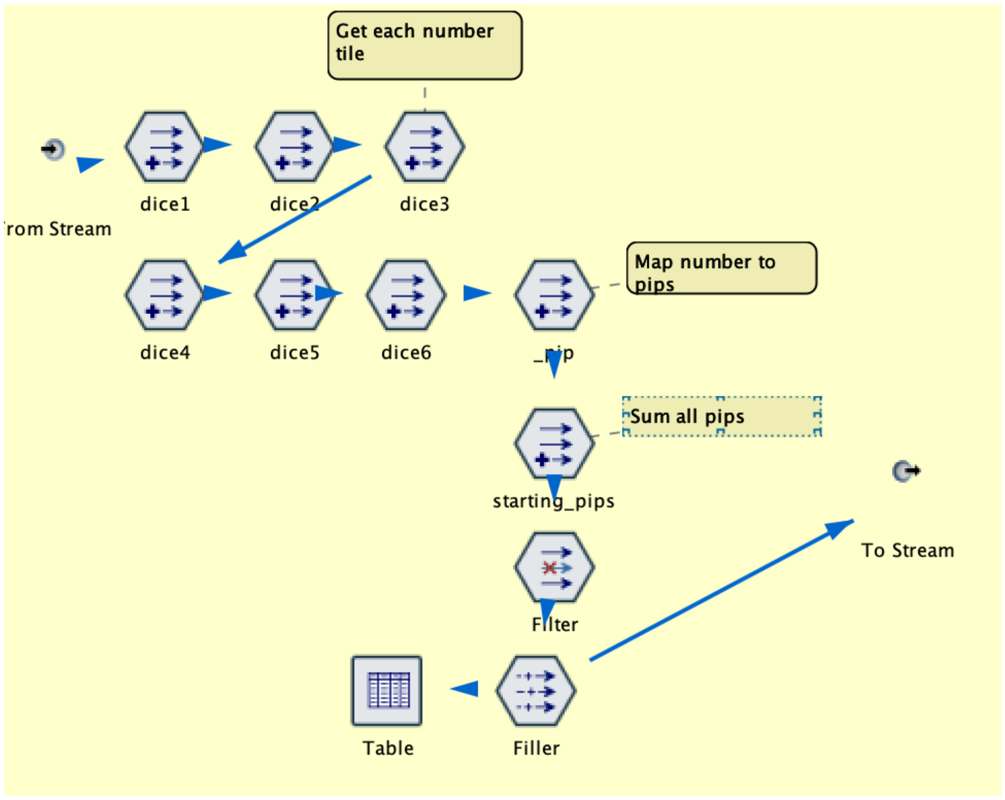


Figure 20: Commented Dice Conversion SuperNode

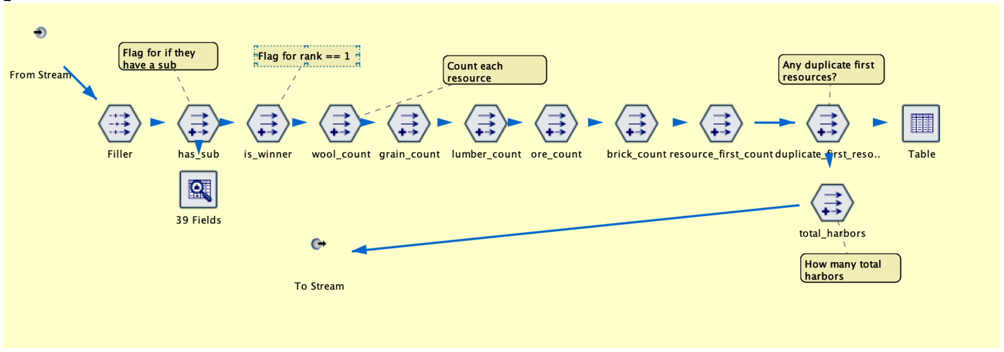


Figure 21: Commented Resource Conversion Supernode