Causal_Inference

November 28, 2023

[ ]:

```
!apt install libgraphviz-dev
!pip install pygraphviz
!pip install dowhy
!pip install pycaret[full]
```

[ ]:

```
class ForwardChainingEngine:
    def __init__(self, rules):
        self.rules = rules
        self.facts = {}

    def _count_equal_conditions_matched(self, conditions):
        return sum(1 for condition in conditions if not condition.get('any') and
                   self.facts.get(condition['key']) == condition['value'])

    def _count_any_conditions_matched(self, conditions):
        count = 0
        for condition in filter(lambda c: c.get('any'), conditions):
            values = self.facts.get(condition['any'][0]['key'])
            condition_values = [c['value'] for c in condition['any']]
            if values and any(v in condition_values for v in values):
                count += 1
        return count

    def _get_similarity(self, rule):
        return (self._count_equal_conditions_matched(rule['conditions']) +
                self._count_any_conditions_matched(rule['conditions']))

    def _applicable_rules(self):
        applicable_rules = [{
            'rule': rule,
            'conditionsMatched': self._get_similarity(rule),
            'recommendation': rule['action']['recommendation']
        } for rule in self.rules]

        # Ordenar reglas aplicadas
        return sorted(applicable_rules, key=lambda r: r['conditionsMatched'],␣
 ↪reverse=True)
```

```python
    def run(self, initial_facts):
        self.facts = initial_facts.copy()
        return self._applicable_rules()

rules = [
  {"conditions": [
    {"key": "cuerpo", "value": "Completo"},
    {"key": "color", "value": "Negra"},
    {"key": "malta", "value": "Negra"},
    {"key": "IBU", "value": 4},
    {"key": "ABV", "value": 3},
    {"any": [
      {"key": "maridaje", "value": "Carnes rojas"},
      {"key": "maridaje", "value": "Sola"}
    ]}
  ], "action": {"recommendation": "Stout"}},

  {"conditions": [
    {"key": "color", "value": "Clara"},
    {"key": "cuerpo", "value": "Ligero"},
    {"key": "malta", "value": "Pálida"},
    {"key": "IBU", "value": 1},
    {"key": "ABV", "value": 2},
    {"any": [
      {"key": "maridaje", "value": "Carnes blancas"},
      {"key": "maridaje", "value": "Salado"},
      {"key": "maridaje", "value": "Sola"}
    ]}
  ], "action": {"recommendation": "Lager"}},

  {"conditions": [
    {"key": "color", "value": "Roja"},
    {"key": "cuerpo", "value": "Medio"},
    {"key": "malta", "value": "Caramelo"},
    {"key": "IBU", "value": 5},
    {"key": "ABV", "value": 5},
    {"any": [
      {"key": "maridaje", "value": "Carnes rojas"},
      {"key": "maridaje", "value": "Sola"}
    ]}
  ], "action": {"recommendation": "IPA"}},

  {"conditions": [
    {"key": "color", "value": "Rubia"},
    {"key": "cuerpo", "value": "Cremoso"},
    {"key": "malta", "value": "Caramelo"},
    {"key": "IBU", "value": 2},
    {"key": "ABV", "value": 5},
    {"any": [
      {"key": "maridaje", "value": "Carnes rojas"},
      {"key": "maridaje", "value": "Sola"}
    ]}
  ], "action": {"recommendation": "Honey"}},

  {"conditions": [
    {"key": "cuerpo", "value": "Medio"},
    {"key": "color", "value": "Roja"},
    {"key": "IBU", "value": 3},
    {"key": "ABV", "value": 1},
    {"key": "malta", "value": "Tostada"},
    {"any": [
      {"key": "maridaje", "value": "Salado"},
      {"key": "maridaje", "value": "Sola"}
    ]}
  ], "action": {"recommendation": "Ale sin alcohol"}},

  {"conditions": [
    {"key": "color", "value": "Rubia"},
    {"key": "cuerpo", "value": "Ligero"},
    {"key": "malta", "value": "Pálida"},
    {"key": "IBU", "value": 1},
    {"key": "ABV", "value": 2},
    {"any": [
      {"key": "maridaje", "value": "Carnes blancas"},
      {"key": "maridaje", "value": "Sola"}
    ]}
  ], "action": {"recommendation": "Rubia"}},

  {"conditions": [
    {"key": "color", "value": "Roja"},
    {"key": "cuerpo", "value": "Medio"},
    {"key": "malta", "value": "Tostada"},
    {"key": "IBU", "value": 3},
    {"key": "ABV", "value": 3},
    {"any": [
      {"key": "maridaje", "value": "Quesos"},
      {"key": "maridaje", "value": "Sola"}
    ]}
  ], "action": {"recommendation": "Ale Roja Irlandesa"}},

  {"conditions": [
    {"key": "color", "value": "Negra"},
    {"key": "cuerpo", "value": "Completo"},
    {"key": "malta", "value": "Chocolate"},
    {"key": "IBU", "value": 5},
    {"key": "ABV", "value": 3},
    {"any": [
      {"key": "maridaje", "value": "Carnes blancas"},
      {"key": "maridaje", "value": "Quesos"},
      {"key": "maridaje", "value": "Quesos"}
    ]}
  ], "action": {"recommendation": "Porter"}}
]


engine = ForwardChainingEngine(rules);

def getbeer(conditions):
  beers = engine.run(conditions)
  if beers: return beers[0]['recommendation']
  return None

def getmatches(conditions):
  beers = engine.run(conditions)
  if beers: return beers[0]['conditionsMatched']
  return None

# Ejemplo de uso del engine
conditions = {'cuerpo':'Completo', 'color': 'Negra', 'malta': 'Negra', 'IBU':␣
 ↪4, 'ABV': 3, 'maridaje': ['Sola']}
print(getbeer(conditions))
print(getmatches(conditions))
```

```
Stout
6
```

[ ]:

```
# Datos sinteticos: todas las posibles combinaciones de los atributos

import pandas as pd
import itertools

# Valores posibles de los atributos
colors = ["Clara", "Rubia", "Roja", "Negra"]
cuerpos = ["Ligero", "Medio", "Completo", "Cremoso"]
maltas = ["Pálida", "Caramelo", "Tostada", "Chocolate", "Negra"]
IBUs = [1,2,3,4,5]
ABVs = [1,2,3,4,5]
maridajes = ["Salado", "Torta", "Carnes rojas", "Carnes blancas", "Quesos",␣
 ↪"Sola"]
```

```python
# El atributo de maridajes es una lista, nos quedamos con solo listas de un elemento para
# quedarnos dentro de los < 15k datos.
nmaridajes = [1]
posibles_maridajes = [list(comb) for r in nmaridajes for comb in itertools.combinations(maridajes, r)]

all_combinations = list(itertools.product(colors, cuerpos, maltas, IBUs, ABVs, posibles_maridajes))
df = pd.DataFrame(all_combinations, columns=['color', 'cuerpo', 'malta', 'IBU', 'ABV', 'maridaje'])
df
```

```
[ ]:        color   cuerpo   malta  IBU  ABV          maridaje
     0      Clara   Ligero  Pálida    1    1          [Salado]
     1      Clara   Ligero  Pálida    1    1           [Torta]
     2      Clara   Ligero  Pálida    1    1     [Carnes rojas]
     3      Clara   Ligero  Pálida    1    1   [Carnes blancas]
     4      Clara   Ligero  Pálida    1    1          [Quesos]
     …      …        …       …    …   …              …
     11995  Negra  Cremoso   Negra    5    5           [Torta]
     11996  Negra  Cremoso   Negra    5    5     [Carnes rojas]
     11997  Negra  Cremoso   Negra    5    5   [Carnes blancas]
     11998  Negra  Cremoso   Negra    5    5          [Quesos]
     11999  Negra  Cremoso   Negra    5    5            [Sola]

     [12000 rows x 6 columns]
```

```python
[ ]: # Corremos el engine sobre todos nuestros datos
     df['cerveza'] = df.apply(lambda row: getbeer(row.to_dict()), axis=1)
     df['matches'] = df.apply(lambda row: getmatches(row.to_dict()), axis=1)
     df
```

```
[ ]:        color   cuerpo   malta  IBU  ABV          maridaje cerveza  matches
     0      Clara   Ligero  Pálida    1    1          [Salado]   Lager        5
     1      Clara   Ligero  Pálida    1    1           [Torta]   Lager        4
     2      Clara   Ligero  Pálida    1    1     [Carnes rojas]   Lager        4
     3      Clara   Ligero  Pálida    1    1   [Carnes blancas]   Lager        5
     4      Clara   Ligero  Pálida    1    1          [Quesos]   Lager        4
     …      …        …       …    …   …              …      …       …
     11995  Negra  Cremoso   Negra    5    5           [Torta]   Stout        2
     11996  Negra  Cremoso   Negra    5    5     [Carnes rojas]   Stout        3
     11997  Negra  Cremoso   Negra    5    5   [Carnes blancas]  Porter        3
     11998  Negra  Cremoso   Negra    5    5          [Quesos]  Porter        3
     11999  Negra  Cremoso   Negra    5    5            [Sola]   Stout        3

     [12000 rows x 8 columns]
```

```
[ ]: # Usamos de treatment los casos donde tenemos mas de 3 matches, y descartamos␣
      ↪la columna
     df['treatment'] = df['matches'] >= 3
     del df['matches']

     df['treatment'].value_counts()
```

```
[ ]: True     9066
     False    2934
     Name: treatment, dtype: int64
```

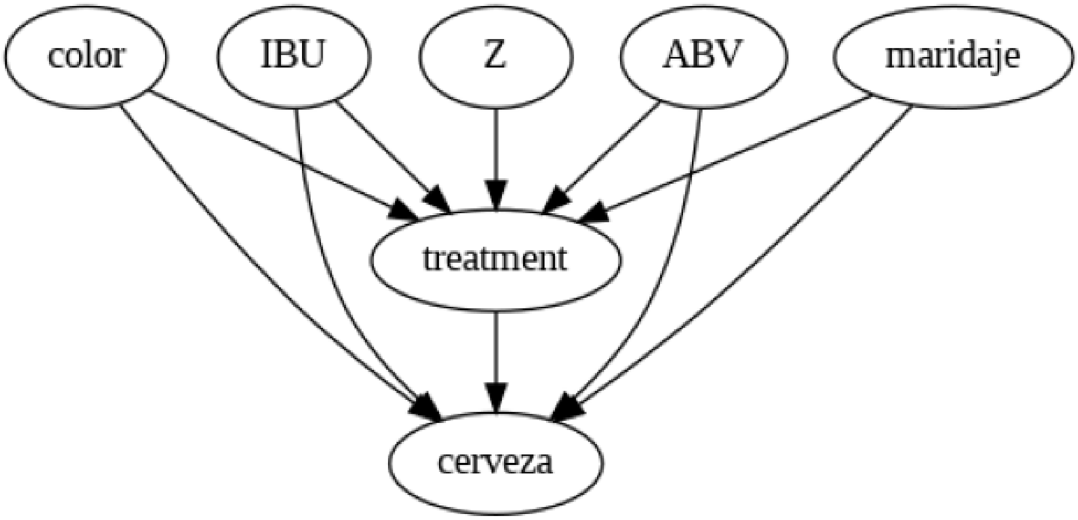
```
[ ]: from sklearn.preprocessing import LabelEncoder
     label_encoder = LabelEncoder()

     # Convertimos la lista de maridajes en un string
     df['maridaje'] = df['maridaje'].apply(lambda m: m[0] if len(m) == 1 else␣
      ↪str(sorted(m)))

     # Convertimos todas las variables categoricas en numeros
     for column in ['cuerpo', 'color', 'malta', 'maridaje', 'cerveza']:
       df[column] = label_encoder.fit_transform(df[column])

     df
```

```
[ ]:        color  cuerpo  malta  IBU  ABV  maridaje  cerveza  treatment
     0          0       2      3    1    1         3        4       True
     1          0       2      3    1    1         5        4       True
     2          0       2      3    1    1         1        4       True
     3          0       2      3    1    1         0        4       True
     4          0       2      3    1    1         2        4       True
     …      …     …     …  …  …      …      …        …
     11995      1       1      2    5    5         5        7      False
     11996      1       1      2    5    5         1        7       True
     11997      1       1      2    5    5         0        5       True
     11998      1       1      2    5    5         2        5       True
     11999      1       1      2    5    5         4        7       True

     [12000 rows x 8 columns]
```

```
[ ]: # TESIS original: el cuerpo de la cerveza influye en el ABV, y la malta influye␣
      ↪en el IBU y el color
     # por ende estas variables son redundantes y podrian ser removidas del grafo de␣
      ↪dependencias

     # Agregamos la variable Z externa -> "sospecho que debe haber otra cosa"
     import dowhy
```

```
causal_graph = """digraph {
                        Z->treatment;
                        color->treatment;
                        IBU->treatment;
                        ABV->treatment;
                        maridaje->treatment;
                        color->cerveza;
                        IBU->cerveza;
                        ABV->cerveza;
                        maridaje->cerveza;
                        treatment->cerveza;
                }"""
model = dowhy.CausalModel(data=df, treatment="treatment", outcome="cerveza",␣
 ↪graph=causal_graph)
model.view_model()
```

```
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 7 variables. 6 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'maridaje', 'treatment']'. The
unobserved variables are: '['Z']'. If this matches your expectations for
observations, please continue. If you expected any of the unobserved variables
to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 2 variables in the dataset
that are not in the graph. Variable names are: '['cuerpo', 'malta']'
```



```
[ ]: identified_estimand = model.identify_effect(proceed_when_unidentifiable=True)
     print(identified_estimand)
```

```
Estimand type: EstimandType.NONPARAMETRIC_ATE

### Estimand : 1
Estimand name: backdoor
Estimand expression:
     d
      (E[cerveza|maridaje,IBU,color,ABV])
d[treatment]
Estimand assumption 1, Unconfoundedness: If U→{treatment} and U→cerveza then
P(cerveza|treatment,maridaje,IBU,color,ABV,U) =
P(cerveza|treatment,maridaje,IBU,color,ABV)

### Estimand : 2
Estimand name: iv
Estimand expression:
                                   -1
  d              d
E   (cerveza)   ([treatment])
 d[Z]           d[Z]
Estimand assumption 1, As-if-random: If U→→cerveza then ¬(U →→{Z})
Estimand assumption 2, Exclusion: If we remove {Z}→{treatment}, then
¬({Z}→cerveza)

### Estimand : 3
Estimand name: frontdoor
No such variable(s) found!
```

```
[ ]: estimate = model.estimate_effect(identified_estimand,
                                      method_name='backdoor.
      ↪propensity_score_matching',
                                      target_units='att')
     print(estimate)
```

```
*** Causal Estimate ***

## Identified estimand
Estimand type: EstimandType.NONPARAMETRIC_ATE

### Estimand : 1
Estimand name: backdoor
Estimand expression:
     d
       (E[cerveza|maridaje,IBU,color,ABV])
d[treatment]
Estimand assumption 1, Unconfoundedness: If U→{treatment} and U→cerveza then
P(cerveza|treatment,maridaje,IBU,color,ABV,U) =
P(cerveza|treatment,maridaje,IBU,color,ABV)

## Realized estimand
b: cerveza~treatment+maridaje+IBU+color+ABV
Target units: att

## Estimate
Mean value: -1.342598720494154
```

[ ]:

```python
# Si el P value es alto, significa que nuestra tesis es un efecto placebo
# Si es  chico, la tesis es correcta
# (a ojo) bajo: < 0.2 - alto: >= 0.2
refutation = model.refute_estimate(identified_estimand,
                                   estimate,
                                   method_name='placebo_treatment_refuter',
                                   placebo_type='permute',
                                   num_simulations=20)
print(refutation)
```

```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:-0.012943966468122658
p value:0.4059750257188881
```

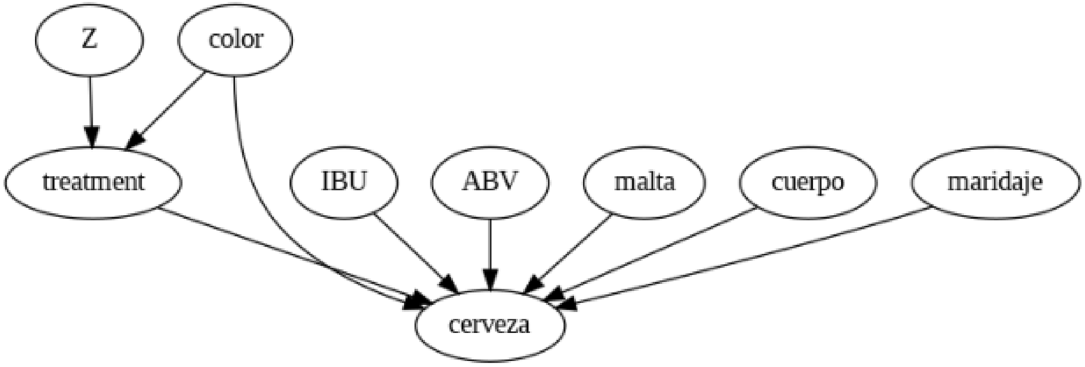
[ ]:

```python
# Corremos el modelo para cada una de las variables apuntando a treatment para␣
 ↪ver su significancia

for c in ['color', 'cuerpo', 'malta', 'maridaje', 'IBU', 'ABV']:
  graph = f"""
  digraph {{
              Z->treatment;
              color->cerveza;
              IBU->cerveza;
              ABV->cerveza;
              malta->cerveza;
              cuerpo->cerveza;
```

```python
            maridaje->cerveza;
            treatment->cerveza;
            {c}->treatment
}}
"""

print(f"TEST: {c}")

model = dowhy.CausalModel(data=df, treatment="treatment", outcome="cerveza", graph=graph)
model.view_model()
identified_estimand = model.identify_effect(proceed_when_unidentifiable=True)
estimate = model.estimate_effect(identified_estimand,
                               method_name='backdoor.propensity_score_matching',
                               target_units='att')
refutation = model.refute_estimate(identified_estimand,
                                 estimate,
                                 method_name='placebo_treatment_refuter',
                                 placebo_type='permute',
                                 num_simulations=20)
print(refutation)
```

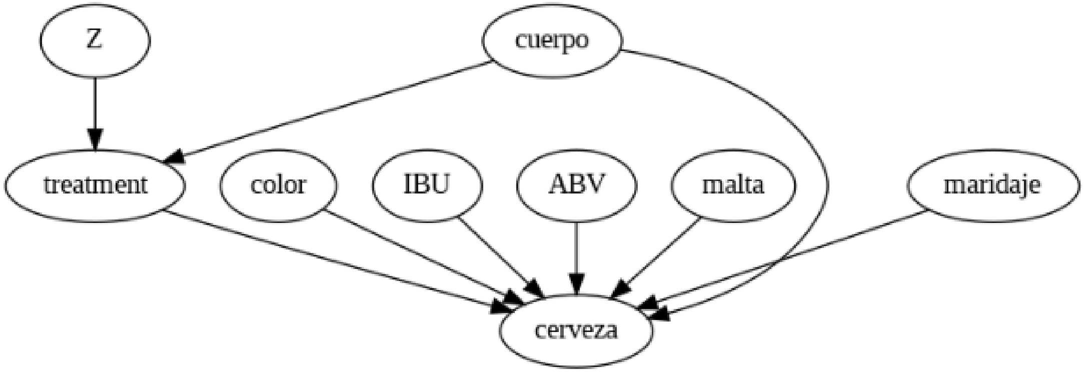
```
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 9 variables. 8 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'cuerpo', 'malta', 'maridaje',
'treatment']'. The unobserved variables are: '['Z']'. If this matches your
expectations for observations, please continue. If you expected any of the
unobserved variables to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 1 variables in the dataset
that are not in the graph. Variable names are: '['propensity_score']'
```

```
TEST: color
```



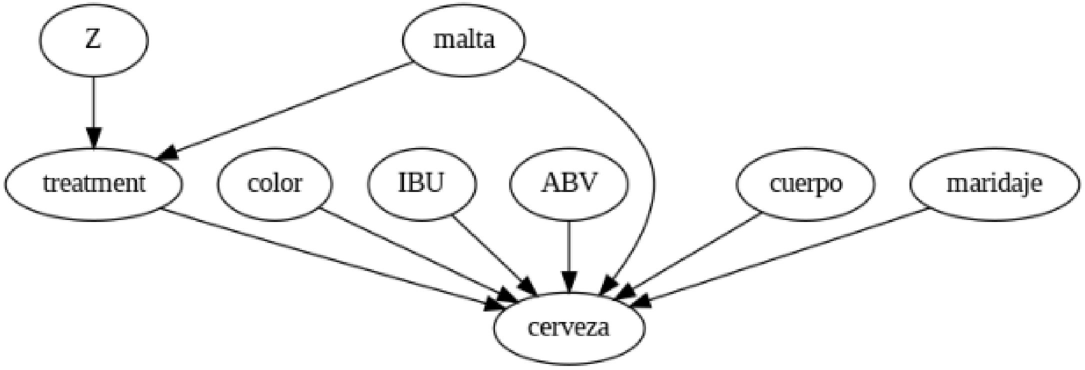
```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 9 variables. 8 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'cuerpo', 'malta', 'maridaje',
'treatment']'. The unobserved variables are: '['Z']'. If this matches your
expectations for observations, please continue. If you expected any of the
unobserved variables to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 1 variables in the dataset
that are not in the graph. Variable names are: '['propensity_score']'
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.004378998455768803
p value:0.47952148524809185

TEST: cuerpo
```



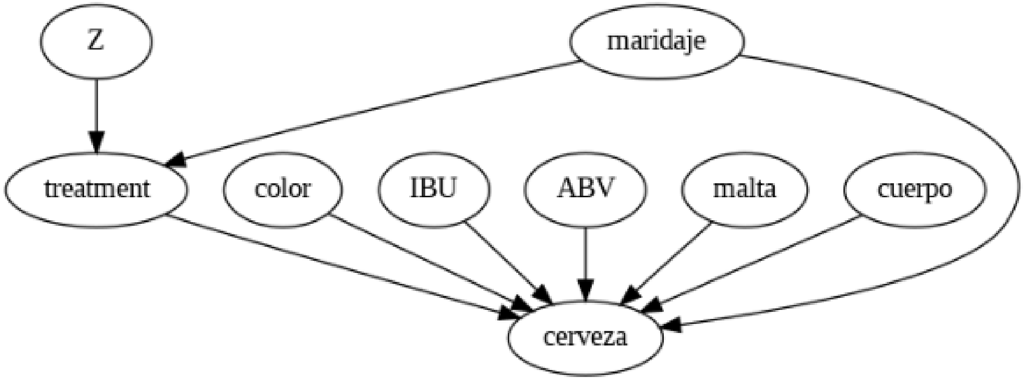
```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 9 variables. 8 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'cuerpo', 'malta', 'maridaje',
'treatment']'. The unobserved variables are: '['Z']'. If this matches your
expectations for observations, please continue. If you expected any of the
unobserved variables to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 1 variables in the dataset
that are not in the graph. Variable names are: '['propensity_score']'
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.0004908449150672846
p value:0.4956344296265684

TEST: malta
```



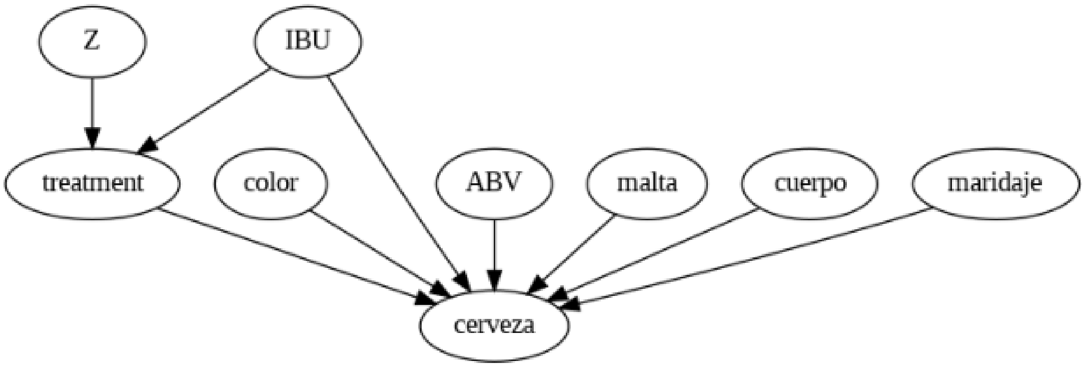
```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 9 variables. 8 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'cuerpo', 'malta', 'maridaje',
'treatment']'. The unobserved variables are: '['Z']'. If this matches your
expectations for observations, please continue. If you expected any of the
unobserved variables to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 1 variables in the dataset
that are not in the graph. Variable names are: '['propensity_score']'
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:-0.03165673946613721
p value:0.3518022251213839

TEST: maridaje
```

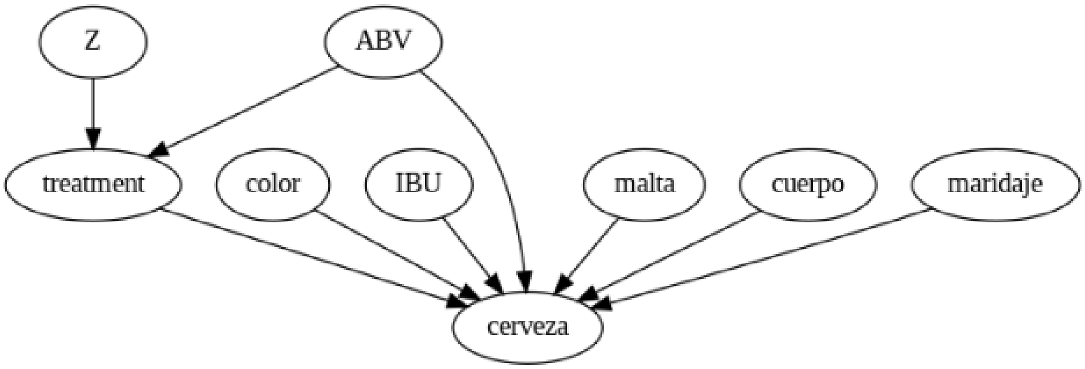


```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 9 variables. 8 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'cuerpo', 'malta', 'maridaje',
'treatment']'. The unobserved variables are: '['Z']'. If this matches your
expectations for observations, please continue. If you expected any of the
unobserved variables to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 1 variables in the dataset
that are not in the graph. Variable names are: '['propensity_score']'
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:-9.375689388925676e-05
p value:0.4994697953683437

TEST: IBU
```



```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 9 variables. 8 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'cuerpo', 'malta', 'maridaje',
'treatment']'. The unobserved variables are: '['Z']'. If this matches your
expectations for observations, please continue. If you expected any of the
unobserved variables to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 1 variables in the dataset
that are not in the graph. Variable names are: '['propensity_score']'

Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.018073020075005518
p value:0.3896449829554304

TEST: ABV
```



```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.010020957423339949
p value:0.43298021666749253
```

Los resultados muestran que ninguna de las variables individuales (color, cuerpo, malta, maridaje, IBU, ABV) tiene un efecto muy significativo en el tratamiento definido (coincidencia de preferencias de cerveza). Esto se evidencia por los p values relativamente altos obtenidos en las pruebas de refutación para cada variable. Estos p values, todos superiores a 0.35, indican que no hay una diferencia significativa entre los efectos estimados y los efectos obtenidos bajo un tratamiento placebo.

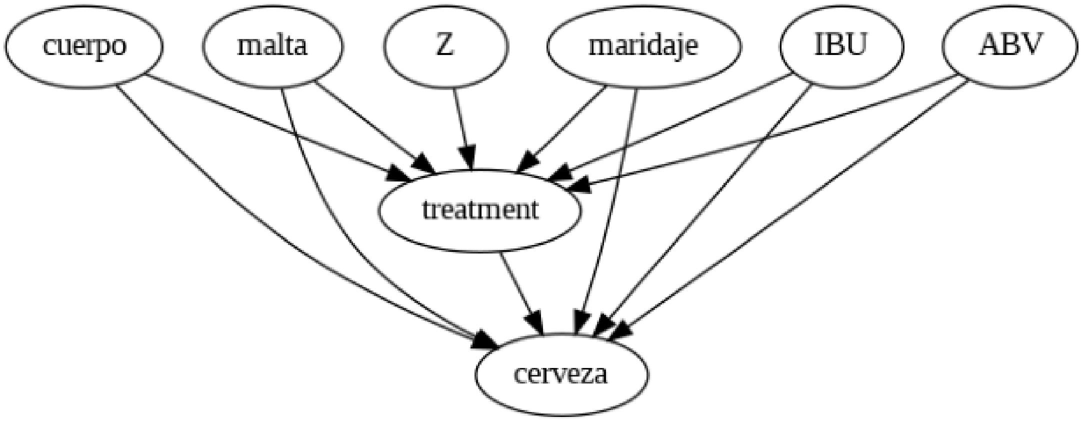
En conclusión, con base en este análisis, parece que ninguna de estas características de la cerveza, consideradas individualmente, tiene un impacto significativo en la probabilidad de que las recomendaciones del motor coincidan con las preferencias del usuario. Este resultado podría sugerir que la elección de la cerveza está influenciada por una combinación de factores o por otros factores no incluidos en el modelo actual.

[ ]:
```
# Corremos el modelo sin cada una de las variables para ver cuanto influencian␣
 ↪en la clase

attrs = ['color', 'cuerpo', 'malta', 'maridaje', 'IBU', 'ABV']
for c in attrs:
  graphattrs = attrs[:]
  graphattrs.remove(c)
  graph = f"""
  digraph {{
              Z->treatment;
              treatment->cerveza;
```

```
            {';'.join(map(lambda x: f"{x}->treatment", graphattrs))}
            {';'.join(map(lambda x: f"{x}->cerveza", graphattrs))}
}}
"""
print(f"TEST: {c}")

model = dowhy.CausalModel(data=df, treatment="treatment", outcome="cerveza",␣
↪graph=graph)
model.view_model()
identified_estimand = model.identify_effect(proceed_when_unidentifiable=True)
estimate = model.estimate_effect(identified_estimand,
                               method_name='backdoor.
↪propensity_score_matching',
                               target_units='att')
refutation = model.refute_estimate(identified_estimand,
                                 estimate,
                                 method_name='placebo_treatment_refuter',
                                 placebo_type='permute',
                                 num_simulations=20)
print(refutation)
```

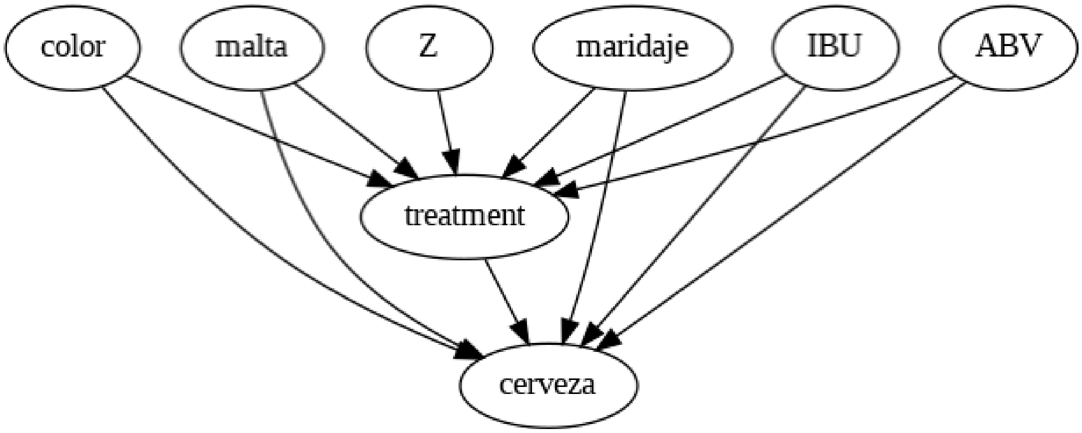
```
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 8 variables. 7 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'cuerpo', 'malta', 'maridaje', 'treatment']'.
The unobserved variables are: '['Z']'. If this matches your expectations for
observations, please continue. If you expected any of the unobserved variables
to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 2 variables in the dataset
that are not in the graph. Variable names are: '['color', 'propensity_score']'
```

```
TEST: color
```



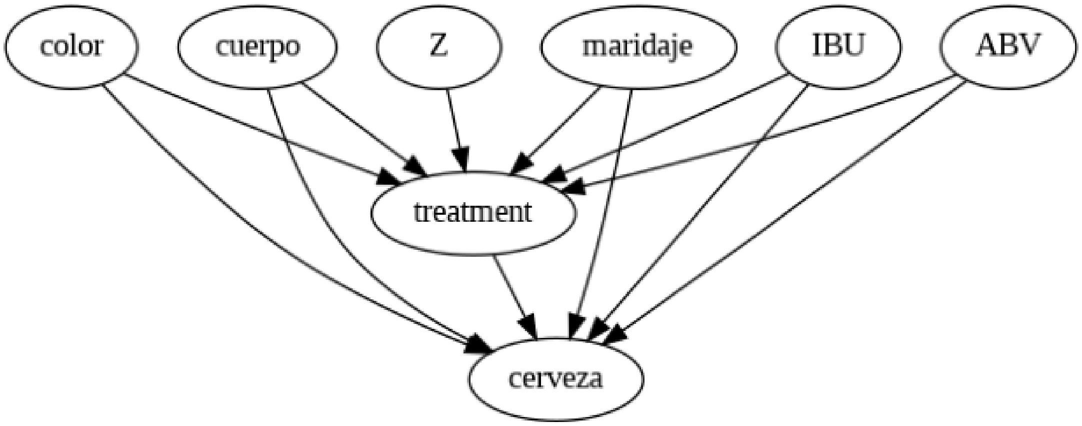
```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 8 variables. 7 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'malta', 'maridaje', 'treatment']'. The
unobserved variables are: '['Z']'. If this matches your expectations for
observations, please continue. If you expected any of the unobserved variables
to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 2 variables in the dataset
that are not in the graph. Variable names are: '['cuerpo', 'propensity_score']'
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.023555040811824395
p value:0.40148520270478993

TEST: cuerpo
```

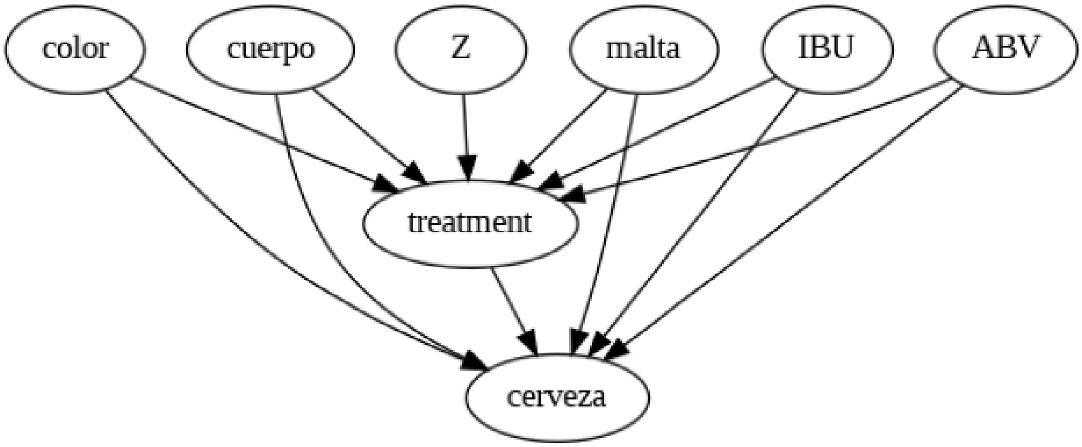


```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 8 variables. 7 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'cuerpo', 'maridaje', 'treatment']'.
The unobserved variables are: '['Z']'. If this matches your expectations for
observations, please continue. If you expected any of the unobserved variables
to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 2 variables in the dataset
that are not in the graph. Variable names are: '['malta', 'propensity_score']'
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.03112728877123318
p value:0.2367931523297505

TEST: malta
```

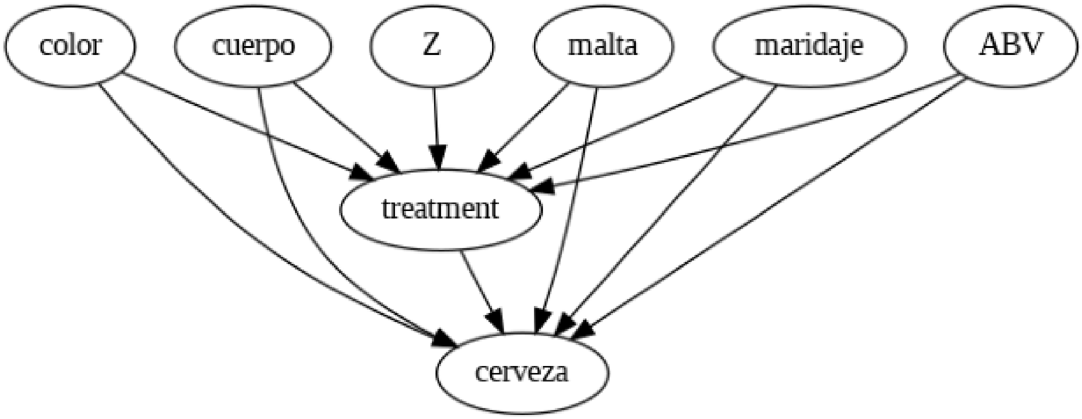


```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 8 variables. 7 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'IBU', 'cerveza', 'color', 'cuerpo', 'malta', 'treatment']'. The
unobserved variables are: '['Z']'. If this matches your expectations for
observations, please continue. If you expected any of the unobserved variables
to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 2 variables in the dataset
that are not in the graph. Variable names are: '['maridaje',
'propensity_score']'

Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:-0.008338848444738586
p value:0.4632596662135783

TEST: maridaje
```



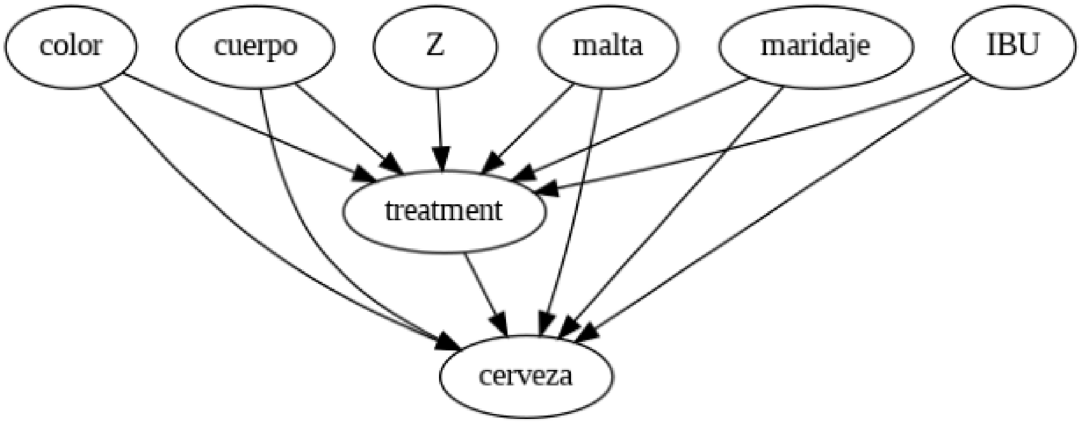
```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 8 variables. 7 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['ABV', 'cerveza', 'color', 'cuerpo', 'malta', 'maridaje', 'treatment']'.
The unobserved variables are: '['Z']'. If this matches your expectations for
observations, please continue. If you expected any of the unobserved variables
to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 2 variables in the dataset
that are not in the graph. Variable names are: '['IBU', 'propensity_score']'
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.02531987646150452
p value:0.3793019028275614

TEST: IBU
```



```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 8 variables. 7 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['IBU', 'cerveza', 'color', 'cuerpo', 'malta', 'maridaje', 'treatment']'.
The unobserved variables are: '['Z']'. If this matches your expectations for
observations, please continue. If you expected any of the unobserved variables
to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 2 variables in the dataset
that are not in the graph. Variable names are: '['ABV', 'propensity_score']'
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.015111405250386059
p value:0.4037711933173521

TEST: ABV
```



```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
```

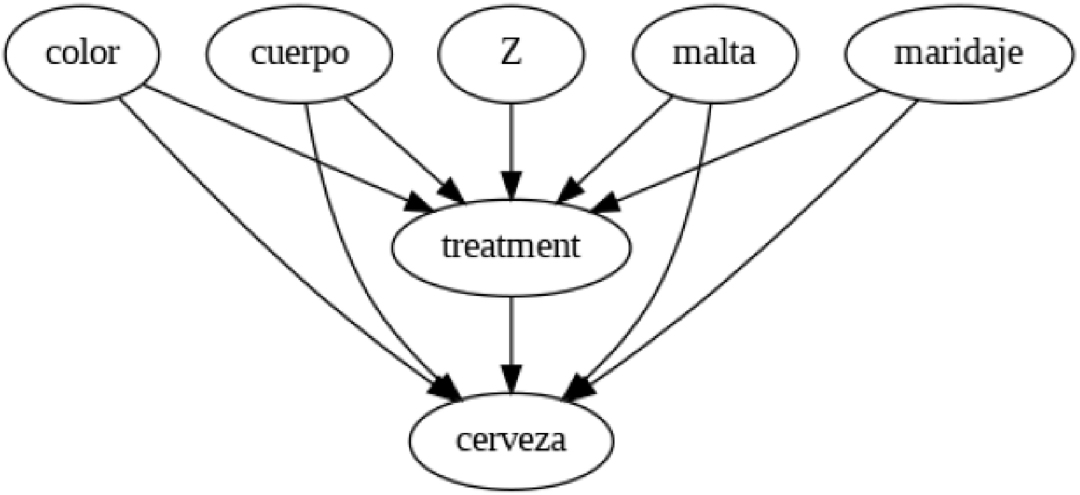
```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.019165012133245094
p value:0.39433735106217666
```

En esta segunda serie de análisis, donde se excluyó individualmente cada atributo (color, cuerpo, malta, maridaje, IBU, ABV) del modelo, los resultados sugieren nuevamente que no hay un efecto significativo de estas variables en la coincidencia de las recomendaciones de cerveza con las preferencias del usuario. Esto se evidencia por los p values obtenidos, que permanecen relativamente altos para todas las variables, excepto para IBU, donde el p value es bastante más bajo (0.2155). Sin embargo, incluso este valor no es suficientemente bajo como para considerarlo estadísticamente significativo en muchos contextos científicos. Estos resultados sugieren que la influencia de estas variables individuales en las recomendaciones de cerveza, si existe, es probablemente pequeña o está siendo eclipsada por otros factores no capturados en el modelo.

```
[ ]: # Ultima tesis... y si nos confundimos, y en realidad la influencia era para el␣
     ↪otro lado?
     # o sea, en vez de sacar cuerpo y malta, sacar ABV e IBU

     import dowhy

     causal_graph = """digraph {
                             Z->treatment;
                             color->treatment;
                             cuerpo->treatment;
                             malta->treatment;
```

```
                           maridaje->treatment;
                           color->cerveza;
                           cuerpo->cerveza;
                           malta->cerveza;
                           maridaje->cerveza;
                           treatment->cerveza;
               }"""
model = dowhy.CausalModel(data=df, treatment="treatment", outcome="cerveza",␣
 ↪graph=causal_graph)
model.view_model()
identified_estimand = model.identify_effect(proceed_when_unidentifiable=True)
estimate = model.estimate_effect(identified_estimand,
                                 method_name='backdoor.
 ↪propensity_score_matching',
                                 target_units='att')
refutation = model.refute_estimate(identified_estimand,
                                   estimate,
                                   method_name='placebo_treatment_refuter',
                                   placebo_type='permute',
                                   num_simulations=20)
print(refutation)
```

```
/usr/local/lib/python3.10/dist-packages/dowhy/causal_model.py:557: UserWarning:
1 variables are assumed unobserved because they are not in the dataset.
Configure the logging level to `logging.WARNING` or higher for additional
details.
  warnings.warn(
WARNING:dowhy.causal_model:The graph defines 7 variables. 6 were found in the
dataset and will be analyzed as observed variables. 1 were not found in the
dataset and will be analyzed as unobserved variables. The observed variables
are: '['cerveza', 'color', 'cuerpo', 'malta', 'maridaje', 'treatment']'. The
unobserved variables are: '['Z']'. If this matches your expectations for
observations, please continue. If you expected any of the unobserved variables
to be in the dataframe, please check for typos.
WARNING:dowhy.causal_model:There are an additional 3 variables in the dataset
that are not in the graph. Variable names are: '['ABV', 'IBU',
'propensity_score']'
```



```
WARNING:dowhy.causal_refuter:We assume a Normal Distribution as the sample has
less than 100 examples.
             Note: The underlying distribution may not be Normal. We assume that
it approaches normal with the increase in sample size.
```

```
Refute: Use a Placebo Treatment
Estimated effect:-1.342598720494154
New effect:0.007820427972645047
p value:0.45292113787486243
```

1 Conclusiones

Basándonos en los resultados obtenidos de los análisis realizados, se puede deducir lo siguiente:

Hipótesis inicial: La hipótesis inicial era que características específicas de la cerveza, como el color, cuerpo, malta, maridaje, IBU y ABV, influirían significativamente en la coincidencia entre las recomendaciones del motor y las preferencias del usuario.

Resultados de la refutación: Los análisis mostraron que ninguna de estas características, consideradas individualmente o excluyendo una a la vez, tiene un efecto estadísticamente significativo en las recomendaciones de cerveza. Incluso el IBU, que mostró el p value más bajo, no alcanzó un nivel de significancia convencional.

Conclusión: Los resultados sugieren que las preferencias de los usuarios y las recomendaciones del motor no están fuertemente influenciadas por ninguna de estas características individuales de la cerveza. Esto podría indicar que las preferencias de los usuarios están seguramente influenciadas por una combinación compleja de factores, o por aspectos no capturados en el modelo actual. También es posible que las características intrínsecas de las cervezas no sean tan decisivas para las preferencias de los usuarios como se esperaba inicialmente.

2 IA Explainable (XAI)

[ ]:
```
from pycaret.classification import setup, compare_models, create_model,␣
 ↪interpret_model, plot_model
```

[ ]:
```
clf1 = setup(data=df, target='cerveza', session_id=123)
```

```
<pandas.io.formats.style.Styler at 0x7c57a094e3b0>
```

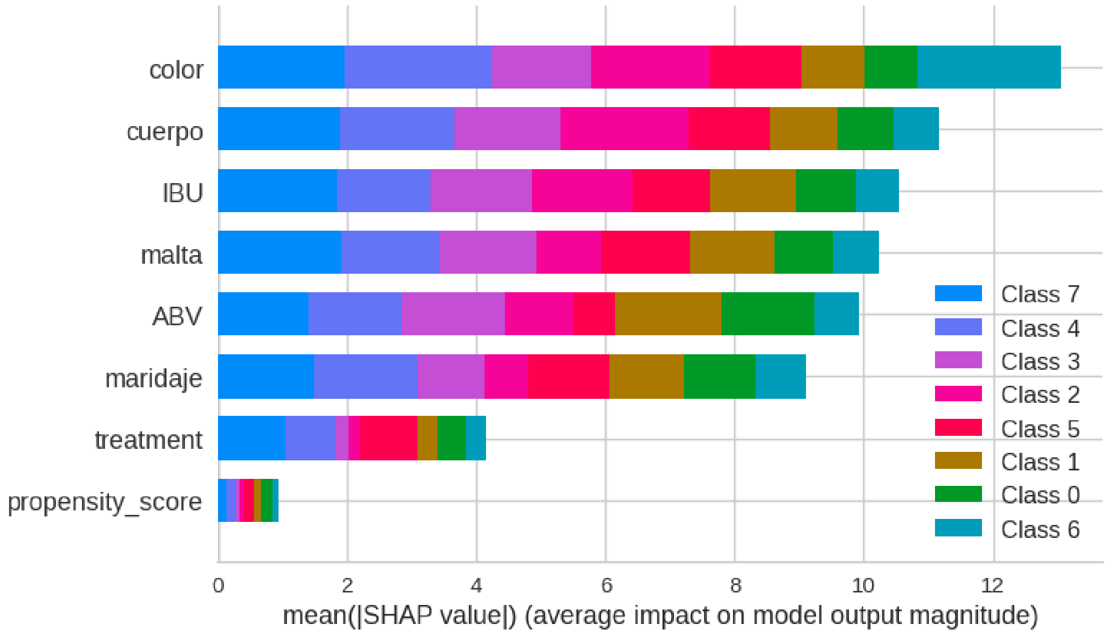
[ ]:
```
best_model = compare_models()
```

```
<IPython.core.display.HTML object>

<pandas.io.formats.style.Styler at 0x7c582c336620>

Processing:   0%|          | 0/69 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

[ ]:
```
# Creando un modelo específico con CatBoost
catboost = create_model('catboost', cross_validation=False)
```

```
<IPython.core.display.HTML object>

<pandas.io.formats.style.Styler at 0x7c57d8318430>

Processing:   0%|          | 0/4 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

[ ]:
```
interpret_model(catboost)
```



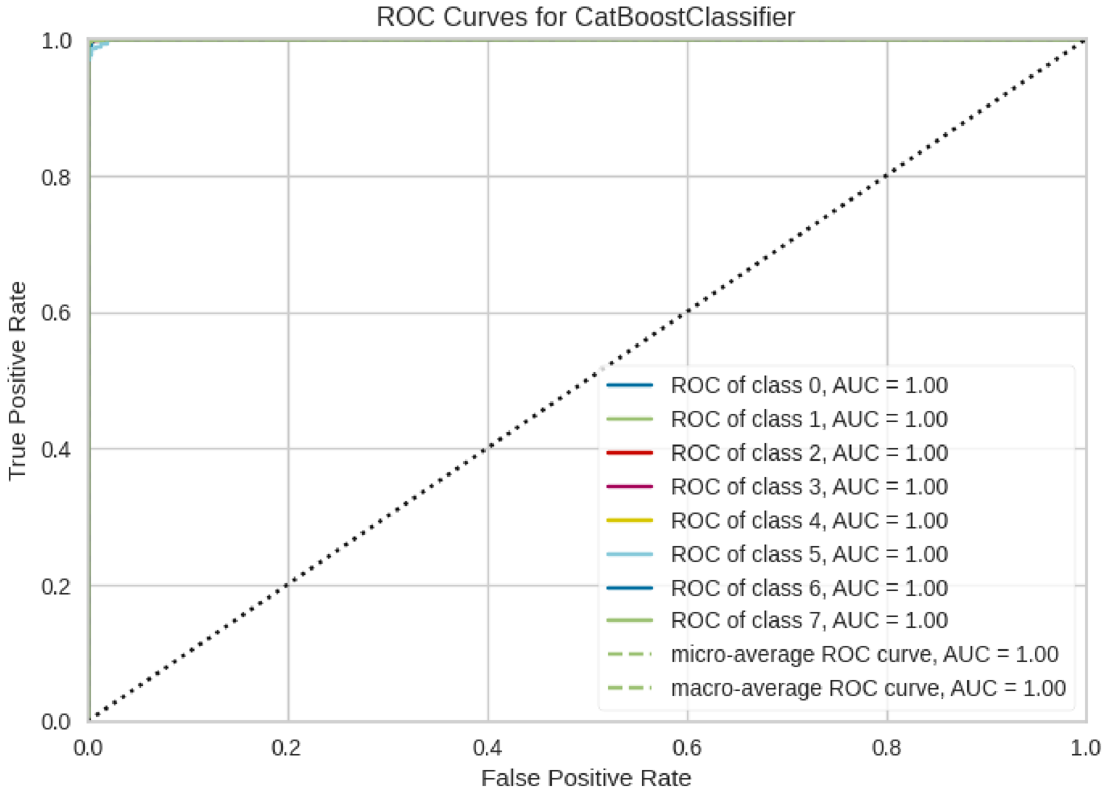
En este gráfico, se observa que las características color, cuerpo, IBU, malta, ABV, y maridaje tienen un impacto significativo en las predicciones del modelo. La característica con el mayor impacto promedio en el modelo es color, seguido por cuerpo y IBU. Esto sugiere que estas son las características más importantes según el modelo CatBoost para predecir la variable objetivo, que es la cerveza.

El propensity_score parece tener un impacto muy bajo comparado con las otras características, lo cual es esperado ya que es un score de inclinación utilizado para el balanceo en el contexto de la estimación causal y no un predictor directo.

[ ]:
```
# Falla con el error typeError: The passed shap_values are a list not an array! If you have a list of explanations try passing shap_values[0] instead to explain the first output class of a multi-output model.

# interpret_model(catboost, plot = 'correlation')
```

[ ]:
```
plot_model(catboost)
```

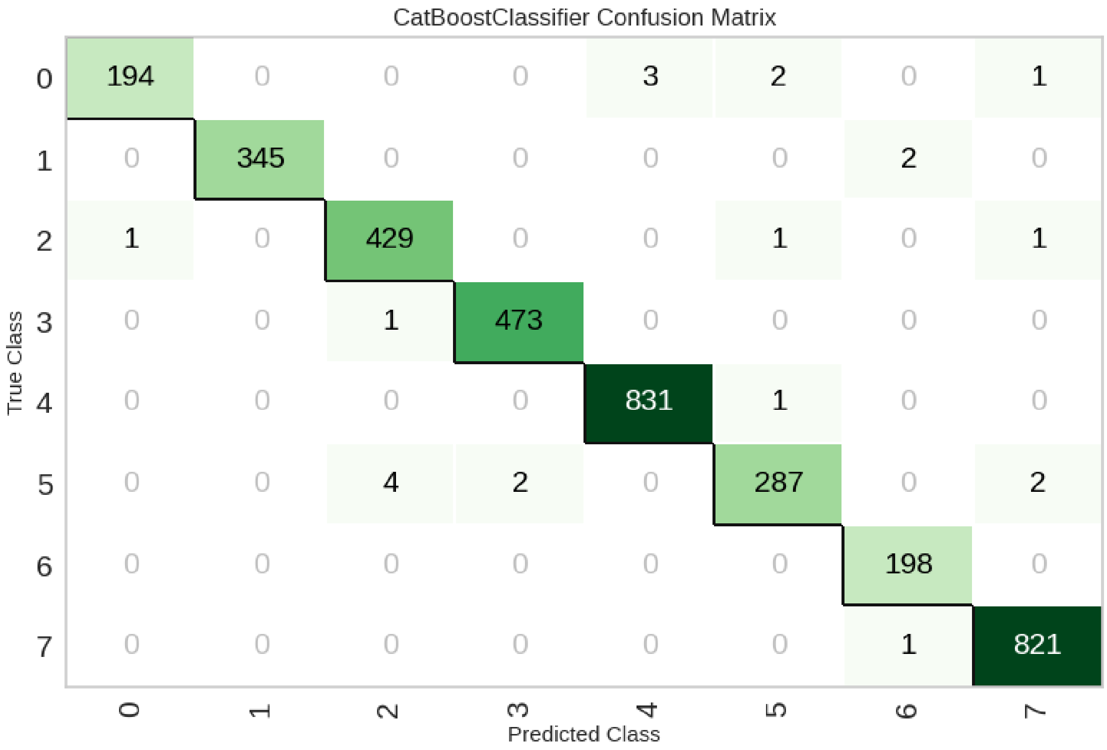
```
<IPython.core.display.HTML object>
```



El hecho de que cada clase tenga un AUC de 1.0 sugiere que el modelo tiene un rendimiento excepcionalmente alto, pudiendo distinguir perfectamente entre clases. Sin embargo, esto es inusual en la práctica y nos indica que hay overfitting

```
[ ]: plot_model(catboost, plot='confusion_matrix')
```

```
<IPython.core.display.HTML object>
```

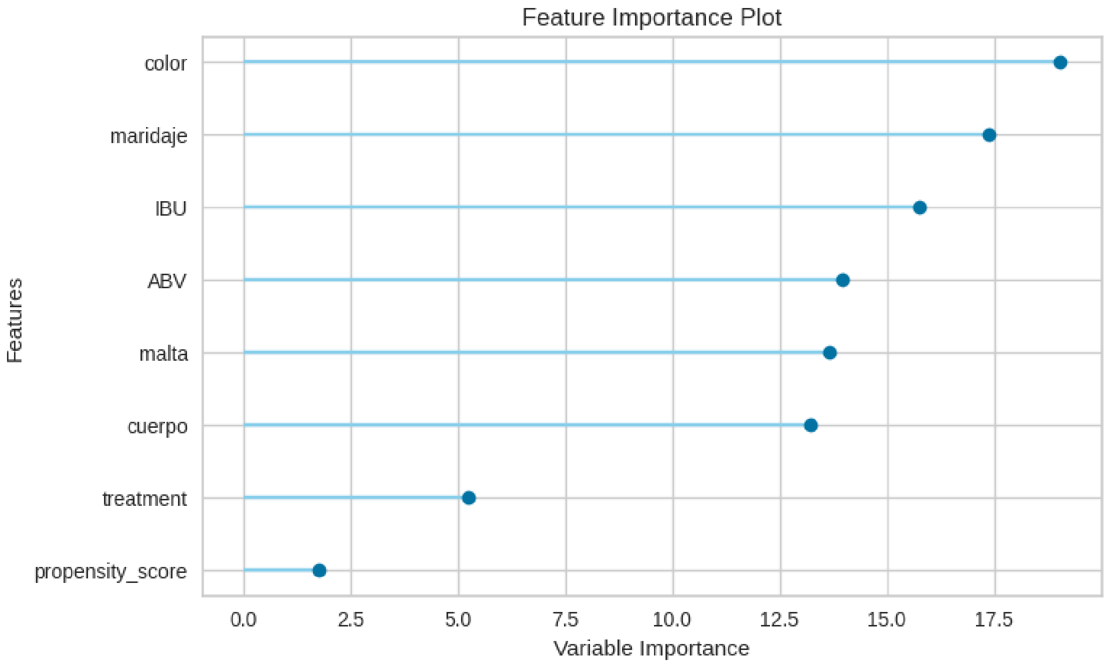


	0	1	2	3	4	5	6	7
0	194	0	0	0	3	2	0	1
1	0	345	0	0	0	0	2	0
2	1	0	429	0	0	1	0	1
3	0	0	1	473	0	0	0	0
4	0	0	0	0	831	1	0	0
5	0	0	4	2	0	287	0	2
6	0	0	0	0	0	0	198	0
7	0	0	0	0	0	0	1	821

Nuevamente, debido al overfitting, el modelo parece tener un buen rendimiento, con la mayoría de las predicciones concentradas en la diagonal principal, lo que indica una alta tasa de aciertos.

```
[ ]: plot_model(catboost, plot='feature')
```

```
<IPython.core.display.HTML object>
```



La característica ‘color’ tiene la mayor importancia, seguida por ‘maridaje’ e ‘IBU’, lo que sugiere que estas son las variables más influyentes en la predicción del modelo. ‘Propensity_score’ tiene la menor importancia, lo que concuerda con el análisis anterior de que tiene un bajo impacto en las predicciones del modelo.

2.1 Apéndice de pruebas no exitosas

[ ]:
```
import pandas as pd

data = [
    {"color": "Negra", "cuerpo": "Completo", "malta": "Negra", "IBU": 4, "ABV": 3, "maridaje": "Carnes rojas", "cerveza": "Stout"},
    {"color": "Negra", "cuerpo": "Completo", "malta": "Negra", "IBU": 4, "ABV": 3, "maridaje": "Sola", "cerveza": "Stout"},
    {"color": "Clara", "cuerpo": "Ligero", "malta": "Pálida", "IBU": 1, "ABV": 2, "maridaje": "Carnes blancas", "cerveza": "Lager"},
    {"color": "Clara", "cuerpo": "Ligero", "malta": "Pálida", "IBU": 1, "ABV": 2, "maridaje": "Salado", "cerveza": "Lager"},
    {"color": "Clara", "cuerpo": "Ligero", "malta": "Pálida", "IBU": 1, "ABV": 2, "maridaje": "Sola", "cerveza": "Lager"},
    {"color": "Roja", "cuerpo": "Medio", "malta": "Caramelo", "IBU": 5, "ABV": 5, "maridaje": "Carnes rojas", "cerveza": "IPA"},
    {"color": "Roja", "cuerpo": "Medio", "malta": "Caramelo", "IBU": 5, "ABV": 5, "maridaje": "Sola", "cerveza": "IPA"},
    {"color": "Rubia", "cuerpo": "Cremoso", "malta": "Caramelo", "IBU": 2,␣
 ↪"ABV": 5, "maridaje": "Carnes rojas", "cerveza": "Honey"},
    {"color": "Rubia", "cuerpo": "Cremoso", "malta": "Caramelo", "IBU": 2,␣
 ↪"ABV": 5, "maridaje": "Sola", "cerveza": "Honey"},
    {"color": "Roja", "cuerpo": "Medio", "malta": "Tostada", "IBU": 3, "ABV":␣
 ↪1, "maridaje": "Salado", "cerveza": "Ale sin alcohol"},
    {"color": "Roja", "cuerpo": "Medio", "malta": "Tostada", "IBU": 3, "ABV":␣
 ↪1, "maridaje": "Sola", "cerveza": "Ale sin alcohol"},
    {"color": "Rubia", "cuerpo": "Ligero", "malta": "Pálida", "IBU": 1, "ABV":␣
 ↪2, "maridaje": "Carnes blancas", "cerveza": "Rubia"},
    {"color": "Rubia", "cuerpo": "Ligero", "malta": "Pálida", "IBU": 1, "ABV":␣
 ↪2, "maridaje": "Sola", "cerveza": "Rubia"},
    {"color": "Roja", "cuerpo": "Medio", "malta": "Tostada", "IBU": 3, "ABV":␣
 ↪3, "maridaje": "Quesos", "cerveza": "Ale Roja Irlandesa"},
    {"color": "Roja", "cuerpo": "Medio", "malta": "Tostada", "IBU": 3, "ABV":␣
 ↪3, "maridaje": "Sola", "cerveza": "Ale Roja Irlandesa"},
    {"color": "Negra", "cuerpo": "Completo", "malta": "Chocolate", "IBU": 5,␣
 ↪"ABV": 3, "maridaje": "Carnes blancas", "cerveza": "Porter"},
    {"color": "Negra", "cuerpo": "Completo", "malta": "Chocolate", "IBU": 5,␣
 ↪"ABV": 3, "maridaje": "Quesos", "cerveza": "Porter"},
    {"color": "Negra", "cuerpo": "Completo", "malta": "Chocolate", "IBU": 5,␣
 ↪"ABV": 3, "maridaje": "Sola", "cerveza": "Porter"},


    {"color": "Clara", "cuerpo": "Completo", "malta": "Negra", "IBU": 4, "ABV":␣
 ↪3, "maridaje": "Carnes rojas", "cerveza": "Stout"},
    {"color": "Roja", "cuerpo": "Medio", "malta": "Caramelo", "IBU": 5, "ABV":␣
 ↪5, "maridaje": "Carnes blancas", "cerveza": "IPA"},
    {"color": "Rubia", "cuerpo": "Medio", "malta": "Caramelo", "IBU": 2, "ABV":␣
 ↪5, "maridaje": "Carnes rojas", "cerveza": "Honey"},
    {"color": "Roja", "cuerpo": "Medio", "malta": "Tostada", "IBU": 3, "ABV":␣
 ↪2, "maridaje": "Sola", "cerveza": "Ale sin alcohol"},
    {"color": "Rubia", "cuerpo": "Ligero", "malta": "Tostada", "IBU": 1, "ABV":␣
 ↪2, "maridaje": "Sola", "cerveza": "Rubia"},
    {"color": "Roja", "cuerpo": "Medio", "malta": "Tostada", "IBU": 4, "ABV":␣
 ↪3, "maridaje": "Quesos", "cerveza": "Ale Roja Irlandesa"},
]

df2 = pd.DataFrame(data)

df2
```

```
[ ]:    color    cuerpo     malta  IBU  ABV        maridaje            cerveza
     0  Negra  Completo     Negra    4    3    Carnes rojas              Stout
     1  Negra  Completo     Negra    4    3            Sola              Stout
     2  Clara    Ligero    Pálida    1    2  Carnes blancas              Lager
```

```
3   Clara    Ligero     Pálida   1   2         Salado                Lager
4   Clara    Ligero     Pálida   1   2           Sola                Lager
5    Roja     Medio   Caramelo   5   5   Carnes rojas                  IPA
6    Roja     Medio   Caramelo   5   5           Sola                  IPA
7   Rubia   Cremoso   Caramelo   2   5   Carnes rojas                Honey
8   Rubia   Cremoso   Caramelo   2   5           Sola                Honey
9    Roja     Medio    Tostada   3   1         Salado      Ale sin alcohol
10   Roja     Medio    Tostada   3   1           Sola      Ale sin alcohol
11  Rubia    Ligero     Pálida   1   2 Carnes blancas                Rubia
12  Rubia    Ligero     Pálida   1   2           Sola                Rubia
13   Roja     Medio    Tostada   3   3         Quesos  Ale Roja Irlandesa
14   Roja     Medio    Tostada   3   3           Sola  Ale Roja Irlandesa
15  Negra  Completo  Chocolate   5   3 Carnes blancas               Porter
16  Negra  Completo  Chocolate   5   3         Quesos               Porter
17  Negra  Completo  Chocolate   5   3           Sola               Porter
18  Clara  Completo      Negra   4   3   Carnes rojas                Stout
19   Roja     Medio   Caramelo   5   5 Carnes blancas                  IPA
20  Rubia     Medio   Caramelo   2   5   Carnes rojas                Honey
21   Roja     Medio    Tostada   3   2           Sola      Ale sin alcohol
22  Rubia    Ligero    Tostada   1   2           Sola                Rubia
23   Roja     Medio    Tostada   4   3         Quesos  Ale Roja Irlandesa
```

[ ]:
```python
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()

for column in df2.columns:
    df2[column] = label_encoder.fit_transform(df2[column])

# Ahora todas las columnas categóricas están convertidas a numéricas
print(df2.head())
```

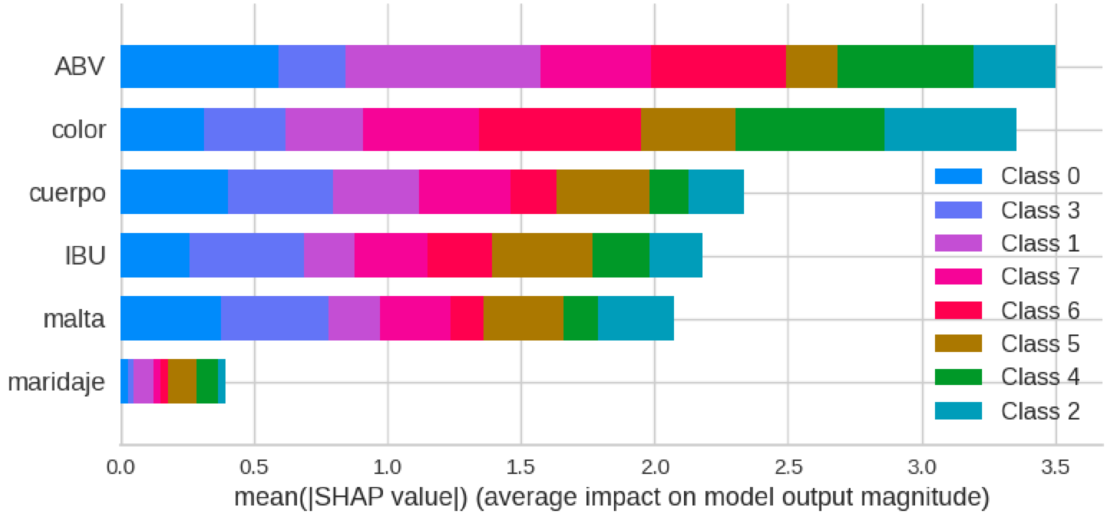
```
   color  cuerpo  malta  IBU  ABV  maridaje  cerveza
0      1       0      2    3    2         1        7
1      1       0      2    3    2         4        7
2      0       2      3    0    1         0        4
3      0       2      3    0    1         3        4
4      0       2      3    0    1         4        4
```

[ ]:
```python
from pycaret.classification import setup, compare_models, create_model,␣
  ↪interpret_model, plot_model
clf2 = setup(data=df2, target='cerveza', session_id=124)
```

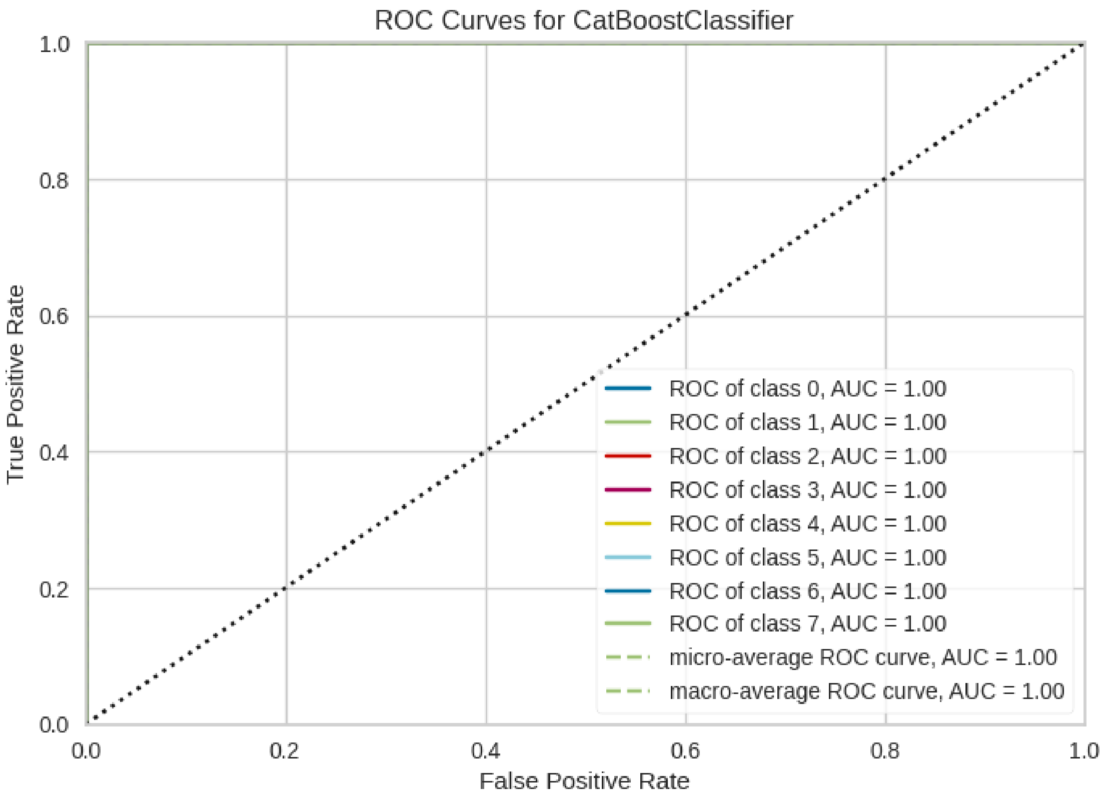
```
<pandas.io.formats.style.Styler at 0x792ce9e57a90>
```

[ ]:
```python
catboost2 = create_model('catboost', cross_validation=False)
interpret_model(catboost2)
plot_model(catboost2)
plot_model(catboost2, plot='confusion_matrix')
plot_model(catboost2, plot='feature')
```

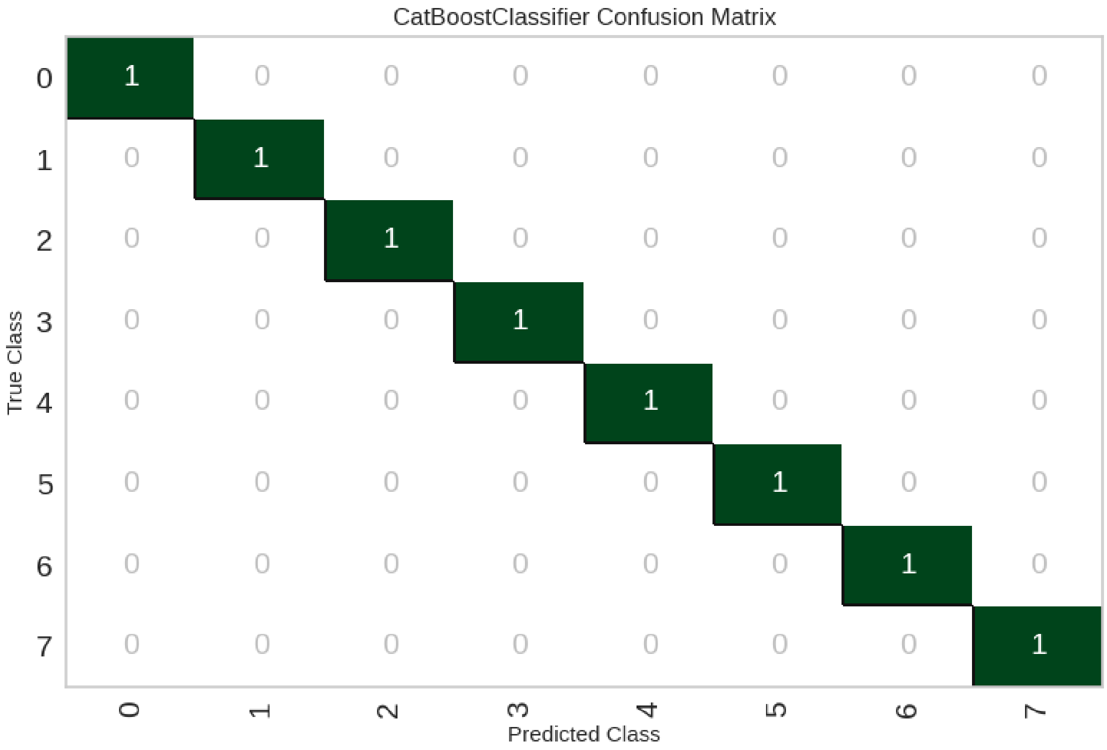
```
<IPython.core.display.HTML object>

<pandas.io.formats.style.Styler at 0x792bf5c06c50>

Processing:   0%|          | 0/4 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```



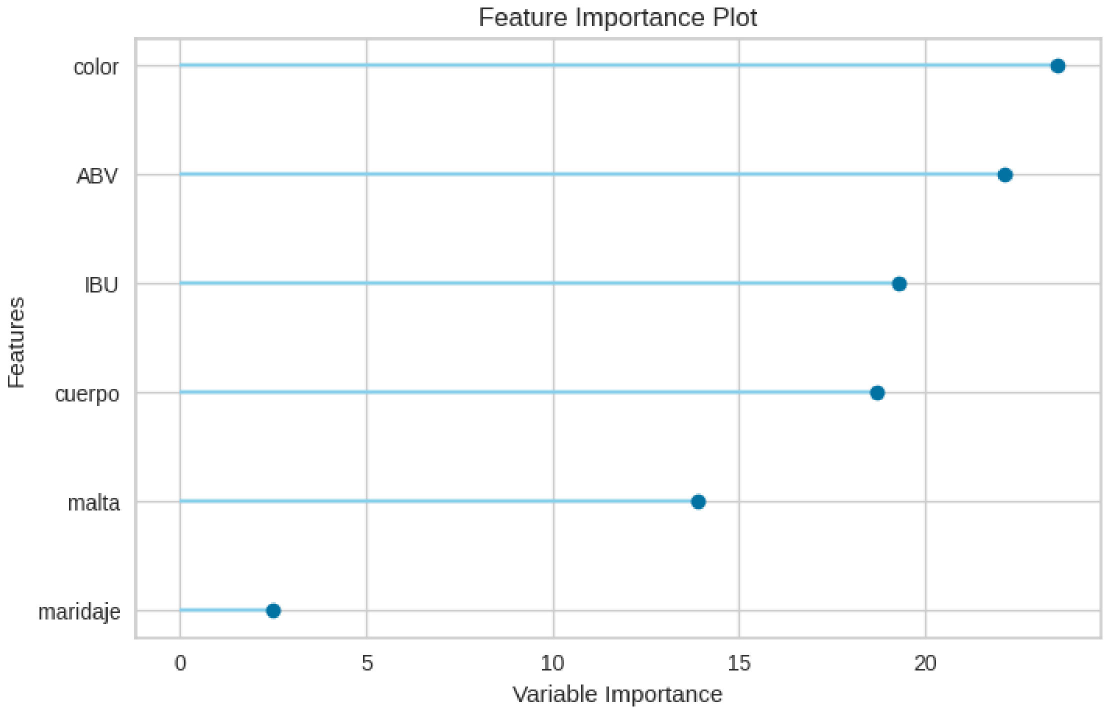
```
<IPython.core.display.HTML object>
```


ROC Curves for CatBoostClassifier
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC of class 0, AUC = 1.00
ROC of class 1, AUC = 1.00
ROC of class 2, AUC = 1.00
ROC of class 3, AUC = 1.00
ROC of class 4, AUC = 1.00
ROC of class 5, AUC = 1.00
ROC of class 6, AUC = 1.00
ROC of class 7, AUC = 1.00
micro-average ROC curve, AUC = 1.00
macro-average ROC curve, AUC = 1.00

```
<IPython.core.display.HTML object>
```


CatBoostClassifier Confusion Matrix
True Class
0
1
2
3
4
5
6
7
1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1
0
1
2
3
4
5
6
7
Predicted Class

```
<IPython.core.display.HTML object>
```


Feature Importance Plot
Features
color
ABV
IBU
cuerpo
malta
maridaje
0
5
10
15
20
Variable Importance