Online Appendix - *Strategic Information Disclosure to Classification Algorithms: An Experiment*

Jeanne Hagenbach* Aurélien Salas†

August 25, 2025

This Online Appendix includes (1) the table of correlations between all the answers given by subjects in the pre-study, (2) the analysis of the treatment *Others* which we had additionally pre-registered, (3) the pre-study questionnaire and the detailed instructions for the main experiment, (4) the codes for the Naive Bayes algorithm and for the RATIO procedure.

*CNRS, Sciences Po, WZB, CEPR, CESifo - jeanne.hagenbach@sciencespo.fr
†Sciences Po - aurelien.salas@sciencespo.fr

1 Correlations between answers from the pre-study

Participants to the pre-study answered a questionnaire made up of 30 questions with binary answers (except for gender). The table below gives the Pearson correlation coefficient for any pair of answers. Blue color indicates positive correlation and red color indicates negative correlation. As explained in section A.4 of the Appendix, we used this table to construct the questionnaire presented in Part 1 of the main experiment.

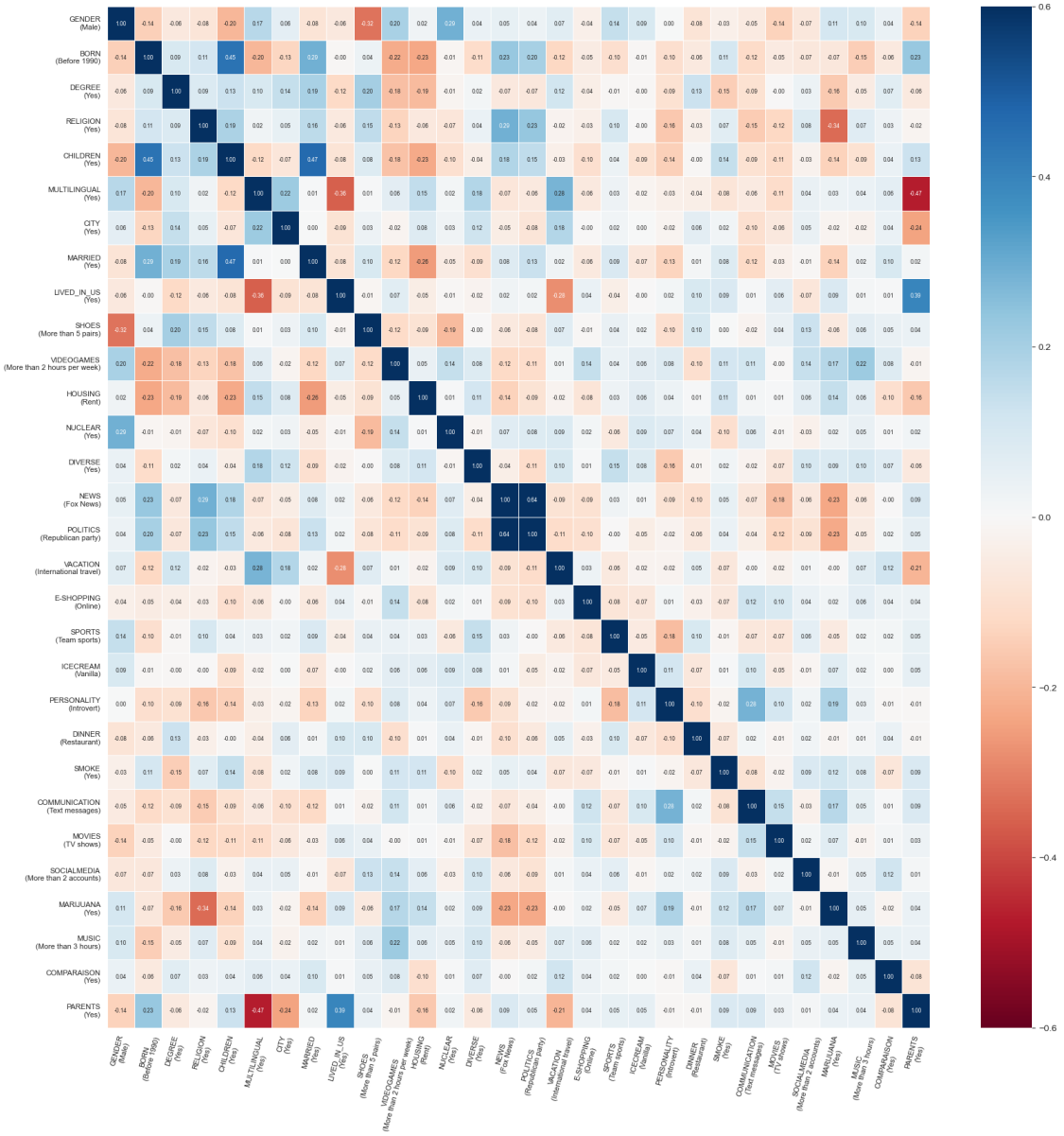

Figure 9: Matrix of correlation coefficients for all questions of the pre-study

2 Additional pre-registered treatment - *Others*

We pre-registered three treatments, randomly assigned between subjects: *Control*, *Info*, *Others*. In the main text, we focus on the comparison between *Control* and *Info*. We now describe the *Others* treatment, and how it affects subjects' disclosure strategies compared to *Control*.

2.1 Description of the *Others* treatment

The objective of *Others* is to study whether subjects disclose information differently when they learn that disclosed information may be further used to train the algorithm. Precisely, in the *Others* treatment, subjects read the same sentences as in the *Control* treatment but we add the following text: *In subsequent experiments, we may use the answers you disclosed to further train our algorithm and make it better at guessing the answers of other individuals. This will be done in full anonymity as we record your answers anonymously.*

For this treatment, we had pre-registered the following hypothesis. It is based on the idea that subjects' behavior could be influenced by concerns about the future use of their disclosed information. Specifically, when subjects are explicitly told that their answers could be used to improve the algorithm's performance against others in future experiments, they may experience discomfort, guilt or hesitation about contributing to this process.

Hypothesis 1 (Others) Pooling all target questions, subjects hide more answers in the *Others* treatment than in the *Control* treatment.

2.2 Implementation

A total of 488 subjects were allocated to the *Others* treatment, each of them going through the four rounds of game against the algorithm. Considering the *Control* and *Others* treatments, we have 3860 observations in total.

The proportion of each answer given by subjects in Part 1 is not significantly different in *Control* and *Others*, except for slightly fewer subjects who do not have children and more subjects who prefer Vanilla ice cream in *Others*.

2.3 Analysis

We use the same experimental outcomes as in the main experiment, namely the frequency of optimal strategy and the number of hidden answers.

First, the frequencies with which subjects hide different number of answers is given in Figure 10. Over all observations, subjects hide on average 1.95 answers in *Others* and 1.88 in *Control*

Table 11: Characteristics of subjects in the *Control* and *Others* treatments

	Control	*Others*	Total	Diff. Control vs. Others (p-values)
ICE (% of Vanilla)	47.59	51.02	49.32	0.033
MUS (% of 3h+)	61.84	63.11	62.48	0.415
MAR (% of Yes)	54.30	56.56	55.44	0.158
NUC (% of Yes)	57.02	55.33	56.17	0.289
GEN (% of Male)	50.73	50.20	50.46	0.743
CHI (% of Yes)	53.04	50.20	51.60	0.078
Stats (% of Yes)	45.28	51.02	48.18	< 0.001
Age (mean)	41.92	42.20	42.06	0.523

($p = 0.142$). The distributions of these frequencies are not significantly different between *Control* and *Others* according to the Kolmogorov-Smirnov test ($p = 0.665$). This findings invalidate Hypothesis 1 (Others).

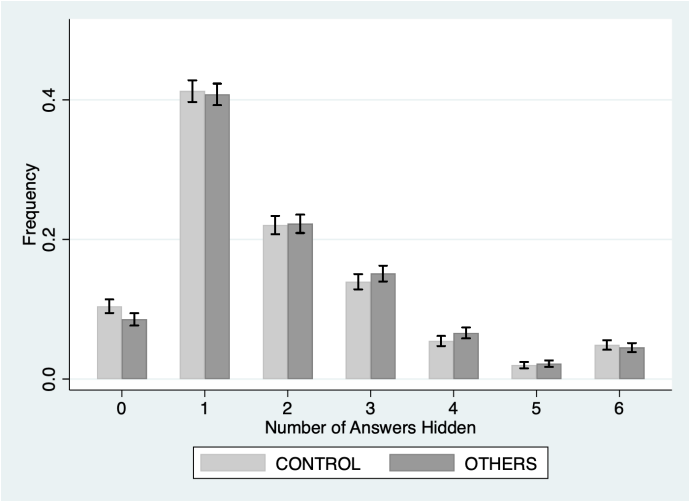

Figure 10: Hiding 0 to 6 answers, per treatment and pooling all targets

Second, over all observations, subjects play the optimal strategy 37.26% of the times in *Control* against 36.48% in *Others*, which is not significantly different ($p = 0.612$). The absence of effect of the *Others* treatment compared to the *Control* is confirmed by the regressions presented in Table 12. When looking at each target question separately, we also observe no significant difference in the frequency of optimal strategy in *Control* and *Others* except for NUC (p-values for ICE, MUS, MAR and NUC are 0.924, 0.965, 0.631 and 0.024 respectively).

Result 1 (Others) Pooling all target questions, there is no significant effect of the *Others* treatment compared to the *Control* treatment on the number of hidden answers or on the frequency of optimal strategy.

Table 12: Optimal Strategies - All Targets

	Optimal Strategy		
	(1)	(2)	(3)
Others	-0.008	-0.008	-0.010
	(0.022)	(0.022)	(0.022)
Round		0.024***	0.024***
		(0.006)	(0.006)
Stats			0.043*
			(0.022)
Female			-0.002
			(0.022)
Age			-0.001
			(0.001)
Constant	0.373***	0.312***	0.331***
	(0.016)	(0.021)	(0.043)
Observations	3860	3860	3860

Note: The table reports OLS coefficients (standard errors, clustered by ID, appear in parentheses).
* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$.

3 Instructions

3.1 Questionnaire of the pre-study

You are about to answer 30 questions about yourself. There is no right or wrong answer. **Please, read the questions carefully and answer honestly.** Your answers are important to our study.

What gender are you currently? *Male, Female, Non-binary*
When were you born? *After 1990, Before 1990*
Have you completed college? *Yes, No*
Does religion play an important role in your life? *Yes, No*
Do you have children? *Yes, No*
Do you speak a language other than English? *Yes, No*
Do you live in a city (more than 50,000 inhabitants)? *Yes, No*
Are you married or in a domestic partnership? *Yes, No*
Have you lived in the United States your whole life? *Yes, No*

How many pairs of shoes do you currently have? *More than 5 pairs, 5 pairs or fewer*
Do you spend more than 2 hours a week playing video games? *More than 2 hours per week, 2 hours per week or less*
Do you currently rent or own your home? *Rent, Own*
Are you in favor of the use of nuclear power? *Yes, No*
Would you say that your social circle is culturally diverse? *Yes, No*
Which of these two news outlets do you consider more credible? *CNN, Fox News*
Do you lean closer to the Democrat or Republican party? *Democrat, Republican*
When traveling for vacation, which do you prefer? *International, Domestic*
Other than for grocery shopping, would you say that you buy more often online or in-store? *Online, In-store*
Do you prefer team or individual sports? *Team, Individual*
Which flavor of ice cream do you prefer? *Chocolate, Vanilla*
Which describes your personality better? *Introvert, Extrovert*
Would you rather have dinner with friends at home or in a restaurant? *Restaurant, Home*
Do you smoke or vape? *Yes, No*
Which form of communication do you prefer? *Text messages, Phone calls*
What do you watch more, movies or TV shows? *TV shows, Movies*
How many active social media accounts do you have? *2 accounts or fewer, More than 2 accounts*
Are you in favor of the legalization of marijuana for recreational use? *Yes, No*
How much time do you spend listening to music per week? *3 hours or less, More than 3 hours*
When making important purchases, do you usually use price comparison websites? *Yes, No*
Are both of your parents born in the United States? *Yes, No*

3.2 Instructions for the Main Experiment

The text in italic corresponds to indications for the reader; these indications were not seen by subjects.

We are researchers at Sciences Po, France. We are currently studying how humans interact with computer programs.

This study has three parts:

- In **Part 1**, you will complete a **questionnaire about yourself** (including questions about gender, habits, and preferences). For completing this questionnaire, you will receive £1.2.
- In **Part 2**, you will play a **game against an algorithm / a computer program**. In this

part, you can gain between £0 and £3.2 depending on your decisions.

- ◇ In **Part 3**, you will answer some **final questions**. In this part, you can gain between £0 and £0.4 depending on your answers.

The study should last about 7 minutes.

—— *New Screen* ——

Part 1 - Questionnaire about yourself

You are about to answer six questions about yourself. There is no right or wrong answer. **Please, read the questions carefully and answer honestly.** Your answers are important to our study. You will get £1.2 for completing the questionnaire.

—— *New Screen* ——

Questions are presented in one of three orders, randomly selected.

Part 1 - Questionnaire about yourself

What gender are you currently?

Answers: Male, Female, Non-binary

Do you have children?

Answers: Yes, No

Are you in favor of the use of nuclear power?

Answers: Yes, No

How much time do you spend listening to music per week?

Answers: 3 hours or less, More than 3 hours

Are you married or in a domestic partnership?

Answers: Yes, No

Which flavor of ice cream do you prefer?

Answers: Chocolate, Vanilla

—— *New Screen* ——

Part 2 - Game against an algorithm

You will now play **four rounds of a game against an algorithm.** You can gain money in each round. At the end of the experiment, we will randomly select one of the four rounds and give you the money you gained in this round. Thus, it is in your interest to **try to gain as much money as possible in every round.**

The four rounds are independent of each other: **what you do in one round does not affect the following rounds.**

—— *New Screen* ——

Subjects are randomly assigned to the Control or to the Info treatment

Part 2 - Game against an algorithm

In every round of the game, you play against an algorithm. The algorithm does not know the answers you gave in Part 1 but it has been **programmed to guess these answers.**

In every round, your objective is to **prevent the algorithm from correctly guessing your answer to one specific question, the "target question".** Said differently, in every round, you must prevent the algorithm from learning one specific thing about you.

In every round, you will have to decide, for each answer you gave in Part 1, whether you want to DISCLOSE it or HIDE it to the algorithm. **The algorithm will use the answers you disclose to deduce your answer to the target question.**

Additional text seen by subjects in the Info treatment only

To make this deduction, the algorithm has been trained on 500 subjects, who previously completed the same questionnaire as the one you completed in Part 1. The algorithm uses their answers to identify correlations between answers. For example, it can identify whether women are more or less likely than men to listen to more than 3 hours of music per week.

—— *New Screen* ——

Part 2 - Game against an algorithm

How can I gain money in a round?

In every round, **you start with £3.2.**

This amount will be reduced:

1. **every time you hide one of your answers to the algorithm.**
2. **when the algorithm guesses more correctly your answer to the target question.**

Let us explain in more details:
1. For every answer that you hide to the algorithm, the £3.2 are reduced by £0.2.

Example: if you hide two answers, the £3.2 are reduced by £0.4. If you hide all answers, the £3.2 are reduced by £1.2.

2. After you have decided which answers to hide or disclose, the algorithm makes a guess of your answer to the target question. Your payment is lower when this guess is more accurate. Precisely, the algorithm attributes a probability to your answer to the target question. The £3.2 are reduced by two times this probability.

Example: Imagine that the target question is "Which flavor of ice cream do you prefer?" and you answered "Vanilla". If the algorithm attributes a probability of 40% to the answer "Vanilla", the £3.2 are reduced by 2*0.40 = £0.80. If the algorithm attributes a probability 100% to the answer "Vanilla" (the algorithm guesses your answer perfectly), the £3.2 are reduced by 2*1.00 = £2.

—— *New Screen* ——

Part 2 - Game against an algorithm

Before you start playing, please select the true statements below. It is for us to check you understood the game.

In every round of the game:
Possible answers: It is costly to hide my answers to the algorithm OR It is costly to disclose my answers to the algorithm.

The different rounds of the game are:

Possible answers: Independent from each other OR Dependent on each other.

In every round, my objective is:
Possible answers: To prevent the algorithm from guessing correctly my answer to the target question OR To help the algorithm guess my answer to the target question.

Regarding Part 2 of this study:
Possible answers: I will be paid for every round of the game OR One of the four rounds will be picked at random for payment.

Subjects had to answer correctly to the 5 questions to move to the next step.

—— *New Screen* ——

The four target questions are presented in random order. We only present instructions for round 1 below, as every round has the exact same structure.

You are now going to play round 1.

—— *New Screen* ——

Part 2 - Game against an algorithm
The target question is: **Are you in favor of the use of nuclear power?**
Your task is to prevent the algorithm from guessing your answer was **Yes**.
Now you can decide which of your answers you want to disclose to the algorithm and which of your answers you want to hide.
In what follows, the subject is always reminded of the answers he/she gave in Part 1.

What gender are you currently?
You answered Female
Subject chooses whether to disclose or hide this answer

Do you have children?
You answered Yes
Subject chooses whether to disclose or hide this answer

Are you in favor of the use of nuclear power?

You answered No

Subject chooses whether to disclose or hide this answer

How much time do you spend listening to music per week?

You answered More than 3 hours

Subject chooses whether to disclose or hide this answer

Are you married or in a domestic partnership?

You answered Yes

Subject chooses whether to disclose or hide this answer

Which flavor of ice cream do you prefer?

You answered Vanilla

Subject chooses whether to disclose or hide this answer

—— *New Screen* ——

Part 3 - Final Questions

There are four questions on this screen and each question has only one correct answer. For every correct answer you give, you will gain £0.10.

Imagine you have to guess someone's answer to the question Are you in favor of the use of nuclear power?.

To make this guess, if you could see this person's answer to one other question, which one would be most useful?

Possible answers: What gender are you currently?, How much time do you spend listening to music per week?, Are you married or in a domestic partnership?, Which flavor of ice cream do you prefer?, Do you have children?, None of the above questions would help me much to make my guess

Imagine you have to guess someone's answer to the question Which flavor of ice cream do you prefer?.

To make this guess, if you could see this person's answer to one other question, which one would be most useful?

Possible answers: What gender are you currently?, How much time do you spend listening to music per week?, Are you married or in a domestic partnership?, Are you in favor of the use of nuclear power?, Do you have children?, None of the above questions would help me much to make my guess

Imagine you have to guess someone's answer to the question Are you married or in a domestic partnership?.
To make this guess, if you could see this person's answer to one other question, which one would be most useful?
Possible answers: What gender are you currently?, How much time do you spend listening to music per week?, Are you in favor of the use of nuclear power?, Which flavor of ice cream do you prefer?, Do you have children?, None of the above questions would help me much to make my guess

Imagine you have to guess someone's answer to the question How much time do you spend listening to music per week?.
To make this guess, if you could see this person's answer to one other question, which one would be most useful?
Possible answers: What gender are you currently?, Are you in favor of the use of nuclear power?, Are you married or in a domestic partnership?, Which flavor of ice cream do you prefer?, Do you have children?, None of the above questions would help me much to make my guess

—— *New Screen* ——

Subject answers a final questionnaire. They report (i) their age, (ii) their level of education, (iii) whether they ever took a course in statistics, (iv) whether they noticed that websites sometimes propose contents that match their interests, (v) whether they know that online platforms collect data about them to make recommendations, (vi) whether they feel uncomfortable with the amount of data that online platforms collect about them, (vii) whether they sometimes actively try to limit the amount of data that online platforms collect about them, and (viii) whether they have ever tried to "game" personalized recommendation systems by giving false information or clicking on items they did not intend to buy.

—— *New Screen* ——

One of the four round is randomly selected for payment in Part 2.

The study is over. Thanks !

You have gained £1.2 in Part 1 and £x in Part 3. Round y of the game has been selected for your payment in Part 2. We will compute the payment for Part 2 in the next 48 hours and will pay you through the Prolific platform.

Please click to be redirected to the Prolific platform.

Redirection to the Prolific website.

4 The codes

4.1 The Naive Bayes Algorithm

In this section, we give the Python code of the Naive Bayes algorithm that we use in the experiment. The code is used to predict the probability of the two possible answers to a target question based on a set of disclosed answers. The function accepts a training dataset d (the pre-study data), a target question t, and a vector of disclosed answers v. When no answers are disclosed, the algorithm outputs the frequencies in the pre-study data.

```
from sklearn.naive_bayes import BernoulliNB
from sklearn.preprocessing import LabelEncoder



def predict_proba(d, t, v):
    """
    Algorithm that uses a Naive Bayes Classifier to predict the probability of
    ↪  answering one answering one answer in target column,
    given a vector of disclosed answers.
    :param d: The training data. Pandas dataframe
    :param t: The target column, it is the name of the target column. String
    :param v: The disclosed answers. It is a dataframe with one row and as many
    ↪  columns as disclosed answers. Names of
    the columns need to match those of d.
    :return: A dictionary that has as keys both possible answers to target question
    ↪  t, and the respective probability
    output by the Naive Bayes Classifier
    """
    # If v is empty, return the prior probabilities
```

```
    if v.empty:
        prior_prob = d[t].value_counts(normalize=True).to_dict()
        return prior_prob

    # Recode binary variables with LabelEncoder
    le = LabelEncoder()
    d_sub = d[v.columns].copy()
    for col in v.columns:
        if len(d[col].unique()) == 2:
            col_idx = d_sub.columns.get_loc(col)
            d_sub[d_sub.columns[col_idx]] = le.fit_transform(d_sub.loc[:, col])
            v[v.columns[col_idx]] = le.transform(v.loc[:, col])

    # Train the Bernoulli Naive Bayes model
    nb = BernoulliNB()
    nb.fit(d_sub, d[t])

    # Make predictions for the test data
    yproba = nb.predict_proba(v)

    # Create a dictionary to store the probabilities
    prob_dict = {}

    # Get the two possible string values of the target column
    target_vals = d[t].unique()

    # Loop through the target values and add probabilities to dictionary
    for val in target_vals:
        t_index = nb.classes_.tolist().index(val)
        prob_dict[val] = yproba[0, t_index]

    return prob_dict
```

4.2 The RATIO procedure

We remind that the ratio procedure is used to compare the payoff obtained when disclosing a set D and a set D' strictly contained in D. Precisely, this procedure does this comparison by computing $r(D, D') = \frac{g_D - g_{D'}}{|A \setminus D'| - |A \setminus D|}$, where g_D and $g_{D'}$ are the algorithm guesses with D and D', and comparing

this number to 0.1. For each Proposition (1,2,3 and 4 in the main text), the proof requires considering different sets D and D' for different subjects characterized by sets of answers A. In the procedure code, we can input constraints on A, D and D' by using the following parameters:

- ◇ *ty*: this "type" parameter equals "common" or "uncommon" depending on whether one wants to restrict attention to sets of answers A characterizing common or uncommon subjects.
- ◇ *fixdev*: fix a set of answers that are always included in D and excluded of D'.
- ◇ *always_disclose*: fix a set of answers that are always included in both D and D'.
- ◇ *always_hide*: fix a set of answers that are always excluded in both D and D'.

By combining values of these parameters, we create three modes - MODE 1, MODE 2, MODE 3 - that are used in the proofs presented in subsection 4.3.

MODE 1: This mode is used to show that a given disclosure set D' dominates another given disclosure set D, where $D' \subset D$. To activate this mode:

(a) Set the answers that are included both in D and D' as *always_disclose*.

(b) Set the answers that are excluded both of D and D' as *always_hide*.

(c) Set the set of answers that are included in D and excluded of D' as *fixdev*.

MODE 2: This mode is used to show that for all disclosure sets D which contain a given set of answers T, the disclosure set $D' = D \setminus T$ dominates D. To activate this mode:

(a) Leave *always_hide* and *always_disclose* empty.

(b) Set the set of answers T as *fixdev*.

MODE 3: This mode is used to show that a given disclosure set D dominates all disclosure set $D' \subset D$. This mode also has the option to keep a fixed set of answers disclosed both in D and D'. To activate this mode:

(a) Set the answers that are included both in D and D' as *always_disclose* (optional).

(b) Set the answers that are excluded both of D and D' as *always_hide*.

(c) Leave *fixdev* empty.

4.3 The code for RATIO

Below is the general code of the procedure. The instructions on how to run the code for each of the proofs are given in subsection 4.3.

```python
import numpy as np
import itertools
import pandas as pd
import Algorithm
from itertools import combinations


def generate_all_profiles(df):
    """
    Generate all possible combinations of values (sets of answers) for each column (question) in the given DataFrame.

    Args:
    df (pandas.DataFrame): The DataFrame for which to generate all possible sets of answers.

    Returns:
    pandas.DataFrame: A DataFrame containing all possible sets of answers.
    """
    # Get sorted unique values for each column
    unique_values = [sorted(df[col].unique()) for col in df.columns]

    # Generate all possible combinations of these values
    all_profiles = np.array(list(itertools.product(*unique_values)))

    # Return these combinations as a DataFrame
    return pd.DataFrame(all_profiles, columns=df.columns)


def determine_type(row, target_col):
    """
    Determine the type of a row based on the target column and specific relationships within the row.

    Args:
    row (pandas.Series): A single row from a DataFrame, representing a profile.
    target_col (str): The target column based on which the type is determined.
```

```
35
36      Returns:
37      str: The determined type of the row. Possible values are 'Uncor', 'Common',
        ↪  'Uncommon', or 'Other'.
38      """
39      # If the target column is either 'ICE' or 'MUS'
40      if target_col in ['ICE', 'MUS']:
41          # These columns are considered 'Uncorrelated'
42          return 'Uncor'
43
44      # If the target column is 'MAR'
45      elif target_col == 'MAR':
46          # Return 'Uncommon' if the values in 'MAR' and 'CHI' columns are different,
            ↪  else 'Common'
47          return 'Uncommon' if row['MAR'] != row['CHI'] else 'Common'
48
49      # If the target column is 'NUC'
50      elif target_col == 'NUC':
51          # Return 'Uncommon' if the values in 'NUC' and 'GEN' columns are different,
            ↪  else 'Common'
52          return 'Uncommon' if row['NUC'] != row['GEN'] else 'Common'
53
54
55  def procedure(target_col, df, ty=None, fixdev=None, always_disclose=None,
    ↪  always_hide=None):
56      """
57      Find the optimal strategy by computing the ratio r(D, D') for disclosure
        ↪  strategies D and D' (where D' is a subset of D).
58      A ratio higher than 0.1 indicates that D' dominates D.
59
60      Args:
61      target_col (str): The target column for which the algorithm predicts a
        ↪  probability.
62      df (pandas.DataFrame): The DataFrame used for calculations, containing all
        ↪  possible sets of answers A.
63      ty (str, optional): Type of subjects to consider ('Common' or 'Uncommon').
        ↪  Defaults to None.
```

```
    fixdev (list, optional): Answers that are included in D but excluded in D'
    ↪ (fixed deviation). Defaults to None.
    always_disclose (list, optional): Answers that are always disclosed in both D
    ↪ and D'. Defaults to None.
    always_hide (list, optional): Answers that are always hidden in both D and D'.
    ↪ Defaults to None.

    Returns:
    pandas.DataFrame: The maximum ratio among all profiles if no fixdev is specified
    ↪ (we are showing that nothing
    dominates D). The minimum ratio if fixdev is specified (as we are showing that
    ↪ fixdev is a beneficial deviation
    from D).
    """

    # Filter out the target column
    columns = [col for col in df.columns]

    # Generate all unique profiles from the DataFrame
    unique_profiles = generate_all_profiles(df)

    # Initialize an empty DataFrame to store results
    result_df = pd.DataFrame(columns=['Profile', 'Initial_P', 'Lowest_P', 'Diff',
    ↪ 'N', 'Ratio', 'Hidden_Cols', 'Type'])

    # Filter columns for additional hiding, excluding always_hide and
    ↪ always_disclose
    additional_hide_columns = [col for col in columns if
                               col not in (always_hide if always_hide else []) and
                               ↪ col not in (
                                   always_disclose if always_disclose else [])]

    # Iterate through each unique profile
    for index, row in unique_profiles.iterrows():

        # MODE 3: Prove that no strategy D' with the constraints always_hide and
        ↪ always_disclose is better than D given
```

```
        # by the whole profile except always_hide.
        if fixdev is None:
            # Initial probability calculation with always disclosed columns
            initial_columns = additional_hide_columns + (always_disclose if
            ↪  always_disclose else [])
            v = row[initial_columns].to_frame().T
            initial_prob_dict = Algorithm.predict_proba(df, target_col, v)
            initial_p = initial_prob_dict[row[target_col]]

            # Determine the 'Type' based on specific criteria
            type_val = determine_type(row, target_col)

            # Variables to track the best strategy
            best_ratio, best_hidden_columns, best_diff, best_n, best_p = 0, [], 0,
            ↪  0, 0

            # Check all combinations of hiding columns
            for r in range(len(additional_hide_columns) + 1):
                for hidden_cols in combinations(additional_hide_columns, r):
                    # Calculate the probability for the current strategy
                    total_hidden = list(hidden_cols) + (always_hide if always_hide
                    ↪  else [])
                    revealed_columns = [col for col in initial_columns if col not in
                    ↪  total_hidden]
                    v = row[revealed_columns].to_frame().T
                    prob_dict = Algorithm.predict_proba(df, target_col, v)
                    p = prob_dict[row[target_col]]
                    diff = initial_p - p
                    n = len(total_hidden)
                    n2 = len(always_hide)
                    ratio = diff / (n - n2) if (n - n2) > 0 else 0

                    # Update the best strategy if a better one is found
                    if ratio >= best_ratio:
                        best_ratio = ratio
                        best_hidden_columns = total_hidden
                        best_diff = diff
```

```
                        best_n = n
                        best_p = p

        # MODE 2 : Prove that for any strategy D with no constraints, the deviation
        ↪  D' = D - fixdev always dominates
        # D
        else:
            if always_disclose is None and always_hide is None:
                # Determine the 'Type' based on specific criteria
                type_val = determine_type(row, target_col)
                c = [col for col in df.columns if col not in fixdev]
                # Variables to track the best strategy
                best_ratio, best_hidden_columns, best_diff, best_n, best_p = 1, [],
                ↪  0, 0, 0
                for r in range(len(c) + 1):
                    for hidden_cols in combinations(c, r):
                        total_hidden = list(hidden_cols)
                        revealed_columns = [col for col in df.columns if col not in
                        ↪  total_hidden]
                        v = row[revealed_columns].to_frame().T
                        prob_dict = Algorithm.predict_proba(df, target_col, v)
                        p = prob_dict[row[target_col]]

                        total_hidden_b = list(hidden_cols) + fixdev
                        revealed_columns_b = [col for col in df.columns if col not
                        ↪  in total_hidden_b]
                        v_b = row[revealed_columns_b].to_frame().T
                        prob_dict_b = Algorithm.predict_proba(df, target_col, v_b)
                        p_b = prob_dict_b[row[target_col]]

                        diff = p - p_b
                        n = len(total_hidden)
                        n_b = len(total_hidden_b)
                        ratio = diff / (n_b - n) if (n_b - n) > 0 else 0

                        if ratio < best_ratio:
                            best_ratio = ratio
```

```
                        best_hidden_columns = total_hidden_b
                        best_diff = diff
                        best_n = n_b
                        best_p = p_b
                        initial_p = p

        # MODE 1 : Prove that for a given strategy D = always_disclose + fixdev -
        ↪  always_hide, the deviation
        # D' = always_disclose - fixdev - always_hide always dominates D
            else:
                # Determine the 'Type' based on specific criteria
                type_val = determine_type(row, target_col)

                # Initial probability calculation with always disclosed columns
                initial_columns = additional_hide_columns + (always_disclose if
                ↪  always_disclose else [])
                v = row[initial_columns].to_frame().T
                initial_prob_dict = Algorithm.predict_proba(df, target_col, v)
                initial_p = initial_prob_dict[row[target_col]]

                total_hidden = (always_hide if always_hide else []) + fixdev
                revealed_columns = [col for col in initial_columns if col not in
                ↪  total_hidden]
                v = row[revealed_columns].to_frame().T
                prob_dict = Algorithm.predict_proba(df, target_col, v)
                p = prob_dict[row[target_col]]
                diff = initial_p - p
                n = len(total_hidden)
                n2 = len(always_hide)
                ratio = diff / (n - n2) if (n - n2) > 0 else 0
                best_ratio = ratio
                best_hidden_columns = total_hidden
                best_diff = diff
                best_n = n
                best_p = p

        # Add the result of the current profile to the result DataFrame
```

```
        result_row = {
            'Profile': index,
            'Type': type_val,
            'Initial_P': initial_p,
            'Lowest_P': best_p,
            'Diff': best_diff,
            'N': best_n,
            'Ratio': best_ratio,
            'Hidden_Cols': best_hidden_columns
        }
        result_df = pd.concat([result_df, pd.DataFrame([result_row])],
        ↪  ignore_index=True)

    # Filter results based on the specified type (ty)
    if ty == "Common":
        ratio_minmax = [min(result_df[result_df["Type"] == "Common"]["Ratio"]),
                        max(result_df[result_df["Type"] == "Common"]["Ratio"])]
    elif ty == "Uncommon":
        ratio_minmax = [min(result_df[result_df["Type"] == "Uncommon"]["Ratio"]),
                        max(result_df[result_df["Type"] == "Uncommon"]["Ratio"])]
    else:
        ratio_minmax = [min(result_df["Ratio"]), max(result_df["Ratio"])]

    # Return the result DataFrame
    return ratio_minmax[1] if fixdev is None else ratio_minmax[0]
```

4.4 Applications of the procedure

4.4.1 Proposition 1

```
# Show, for each target question, that it is always strictly beneficial for the
↪  subjects to hide the answer to the target question (MODE 2)
procedure("ICE", df, fixdev=["ICE"])
procedure("MUS", df, fixdev=["MUS"])
procedure("MAR", df, fixdev=["MAR"])
procedure("NUC", df, fixdev=["NUC"])
```

4.4.2 Proposition 2(a)

```
# Show, for each uncorrelated target question, that it is optimal for every subject to hide only the answer to the target question (MODE 3)
procedure("ICE", df, always_hide=["ICE"])
procedure("MUS", df, always_hide=["MUS"])
```

4.4.3 Proposition 2(b)

```
# Show, for each correlated target question, that it is optimal for every common subject to hide exactly two answers: the answer to the target question and the answer to its correlated question (resp. CHI or GEN).

# Show that no strategy that discloses CHI is better than hiding only MAR (MODE 3)
procedure("MAR", df, ty="Common", always_disclose=["CHI"], always_hide=["MAR"])
# Show that the strategy that hides only CHI is better than hiding only MAR (MODE 1)
procedure("MAR", df, ty="Common", always_disclose=["NUC", "ICE", "GEN", "MUS"], always_hide=["MAR"], fixdev = ["CHI"])
# Show that no strategy that hides MAR, CHI and other answers is better than hiding only MAR and CHI (MODE 3)
procedure("MAR", df, ty="Common", always_hide=["MAR", "CHI"])

# Show that no strategy that discloses GEN is better than hiding only NUC (MODE 3)
procedure("NUC", df, ty="Common", always_disclose=["GEN"], always_hide=["NUC"])
# Show that the strategy that hides only GEN is better than hiding only NUC (MODE 1)
procedure("NUC", df, ty="Common", always_disclose=["MAR", "ICE", "GEN", "MUS"], always_hide=["NUC"], fixdev = ["GEN"])
# Show that no strategy that hides NUC, GEN and other answers is better than hiding only NUC and GEN (MODE 3)
procedure("NUC", df, ty="Common", always_hide=["NUC", "GEN"])
```

4.4.4 Proposition 2(c)

```
# Show, for each correlated target question, that it is optimal for every
↪ uncommon subject to hide only the answer to the target question (MODE 3)
procedure("MAR", df, ty="Uncommon", always_hide=["MAR"])
procedure("NUC", df, ty="Uncommon", always_hide=["NUC"])
```