Machine Learning: Assignment-2

Experiment: Predicting Car Prices in "Forza Horizon 5"

Dataset from https://www.kaggle.com/datasets/deepcontractor/froza-horizon-5-cars-dataset

GitHub Repository at https://github.com/rogue0xbyte/learning-ml

Description

Forza Horizon 5 is a 2021 racing video game set in an open world environment based in a fictional representation of Mexico. We will be attempting to use the vast dataset and learn about car pricing in this game.

2 separate sections of this experiment - 2 different ML concepts are implemented. (1) Gaussian Naive Bayes model is implemented to predict prices, using sci-kit, to demonstrate Learning from Examples. However, it results in an extremely inaccurate result. (2) We have implemented General-to-specific ordering over hypotheses.

Team Members:

1. Aaditya Rengarajan (21Z202)
2. Aadil Arsh S R (21Z201)
3. Abhinav D (21Z203)
4. Gali Likith Sai (21Z216)
5. Mithilesh E N (21Z229)
6. R Vishal (21Z240)
7. Soorya Subramani (21Z258)
8. M R Roohith (22Z432)

- Assignment submission by 8-member team, students of B.E CSE G1, PSG College of Technology

1. Run `main.py`

1. Read `Forza_Horizon_Cars.csv` dataset.
2. `preprocess(data)` using the `preprocess()` function defined in `conditioning.py`. a. 'In_Game_Price' is the target variable for prediction, so we convert string price to numerical values (remove commas) b. Handle '??' and 'info_not_found' values in all columns by replacing them with 0 c. One-hot encode 'stock_specs' column d. One-hot encode 'Drive_Type' column e. Process a few other columns - converting them to floating point values f. Eliminate NaN values by replacing them with 0
3. Define the required features in the `features` variable.
4. Split the dataset into training and testing data using `train_test_split()`
5. Predict the prices using GaussianNB, output the data onto a CSV file and also capture the performance measures of the model onto a TXT file.
6. Generate n hypotheses (defaulted to 100 hypotheses). a. Specificity levels are randomly assigned and shuffled to ensure randomness in selecting constraints' specificity. b. For each specificity level in specificity_levels: i. A random row is selected from the provided dataset (dataframe). ii. An empty hypothesis list is initialized to store column constraints. c. For each column in the randomly selected row, a decision is made based on the specificity level: i. If a randomly generated number falls within the specificity threshold (as defined by the level), the column and its specific value from the row are included in the hypothesis. ii. If the random number is outside the specificity threshold, a placeholder (?) is used to indicate an unspecified value for that column in the hypothesis. We ensure that 33% of hypotheses are specific, 33% are general and the rest lie in-between.
7. Map all the instances of records in the dataset to the generated hypotheses.
8. Store this output to an excel document, and then visualize this data in a graph.

2. Graphical Output

- Graph 1 would contain a scatter plot of all Instances X
- Graph 2 would contain a scatter plot of all Hypotheses H
- Hypotheses are be sorted from bottom to top as General to specific
- 3 Example Hypotheses and 3 sample corresponding instances are printed and highlighted/plotted and labelled on the graph.

Theory

1. One-Hot Encoding

- Used to represent categorical data as binary vectors.
- We use it to convert categorical variables into a format that can be provided to machine learning algorithms to better interpret categorical data.
- Example: 'stock_specs' in our dataset assumes values of `['A', 'B', 'C', 'D', 'S1', 'S2', NaN, 'info_not_found']`. These are considered as separate boolean features.

2. Gaussian Naive Bayes Model

- Utilizes Bayes' theorem to predict the probability of a class given certain features.
- Assumes that all features are independent of each other, which is why it's termed "naive" (even if it's not always true in real-world scenarios).
- Bayes' theorem calculates the probability of a hypothesis (class) given the evidence (features) using conditional probabilities. `Formula: P(Class|Features) = (P(Features|Class) * P(Class)) / P(Features)`
- **Gaussian Naive Bayes:** Assumes that continuous features follow a Gaussian distribution.
- The assumption of feature independence might not hold true in real-world scenarios.

3. Hypotheses

- Here, h is conjunction of constraints on attributes
- Each constraint can be
  - a specific value (e.g., W ater = W arm)
  - don't care (e.g., \W ater =?")
  - no value allowed (e.g.,\Water=;") [Not used in our program]

4. Prototypical Concept Learning Task

- Given
  - Instances X
  - Target function c: In_Game_Price -> `int`
  - Hypotheses H: Conjunctions of literals.
  - Training examples D
- Determine: A hypothesis h in H such that h(x) = c(x) for all x in D.

Demonstration

1. Naive Bayes Results

We observe a failure in implementation of the Gaussian Naive Bayes model for this dataset, with the following scores returned: Accuracy: 0.0 Precision: 0.0 Recall: 0.0 F1 Score: 0.0

This proves that the GNB model is not a good model to suit the data we are handling, and we must implement a different model.

- When encountering unseen combinations of features, it can lead to zero probabilities, impacting predictions.
- Sensitivity to irrelevant features can affect performance.
- Gaussian Naive Bayes (GNB) assumes that features are independent and have a Gaussian distribution. This assumption might not hold true for complex real-world data, especially when dealing with predicting car prices in a video game like Forza Horizon 5.
- Car prices in our dataset could have complex and non-linear relationships between different attributes/features. Naive Bayes assumes linear relationships between features, which might not capture these complexities accurately.

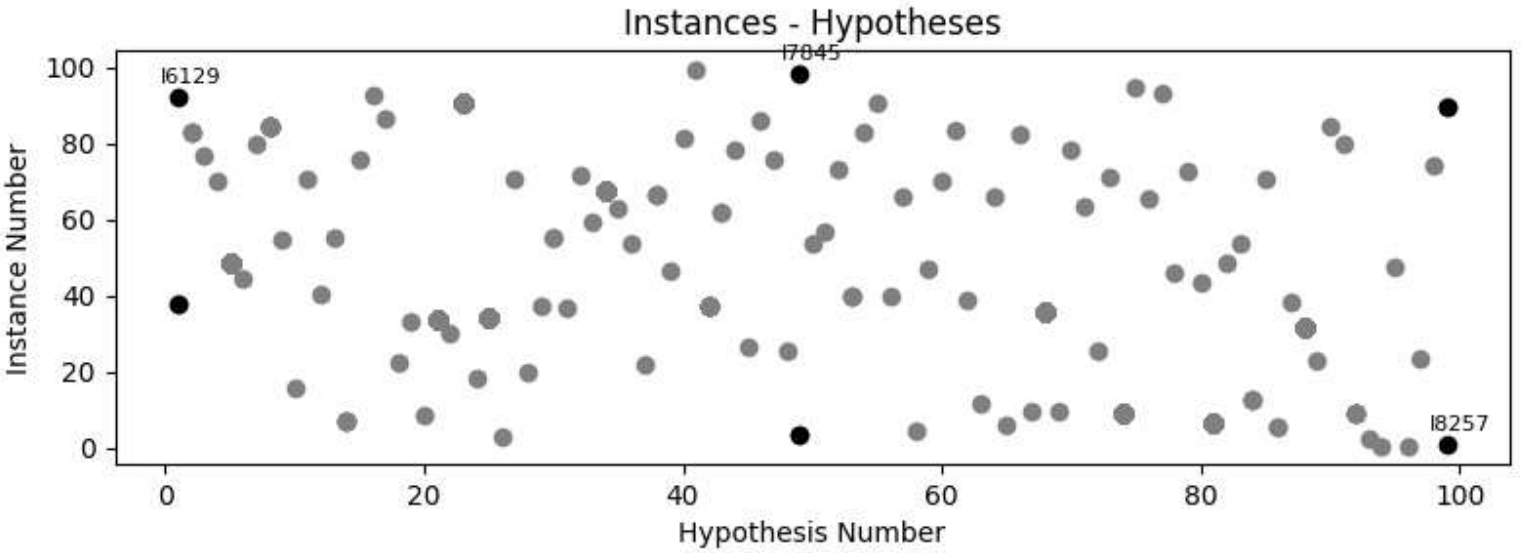
2. Generated Hypotheses

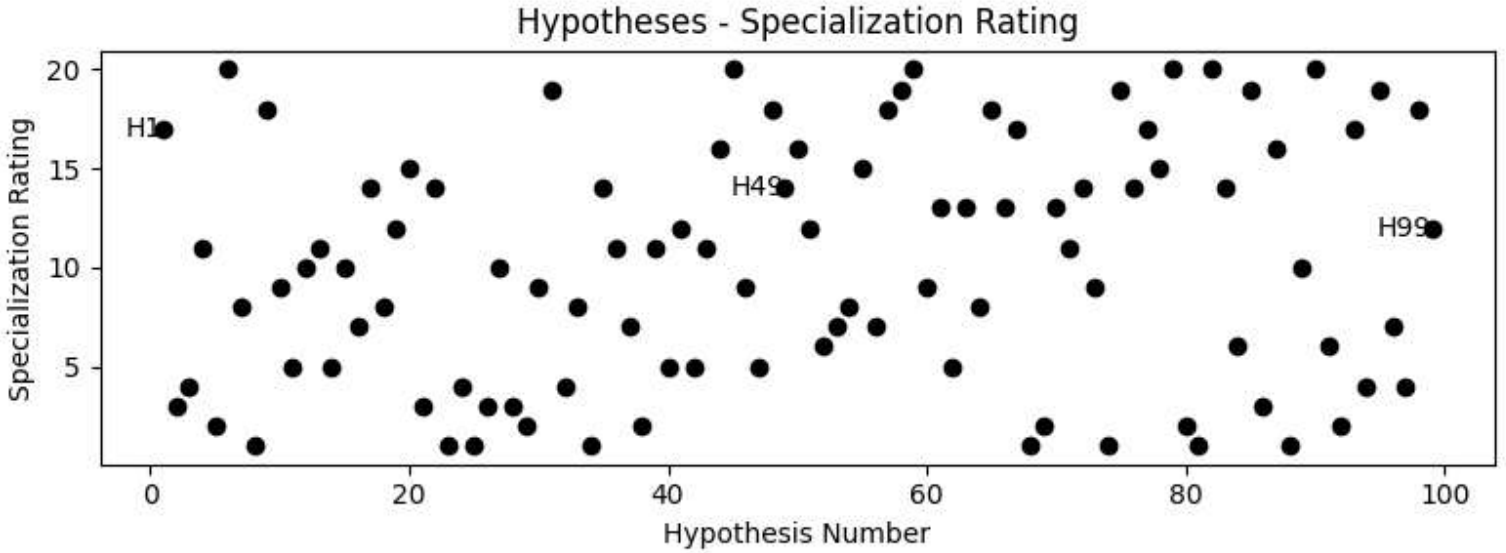
Hypothesis Number	stock_A	stock_B	stock_C	stock_D	stock_S1	stock_S2	drive_AWD	drive_FWD	drive_RWD	speed	handling	acceleration	launch	braking	Offroad	Top_Speed	0-60_Mph	g-force	Horse_Power	Weight_lbs
1	?	?	F	T	F	F	?	F	T	3.1	4.3	2.9	5.3	2.6	5.9	0.0	0.0	0.0	60.0	1235.0
6	F	T	F	F	F	F	F	F	T	5.7	3.8	4.1	3.1	3.0	6.4	157.8	6.2	0.82	410.0	3244.0
45	F	F	T	F	F	F	F	F	T	5.4	5.1	4.1	2.7	3.0	5.1	0.0	0.0	0.0	206.0	3329.0
49	?	F	F	F	F	F	?	?	?	7.3	6.5	?	6.1	6.1	4.7	0.0	?	0.0	608.0	4134.0
59	T	F	F	F	F	F	T	F	F	7.6	6.0	8.7	9.7	4.8	5.1	0.0	0.0	0.0	603.0	4515.0
79	F	F	F	T	F	F	F	F	T	4.8	4.1	3.8	3.3	2.5	5.2	130.5	7.3	0.81	148.0	2315.0
82	F	F	T	F	F	F	F	F	T	4.7	3.6	3.3	3.2	2.4	5.4	0.0	0.0	0.0	450.0	3799.0
90	F	F	F	F	T	F	F	F	T	7.9	7.2	6.3	6.8	6.9	4.1	219.9	3.0	1.02	562.0	3274.0
99	?	?	?	F	?	?	T	F	?	6.3	3.8	5.8	5.6	3.7	6.7	?	0.0	0.0	475.0	?

3. Instances Mapped

stock_A	stock_B	stock_C	stock_D	stock_S1	stock_S2	drive_AWD	drive_FWD	drive_RWD	speed	handling	acceleration	launch	braking	Offroad	Top_Speed	0-60_Mph	g-force	Horse_Power	Weight_lbs	Hypothesis Number
F	F	T	F	F	F	F	F	T	4.7	3.6	3.3	3.2	2.4	5.4	0	0	0	450	3799	82
F	F	F	F	T	F	F	F	T	7.9	7.2	6.3	6.8	6.9	4.1	219.9	3	1.02	562	3274	90
F	F	F	T	F	F	F	F	T	4.8	4.1	3.8	3.3	2.5	5.2	130.5	7.3	0.81	148	2315	79
F	T	F	F	F	F	F	F	T	5.7	3.8	4.1	3.1	3.0	6.4	157.8	6.2	0.82	410	3244	6
T	F	F	F	F	F	T	F	F	7.6	6.0	8.7	9.7	4.8	5.1	0	0	0	603	4515	59

4. Graphical Output





Instance ID: I6129,
Instance Data: [F, F, F, T, F, F, F, F, T, '3.1', '4.3', '2.9', '5.3', '2.6', '5.9', 0.0, 0.0, 0.0, 60.0, 1235.0],
Hypothesis ID: H1

Instance ID: I7845,
Instance Data: [F, F, F, T, F, F, F, F, T, '3.1', '4.3', '2.9', '5.3', '2.6', '5.9', 0.0, 0.0, 0.0, 60.0, 1235.0],
Hypothesis ID: H49

Instance ID: I8257,
Instance Data: [F, F, F, T, F, F, F, F, T, '3.1', '4.3', '2.9', '5.3', '2.6', '5.9', 0.0, 0.0, 0.0, 60.0, 1235.0],
Hypothesis ID: H99