assurances Code Python Case

November 13, 2024

1 Case study: “Underwriter for a day”

1.0.1 Prepared by Thomas De Coninck and Mariève Gendron

1.0.2 Presenter to Franca Glenzer

1.0.3 For the course FINA 60220A

1.0.4 November 14, 2024

Disclaimer: generative AI was used in the making of this project.

[607]:

```
# pip install if necessary

!pip install pandas numpy matplotlib seaborn scipy
!pip install statsmodels
!pip install scikit-learn
!pip install kmodes catboost
```

```
Requirement already satisfied: pandas in /opt/anaconda3/lib/python3.12/site-
packages (2.2.2)
Requirement already satisfied: numpy in /opt/anaconda3/lib/python3.12/site-
packages (1.26.4)
Requirement already satisfied: matplotlib in /opt/anaconda3/lib/python3.12/site-
packages (3.8.4)
Requirement already satisfied: seaborn in /opt/anaconda3/lib/python3.12/site-
packages (0.13.2)
Requirement already satisfied: scipy in /opt/anaconda3/lib/python3.12/site-
packages (1.13.1)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
```

```
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=8 in /opt/anaconda3/lib/python3.12/site-
packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.12/site-
packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: statsmodels in
/opt/anaconda3/lib/python3.12/site-packages (0.14.2)
Requirement already satisfied: numpy>=1.22.3 in
/opt/anaconda3/lib/python3.12/site-packages (from statsmodels) (1.26.4)
Requirement already satisfied: scipy!=1.9.2,>=1.8 in
/opt/anaconda3/lib/python3.12/site-packages (from statsmodels) (1.13.1)
Requirement already satisfied: pandas!=2.1.0,>=1.4 in
/opt/anaconda3/lib/python3.12/site-packages (from statsmodels) (2.2.2)
Requirement already satisfied: patsy>=0.5.6 in
/opt/anaconda3/lib/python3.12/site-packages (from statsmodels) (0.5.6)
Requirement already satisfied: packaging>=21.3 in
/opt/anaconda3/lib/python3.12/site-packages (from statsmodels) (23.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/anaconda3/lib/python3.12/site-packages (from
pandas!=2.1.0,>=1.4->statsmodels) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/opt/anaconda3/lib/python3.12/site-packages (from
pandas!=2.1.0,>=1.4->statsmodels) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
/opt/anaconda3/lib/python3.12/site-packages (from
pandas!=2.1.0,>=1.4->statsmodels) (2023.3)
Requirement already satisfied: six in /opt/anaconda3/lib/python3.12/site-
packages (from patsy>=0.5.6->statsmodels) (1.16.0)
Requirement already satisfied: scikit-learn in
/opt/anaconda3/lib/python3.12/site-packages (1.4.2)
Requirement already satisfied: numpy>=1.19.5 in
/opt/anaconda3/lib/python3.12/site-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in
/opt/anaconda3/lib/python3.12/site-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in
/opt/anaconda3/lib/python3.12/site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/anaconda3/lib/python3.12/site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: kmodes in /opt/anaconda3/lib/python3.12/site-
packages (0.12.2)
Requirement already satisfied: catboost in /opt/anaconda3/lib/python3.12/site-
packages (1.2.7)
```

```
Requirement already satisfied: numpy>=1.10.4 in
/opt/anaconda3/lib/python3.12/site-packages (from kmodes) (1.26.4)
Requirement already satisfied: scikit-learn>=0.22.0 in
/opt/anaconda3/lib/python3.12/site-packages (from kmodes) (1.4.2)
Requirement already satisfied: scipy>=0.13.3 in
/opt/anaconda3/lib/python3.12/site-packages (from kmodes) (1.13.1)
Requirement already satisfied: joblib>=0.11 in
/opt/anaconda3/lib/python3.12/site-packages (from kmodes) (1.4.2)
Requirement already satisfied: graphviz in /opt/anaconda3/lib/python3.12/site-
packages (from catboost) (0.20.3)
Requirement already satisfied: matplotlib in /opt/anaconda3/lib/python3.12/site-
packages (from catboost) (3.8.4)
Requirement already satisfied: pandas>=0.24 in
/opt/anaconda3/lib/python3.12/site-packages (from catboost) (2.2.2)
Requirement already satisfied: plotly in /opt/anaconda3/lib/python3.12/site-
packages (from catboost) (5.22.0)
Requirement already satisfied: six in /opt/anaconda3/lib/python3.12/site-
packages (from catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas>=0.24->catboost)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas>=0.24->catboost)
(2024.1)
Requirement already satisfied: tzdata>=2022.7 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas>=0.24->catboost)
(2023.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/anaconda3/lib/python3.12/site-packages (from scikit-learn>=0.22.0->kmodes)
(2.2.0)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib->catboost) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib->catboost) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib->catboost) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib->catboost) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib->catboost) (23.2)
Requirement already satisfied: pillow>=8 in /opt/anaconda3/lib/python3.12/site-
packages (from matplotlib->catboost) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/anaconda3/lib/python3.12/site-packages (from matplotlib->catboost) (3.0.9)
Requirement already satisfied: tenacity>=6.2.0 in
/opt/anaconda3/lib/python3.12/site-packages (from plotly->catboost) (8.2.2)
```

[608]:

```python
# libraries

# Essential libraries for data manipulation, visualization, and statistics
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.ticker import FuncFormatter, MaxNLocator
from scipy.stats import poisson, binom, nbinom, lognorm, pareto, gamma,␣
 ↪scoreatpercentile

# Statsmodels for statistical modeling
import statsmodels.api as sm
import statsmodels.formula.api as smf

# Scikit-learn for preprocessing, model selection, and evaluation
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import classification_report, silhouette_score,␣
 ↪davies_bouldin_score

# Clustering and classification models
from kmodes.kmodes import KModes
from catboost import CatBoostClassifier
```

[609]:

```python
# Load the claims data file into a DataFrame
file_path = "claim_data_group4_2024.csv"
claims_data_df = pd.read_csv(file_path)
claims_data_df
```

[609]:

```
         IDpol  ClaimNb  Exposure Area  VehPower  VehAge  DrivAge  BonusMalus  \
0      2271893        0      0.83    E         5      17       53          64
1      1111864        0      0.24    E         5       2       27          64
2        72908        0      0.50    E         7      11       67          50
3      2283027        0      0.08    B         5       8       28          60
4      1123838        0      0.03    A        11       1       38          50
…          …       …        …   …       …       …      …           …
99995    70445        0      1.00    C         5      11       37          56
99996  4163362        0      0.22    E         6      13       58          50
99997  2081912        0      1.00    E         5       1       49          50
99998  2012998        0      0.71    D         9       9       36          54
99999  3087666        0      0.53    C         9      14       35          51

      VehBrand   VehGas  Density Region  ClaimAmount
0           B2   Diesel     3317    R93          0.0
```

```
1           B3   Diesel     2740     R22          0.0
2           B3  Regular     4762     R93          0.0
3           B1   Diesel       64     R91          0.0
4           B2  Regular       16     R24          0.0
…        …      …      …      …         …
99995       B2   Diesel      317     R82          0.0
99996       B1   Diesel     4762     R93          0.0
99997       B2   Diesel     4998     R11          0.0
99998       B1  Regular     1541     R91          0.0
99999       B3  Regular      161     R31          0.0

[100000 rows x 13 columns]
```

[610]:

```
# Claim frequency will be used instead for number of claims as the period of␣
 ↪exposure, meaning, the time a claim can occur, is also considered.
# Claim severity is the average claim amount per claim and will be used instead␣
 ↪of the claim amount.
claims_data_df['Frequency'] = claims_data_df['ClaimNb'] /␣
 ↪claims_data_df['Exposure'] #Number of claims per year
claims_data_df['Severity'] = claims_data_df['ClaimAmount'] /␣
 ↪claims_data_df['ClaimNb'] #Amount per claim

# Drop the original columns and place the new columns at the same position as␣
 ↪the original columns
claims_data = claims_data_df.drop(columns=['ClaimNb'])

# Fill missing values in the 'Severity' column with 0 (meaning no claim␣
 ↪occurred)
claims_data['Severity'] = claims_data['Severity'].fillna(0)
```

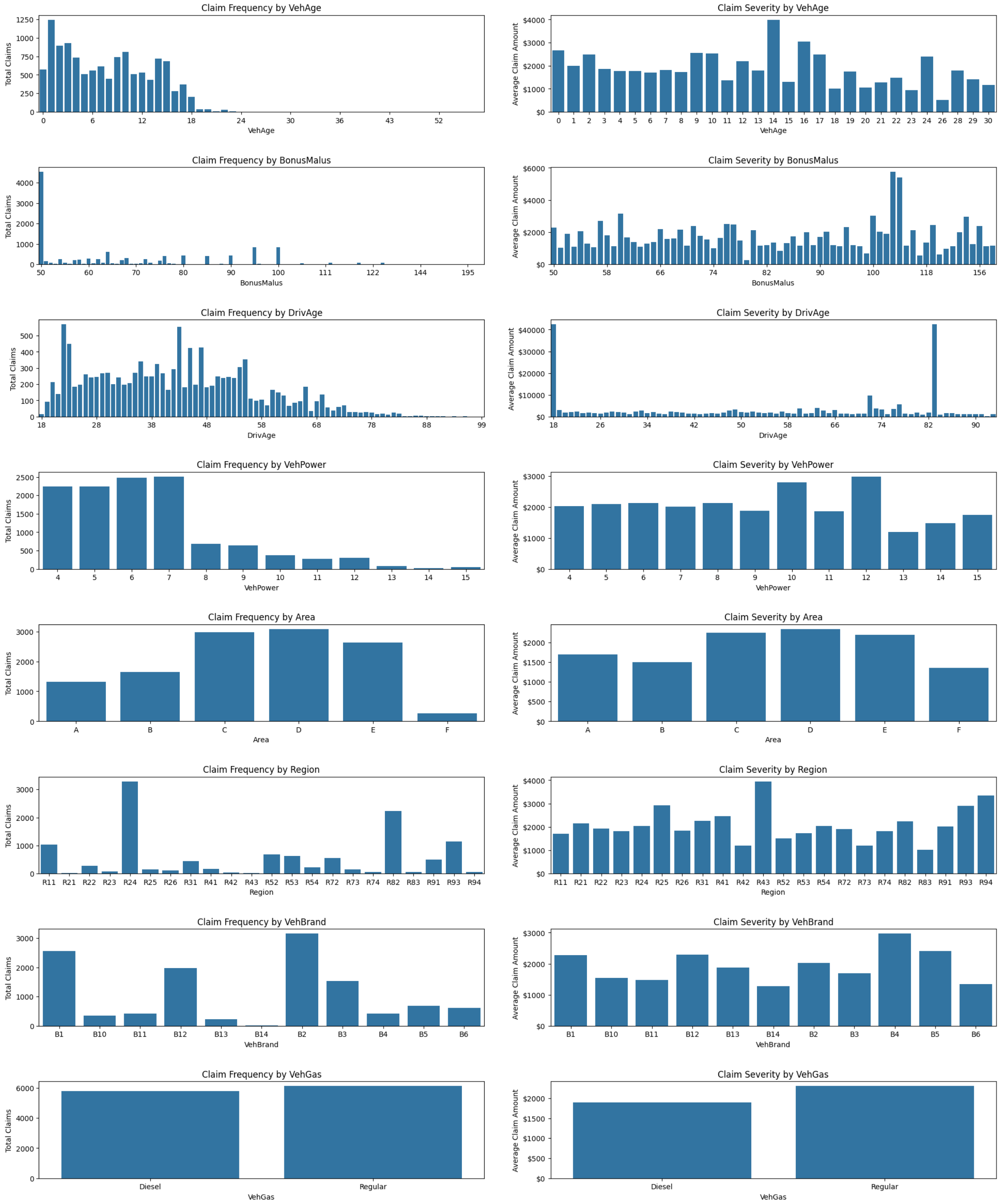
2 1. Descriptive Analysis of Risk Variables

[611]:

```
def plot_variable(data, group_var, ax_freq, ax_sev):
    """
    Function to plot frequency and severity of claims by policyholder␣
 ↪characteristics.

    :param data: DataFrame to plot.
    :param group_var: The variable to group by.
    :param ax_freq: The axes object for frequency plots.
    :param ax_sev: The axes object for severity plots.
    """
    # Calculate frequency and severity
    freq = data.groupby(group_var)['Frequency'].sum()
    sev = data[data['Frequency'] > 0].groupby(group_var)['ClaimAmount'].mean()
    dollar_formatter = FuncFormatter(lambda x, pos: f'${int(x)}')

    # Plot Frequency
    sns.barplot(x=freq.index, y=freq.values, ax=ax_freq)
    ax_freq.set_title(f'Claim Frequency by {group_var}')
    ax_freq.set_xlabel(group_var)
    ax_freq.set_ylabel('Total Claims')
    ax_freq.tick_params(axis='x')
    if group_var in ['BonusMalus','DrivAge', 'VehAge']:
        ax_freq.xaxis.set_major_locator(MaxNLocator(10))

    # Plot Severity
    sns.barplot(x=sev.index, y=sev.values, ax=ax_sev)
    ax_sev.set_title(f'Claim Severity by {group_var}')
    ax_sev.set_xlabel(group_var)
    ax_sev.set_ylabel('Average Claim Amount')
    ax_sev.tick_params(axis='x')
    ax_sev.yaxis.set_major_formatter(dollar_formatter)
    if group_var in ['BonusMalus', 'DrivAge']:
        ax_sev.xaxis.set_major_locator(MaxNLocator(10))
```

[612]:

```
# Variables to plot frequency and severity of claims by policyholder␣
 ↪characteristics
variables = ['VehAge', 'BonusMalus', 'DrivAge', 'VehPower', 'Area','Region',␣
 ↪'VehBrand', 'VehGas']

fig, axes = plt.subplots(nrows=len(variables), ncols=2, figsize=(20, 3 *␣
 ↪len(variables)))

# Loop through each variable and apply the plotting function on subplots
for idx, var in enumerate(variables):
    plot_variable(claims_data, var, axes[idx, 0], axes[idx, 1])

plt.tight_layout(pad=3.0)
plt.show()
```



2.0.1 Driver Age (Binned DrivAge)

Overview: The distribution of claims by driver age shows higher claim frequencies for younger and middle-aged drivers, and decreases with age.

Statistical Summary: - **Mean Age:** 45.48 years - **Standard Deviation:** 14.15 years - **Range:** 18 to 99 years

Justification: As discussed in class, younger drivers exhibit higher risk-seeking behavior, which could imply higher premiums for these subgroups.

Limitation: While age is a strong predictor for risk behavior across drivers and is important for our underwriting process, there are social concerns about using age as a rating variable in premium setting, which could lead to backlash.

2.0.2 Vehicle Age (Binned VehAge)

Overview: Vehicles that are 1-3 years old show the highest claim frequencies, suggesting a rapid depreciation in terms of safety or an increase in incidents due to other factors like increased usage.

Statistical Summary: - **Median Vehicle Age:** 6 years - **Interquartile Range:** 2 to 11 years

Justification: Newer vehicles might be associated with higher costs due to more expensive parts and repairs, influencing our premium calculations.

2.0.3 Vehicle Power (Binned VehPower)

Overview: Lower vehicle power correlates with higher claim frequencies but not severities, indicating riskier driving behavior in lower vehicle power classes.

Statistical Summary: - **Mean Power:** 6.46 - **Standard Deviation:** 2.06

Justification: Vehicles with lower power should potentially carry higher premiums due to an increased risk observed in our historical claims data.

2.0.4 Geographical Area and Vehicle Brand

Overview: Claims frequencies and severities vary significantly across different areas and brands, with some brands and regions showing markedly higher risk profiles.

Statistical Summary: - **Areas with Highest Claims:** Area C and D - **Brands with Higher Claims:** Brand B1 and B2

Justification: Premiums could be adjusted based on the geographical risk factors and vehicle brand-specific risks. However, it's important to note that by considering geographical region, we're putting ourselves at risk of redlining. Thus, it's important to include all individuals in our premium calculation, and to not exclude a particular risky region.

2.0.5 Geographical Regions (Region)

Overview: Claim frequencies and severities vary considerably across different regions, indicating geographic disparities in risk profiles.

Statistical Summary: - **Regions with High Claim Frequency:** Regions R24 and R82 - **Regions with High Claim Severity:** Regions R43 and R94

Justification: - **Premium Adjustments:** Fixing the insurance premiums based on regional risk assessments. Higher premiums could be justified in regions with frequent and severe claims. - **Risk Mitigation Strategies:** Implement targeted risk mitigation strategies such as awareness campaigns, improved road safety measures, and localized driving regulations to reduce claim frequencies and severities in these high-risk regions.

The same redlining risks are applicable for this variable.

2.0.6 Bonus-Malus System

Overview: An unclear trend where higher Bonus-Malus levels sometimes correlate with increased claim severities but not frequencies, and inversely for lower Bonus-Malus scores.

Statistical Summary: - **Mean Bonus-Malus:** 59.82 - **Standard Deviation:** 15.65

[613]:

```
# Get descriptive statistics from the claims data set
descriptive_stats = claims_data.describe()
descriptive_stats
```

[613]:

```
              IDpol       Exposure       VehPower         VehAge  \
count  1.000000e+05  100000.000000  100000.000000  100000.000000   
mean   2.617735e+06       0.528057       6.460230       6.992550   
std    1.643394e+06       0.364232       2.055641       5.637297   
min    1.500000e+01       0.002732       4.000000       0.000000   
25%    1.156127e+06       0.170000       5.000000       2.000000   
50%    2.271008e+06       0.490000       6.000000       6.000000   
75%    4.044791e+06       0.990000       7.000000      11.000000   
max    6.114324e+06       1.000000      15.000000     100.000000   

             DrivAge     BonusMalus      Density    ClaimAmount  \
count  100000.000000  100000.000000  100000.00000  100000.000000   
mean       45.483040      59.822980   1800.69569      76.599887   
std        14.154698      15.652541   3955.08311    1531.841302   
min        18.000000      50.000000      2.00000       0.000000   
25%        34.000000      50.000000     94.00000       0.000000   
50%        44.000000      50.000000    399.00000       0.000000   
75%        55.000000      65.000000   1658.00000       0.000000   
max        99.000000     230.000000  27000.00000  200000.000000   

           Frequency       Severity  
count  100000.000000  100000.000000  
mean        0.119194      70.764054  
std         2.141210    1448.674413  
min         0.000000       0.000000  
25%         0.000000       0.000000  
50%         0.000000       0.000000  
75%         0.000000       0.000000  
max       365.000004  200000.000000  
```

[673]:

```
claims_data
```

[673]:

```
         IDpol  Exposure Area  VehPower  VehAge  DrivAge  BonusMalus VehBrand  \
0      2271893      0.83    E         5      17       53          64       B2   
1      1111864      0.24    E         5       2       27          64       B3   
2        72908      0.50    E         7      11       67          50       B3   
```

```
3      2283027      0.08   B          5       8       28          60        B1
4      1123838      0.03   A         11       1       38          50        B2
…        …       …  …       …     …      …        …      …
99995    70445      1.00   C          5      11       37          56        B2
99996  4163362      0.22   E          6      13       58          50        B1
99997  2081912      1.00   E          5       1       49          50        B2
99998  2012998      0.71   D          9       9       36          54        B1
99999  3087666      0.53   C          9      14       35          51        B3

        VehGas  Density  … ClaimAmount  Frequency  Severity  \
0       Diesel     3317  …         0.0        0.0       0.0
1       Diesel     2740  …         0.0        0.0       0.0
2      Regular     4762  …         0.0        0.0       0.0
3       Diesel       64  …         0.0        0.0       0.0
4      Regular       16  …         0.0        0.0       0.0
…        …       …  …         …        …       …
99995   Diesel      317  …         0.0        0.0       0.0
99996   Diesel     4762  …         0.0        0.0       0.0
99997   Diesel     4998  …         0.0        0.0       0.0
99998  Regular     1541  …         0.0        0.0       0.0
99999  Regular      161  …         0.0        0.0       0.0

       Binned BonusMalus Binned DrivAge Binned VehAge Binned VehPower  \
0                  Bonus      51.0-53.0    15.0-100.0         4.0-5.0
1                  Bonus      25.0-28.0       1.0-2.0         4.0-5.0
2                  Bonus      65.0-72.0     10.0-12.0         6.0-7.0
3                  Bonus      25.0-28.0       6.0-8.0         4.0-5.0
4                  Bonus      36.0-38.0       0.0-1.0        9.0-15.0
…                  …             …            …             …
99995              Bonus      36.0-38.0     10.0-12.0         4.0-5.0
99996              Bonus      57.0-61.0     12.0-15.0         5.0-6.0
99997              Bonus      48.0-51.0       0.0-1.0         4.0-5.0
99998              Bonus      34.0-36.0      8.0-10.0         8.0-9.0
99999              Bonus      34.0-36.0     12.0-15.0         8.0-9.0

      Risk Cluster K-Mode  Predicted Risk Cluster  Total Loss
0                       2                       2         0.0
1                       0                       0         0.0
2                       2                       2         0.0
3                       0                       0         0.0
4                       0                       0         0.0
…                  …                      …         …
99995                   0                       0         0.0
99996                   1                       1         0.0
99997                   0                       0         0.0
99998                   1                       1         0.0
99999                   0                       0         0.0
```

```
[100000 rows x 21 columns]
```

2.0.7 Correlation between risk caracteristics

- Most of the correlations between the variables are **weak** (close to 0), indicating that there is **no strong linear relationship** between most pairs of variables.
- The only moderate correlation is **between DrivAge and BonusMalus** (-0.480037), suggesting that as the driver's age increases, the Bonus-Malus score tends to decrease.
- **Redundancy:** These two variables are correlated, so they provide similar information to the model. Including both might not add significant value.

```
[677]: # Check for correlation between risk caracteristics (VehPower, DrivAge,␣
        ↪BonusMalus, VehAge)
       claims_data[['VehPower', 'DrivAge', 'BonusMalus', 'VehAge']].corr()
```

```
[677]:             VehPower   DrivAge  BonusMalus    VehAge
       VehPower    1.000000  0.028375   -0.077144 -0.006254
       DrivAge     0.028375  1.000000   -0.480037 -0.057351
       BonusMalus -0.077144 -0.480037    1.000000  0.084034
       VehAge     -0.006254 -0.057351    0.084034  1.000000
```

2.0.8 Variables chosen for our analysis

Using the given actuarial criteria:

1. **Accuracy**: These variables have a clear link to expected costs and losses.

   - **Driver age** is linked to risk-taking behavior, with younger drivers often associated with higher risk.
   - **Vehicle age** can affect the likelihood of breakdowns or maintenance issues.
   - **Vehicle power** is correlated with driving speed and potential accident severity.
   - **Vehicle brand** may reflect repair costs and vehicle reliability, impacting the claim costs.
   - **Area** represents regional risk factors, such as accident rates, theft rates, and crime rates.

2. **Homogeneity**: By grouping drivers or vehicles into categories based on these variables, insurers can reduce the variability of expected claims within each group.

3. **Credibility**: These variables typically cover large groups of individuals or vehicles, providing sufficient data for statistical measures.

4. **Reliability or Predictive Stability**: These variables tend to have stable differences over time, making them reliable predictors.

These variables align well with actuarial criteria, making them suitable for insurance underwriting. Also, we have decided to not use region, in order to avoid the use of two very similar risk variables (area and region), reducing redundancy.

See appendix 1 for further descriptive analysis of our variables.

3 2. Risk Group Assignment Algorithm Development

Objective:

The objective was to develop a risk group assignment algorithm capable of categorizing insured individuals into distinct risk classes based on observable characteristics. This classification will help in tailoring insurance premiums and policies that correspond to the risk each client presents for the auto-insurance company.

3.1 2.1. Methodology

The development process of the risk classification system involved several key steps, from data preprocessing to clustering and model validation.

3.1.1 Data Preprocessing Categorization

Before applying the clustering model, a crucial step involves preprocessing the data to create meaningful subgroups among the insurance company's claimants based on their respective characteristics. This process not only helps in reducing noise and managing outliers in the clustering model but also enhances the interpretability of the clustering outcomes. Below, we outline the strategy for binning key continuous variables based on their distribution characteristics:

The features selected for binning include driver's age (`DrivAge`), vehicle age (`VehAge`), and vehicle power (`VehPower`). The binning process was carried out using quantile-based discretization to ensure each bin contained approximately the same number of instances, enhancing the uniformity of the data. A custom Python function was developed to automate the binning process.

The binning method described here is used for each feature in our risk classification algorithm:

[615]:

```
# Function to perform quantile binning and apply it to the claims data set
def quantile_binning_and_apply(claims_data, features, n_bins,␣
 ↪specific_bins=None):
    bin_labels_dict = {}
    for feature in features:
        # Used a specific number of bins for some features or n_bins if not␣
 ↪specified
        bins = specific_bins.get(feature, n_bins) if specific_bins else n_bins
        bin_edges = pd.qcut(claims_data[feature], q=bins, retbins=True,␣
 ↪duplicates='drop')[1]
        bin_edges[-1] += 1e-5
        labels = [f'{bin_edges[i]:.1f}-{bin_edges[i+1]:.1f}' for i in␣
 ↪range(len(bin_edges)-1)]
        claims_data[f'Binned {feature}'] = pd.cut(claims_data[feature],␣
 ↪bins=bin_edges, labels=labels, include_lowest=True, duplicates='drop')
        bin_labels_dict[f'Binned {feature}'] = labels
    return bin_labels_dict

# Treatment of BonusMalus as specified in the project description
claims_data['Binned BonusMalus'] = claims_data['BonusMalus'].apply(lambda x:␣
 ↪'Bonus' if x <= 100 else 'Malus')
```

In our data preprocessing, we've chosen to apply a more granular binning approach to the `DrivAge` feature compared to others like `VehAge` and `VehPower`. While the general approach assigns 10 bins to most features, `DrivAge` is assigned 20 bins. This enhanced granularity allows us to capture finer distinctions in the risk profiles associated with different age groups of drivers.

[616]:
```python
# Call the function for the specified features and number of bins (10)
# Define the numbers of bins for specific features (DrivAge), to get more␣
 ↪granularity into the bins classification
bin_labels_dict = quantile_binning_and_apply(claims_data, ['DrivAge',␣
 ↪'VehAge','VehPower'], 10, {'DrivAge': 20})

# Find the maximum length of the lists in the dictionary
max_length = max(len(v) for v in bin_labels_dict.values())

# Fill shorter lists with NaN values so they all have the same length
for key, value in bin_labels_dict.items():
    bin_labels_dict[key] = value + [None] * (max_length - len(value))

# Create a DataFrame from the dictionary of bin labels
df_bins = pd.DataFrame(bin_labels_dict)

# Display the resulting DataFrame
df_bins.head(3)
```

[616]:
```
  Binned DrivAge Binned VehAge Binned VehPower
0      18.0-25.0       0.0-1.0         4.0-5.0
1      25.0-28.0       1.0-2.0         5.0-6.0
2      28.0-30.0       2.0-3.0         6.0-7.0
```

3.1.2 Optimal Amount of Cluster (k) using the Elbow Method

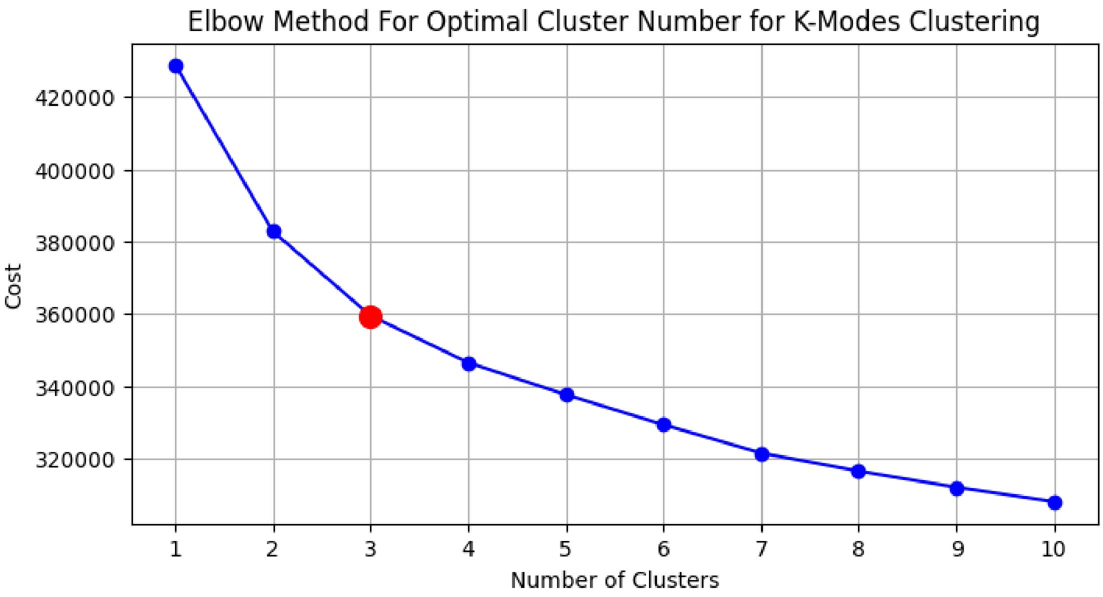
The elbow method is used to identify the optimal number of clusters for categorizing claims data by analyzing the rate of decrease in within-cluster variance as more clusters are added. This method helps spot the point, or "elbow," where adding additional clusters no longer provides significant improvement in variance reduction. This aids in choosing a number of clusters that balances detail and manageability, avoiding overfitting while still capturing meaningful patterns in the data.

The results from running the code for the Elbow method provided us with the following graph which indicates that the ideal number of clusters for the dataset is 3, if all 100 000 policyholders are used in the model.

Disclaimer: the code takes around 3 minutes to run

[620]:
```python
# Additional categorical columns
additional_categorical_columns = ['Area', 'VehBrand','BonusMalus']
features = ['DrivAge', 'VehAge','VehPower']

# Prepare data for one-hot encoding
```

```python
categorical_columns = [f'Binned {feature}' for feature in features] + additional_categorical_columns
for col in categorical_columns:
    claims_data[col] = claims_data[col].astype('category')

# Use only categorical columns for clustering
df_categorical = claims_data[categorical_columns]

# Convert the DataFrame to a numpy array before applying K-Modes
data_matrix = df_categorical.to_numpy()

# Apply the elbow method to find the optimal number of clusters for K-Modes clustering method
cost = []
K = range(1, 11)  # Range of clusters to try for the elbow method
for num_clusters in K:
    kmodes = KModes(n_clusters=num_clusters, random_state=42)
    clusters = kmodes.fit_predict(data_matrix)
    cost.append(kmodes.cost_)

# Plot the elbow graph
plt.figure(figsize=(8, 4))
plt.plot(K, cost, marker='o', color='b')
plt.xlabel('Number of Clusters')
plt.ylabel('Cost')
plt.title('Elbow Method For Optimal Cluster Number for K-Modes Clustering')
plt.xticks(K)
plt.grid(True)

# Highlight the optimal number of clusters (3) in red
optimal_k = 3
plt.plot(optimal_k, cost[optimal_k - 1], marker='o', markersize=10, color='r')
plt.show()
```



3.1.3 Implementation of `K-Modes` Clustering Method

It's crucial to select an appropriate model for clustering our data. After testing various clustering methods such as `DBSCAN` (unsupervised algorithm), `K-means`, `K-prototype`, and `K-Medoids`, we decided to employ the `K-Modes` clustering method, which is particularly suitable for categorical data. The data was prepared by combining previously binned features with additional categorical variables such as `Area` and `VehBrand` to enrich our clustering analysis. The following Python code snippet illustrates the implementation of the K-Modes clustering method:

For more details, we applied the K-Modes clustering algorithm as follows:

1. **Algorithm Initialization:** We initialized the K-Modes algorithm with a specified number of clusters (K=3).

2. **Model Fitting:** The K-Modes model was fitted to the one-hot encoded categorical matrix derived from our data. This step involves the iterative relocation of data points to the nearest clusters and recalculating the modes of each cluster.

3. **Cluster Assignment:** Then, the clustering model assigns each data point to one of the three clusters based on the minimal dissimilarity, measured by the Hamming distance, which counts the number of differing characteristics between data points and the cluster modes.

[623]:
```python
# Additional categorical columns
additional_categorical_columns = ['Area', 'VehBrand']

def apply_k_modes_clustering(claims_data, features,␣
 ↪additional_categorical_columns, n_clusters, random_state):
    """
    Apply K-Modes clustering on categorical data.
    Parameters:
        claims_data (DataFrame): The DataFrame containing the claims data.
        features (list): List of features to be binned and included in the␣
 ↪clustering.
        additional_categorical_columns (list): List of additional categorical␣
 ↪columns to include.
        n_clusters (int, optional): The number of clusters to form. Defaults to␣
 ↪3.
        random_state (int, optional): Random state for reproducibility.␣
 ↪Defaults to 42.

    Returns:
        DataFrame: The DataFrame with an additional column 'Risk Cluster␣
 ↪K-Mode' indicating the cluster assignment.
    """
    # Combine binned features and additional categorical columns
    categorical_columns = [f'Binned {feature}' for feature in features] +␣
 ↪additional_categorical_columns

    # Convert categorical columns to category type
    for col in categorical_columns:
        claims_data[col] = claims_data[col].astype('category')

    # Use only categorical columns for clustering
    df_categorical = claims_data[categorical_columns]

    # Convert the DataFrame to a numpy array
    data_matrix = df_categorical.to_numpy()

    # Apply K-Modes clustering
    kmodes = KModes(n_clusters=n_clusters, random_state=random_state)
    clusters = kmodes.fit_predict(data_matrix)

    # Store the result in 'Risk Cluster K-Mode' column
    claims_data['Risk Cluster K-Mode'] = clusters

    return claims_data

# Implementing the function
updated_claims_data = apply_k_modes_clustering(
    claims_data=claims_data,
    features=features,
    additional_categorical_columns=additional_categorical_columns,
    n_clusters=3,
    random_state=42  # Seed for reproducibility
)
```

3.1.4 Validate the clusters using `CatBoostClassifier`

Following the identification of risk clusters using the K-Modes clustering method, the next step of our approach for risk classification involves employing a CatBoost classifier to validate the three prior clusters identified.

CatBoost is an algorithm for gradient boosting on decision trees, designed to work well with categorical data. We particularly favored this supervised machine learning algorithm for its efficiency, accuracy, and simplicity in handling categorical data.

The hyperparameters - `iterations`, `learning rate`, `depth` and `test size` - were optimized using `GridSearchCV`, a systematic method for parameter tuning through cross-validation.

The `CatBoost classifier` was then setup and trained using these optimized parameters and the pre defined clusters from the K-Modes method:

[624]:

```
# Extract labels from the 'Risk Cluster' column of K-Modes clustering
labels = claims_data['Risk Cluster K-Mode']

# Identify categorical feature indices for CatBoost model
cat_features_indices = [df_categorical.columns.get_loc(col) for col in␣
 ↪categorical_columns]

# Split the data into training and testing sets (20% test data) and use the␣
 ↪rest as training data (20% was the best split ratio based on GridSearchCV␣
 ↪results optimization run)
X_train, X_test, y_train, y_test = train_test_split(df_categorical, labels,␣
 ↪test_size=0.2, random_state=42)

# Train a CatBoost classifier with adjusted parameters
model = CatBoostClassifier(
    iterations=500,          # Number of iterations are based on GridSearchCV␣
 ↪results optimization run to find the best parameters
    learning_rate=0.1,      # Number of iterations are based on GridSearchCV␣
 ↪results optimization run to find the best parameters
    depth=10,                # Depth of the tree are based on GridSearchCV␣
 ↪results optimization run to find the best parameters
    random_seed=42,          # Random seed for reproducibility
    cat_features=cat_features_indices # Categorical feature indices for␣
 ↪CatBoost model to use
    )
model.fit(X_train, y_train)

# Predict on the entire dataset to get the predicted risk clusters (Validation)
claims_data['Predicted Risk Cluster'] = model.predict(df_categorical)
```

```
0:      learn: 0.9568327        total: 189ms    remaining: 1m 34s
```

```
1:      learn: 0.8454651        total: 352ms    remaining: 1m 27s
2:      learn: 0.7534016        total: 459ms    remaining: 1m 16s
3:      learn: 0.6824964        total: 562ms    remaining: 1m 9s
4:      learn: 0.6182636        total: 667ms    remaining: 1m 5s
5:      learn: 0.5666129        total: 806ms    remaining: 1m 6s
6:      learn: 0.5201568        total: 962ms    remaining: 1m 7s
7:      learn: 0.4699258        total: 1.07s    remaining: 1m 6s
8:      learn: 0.4270891        total: 1.25s    remaining: 1m 8s
9:      learn: 0.3869494        total: 1.42s    remaining: 1m 9s
10:     learn: 0.3522274        total: 1.58s    remaining: 1m 10s
11:     learn: 0.3221052        total: 1.7s     remaining: 1m 9s
12:     learn: 0.2961555        total: 1.84s    remaining: 1m 8s
13:     learn: 0.2735080        total: 1.99s    remaining: 1m 8s
14:     learn: 0.2507718        total: 2.12s    remaining: 1m 8s
15:     learn: 0.2322255        total: 2.25s    remaining: 1m 8s
16:     learn: 0.2150353        total: 2.39s    remaining: 1m 7s
17:     learn: 0.2001648        total: 2.52s    remaining: 1m 7s
18:     learn: 0.1860877        total: 2.76s    remaining: 1m 9s
19:     learn: 0.1734123        total: 2.89s    remaining: 1m 9s
20:     learn: 0.1612206        total: 3.04s    remaining: 1m 9s
21:     learn: 0.1486730        total: 3.17s    remaining: 1m 8s
22:     learn: 0.1372958        total: 3.31s    remaining: 1m 8s
23:     learn: 0.1268489        total: 3.44s    remaining: 1m 8s
24:     learn: 0.1177552        total: 3.69s    remaining: 1m 10s
25:     learn: 0.1092655        total: 3.96s    remaining: 1m 12s
26:     learn: 0.1017166        total: 4.31s    remaining: 1m 15s
27:     learn: 0.0948643        total: 4.63s    remaining: 1m 18s
28:     learn: 0.0887449        total: 4.9s     remaining: 1m 19s
29:     learn: 0.0837533        total: 5.17s    remaining: 1m 21s
30:     learn: 0.0786964        total: 5.49s    remaining: 1m 23s
31:     learn: 0.0741440        total: 5.79s    remaining: 1m 24s
32:     learn: 0.0701457        total: 6.08s    remaining: 1m 26s
33:     learn: 0.0662065        total: 6.33s    remaining: 1m 26s
34:     learn: 0.0630863        total: 6.59s    remaining: 1m 27s
35:     learn: 0.0600795        total: 6.9s     remaining: 1m 28s
36:     learn: 0.0573485        total: 7.22s    remaining: 1m 30s
37:     learn: 0.0545086        total: 7.47s    remaining: 1m 30s
38:     learn: 0.0521497        total: 7.77s    remaining: 1m 31s
39:     learn: 0.0499581        total: 8.08s    remaining: 1m 32s
40:     learn: 0.0480696        total: 8.4s     remaining: 1m 34s
41:     learn: 0.0462265        total: 8.74s    remaining: 1m 35s
42:     learn: 0.0443464        total: 9.03s    remaining: 1m 36s
43:     learn: 0.0429101        total: 9.33s    remaining: 1m 36s
44:     learn: 0.0415788        total: 9.62s    remaining: 1m 37s
45:     learn: 0.0398407        total: 9.91s    remaining: 1m 37s
46:     learn: 0.0383131        total: 10.2s    remaining: 1m 38s
47:     learn: 0.0373745        total: 10.5s    remaining: 1m 38s
48:     learn: 0.0360308        total: 10.8s    remaining: 1m 39s
```

```
49:     learn: 0.0349161        total: 11.1s    remaining: 1m 39s
50:     learn: 0.0338479        total: 11.4s    remaining: 1m 40s
51:     learn: 0.0328240        total: 11.7s    remaining: 1m 40s
52:     learn: 0.0320464        total: 12s      remaining: 1m 41s
53:     learn: 0.0313447        total: 12.3s    remaining: 1m 41s
54:     learn: 0.0307608        total: 12.6s    remaining: 1m 41s
55:     learn: 0.0301421        total: 12.9s    remaining: 1m 42s
56:     learn: 0.0292158        total: 13.2s    remaining: 1m 42s
57:     learn: 0.0285632        total: 13.6s    remaining: 1m 43s
58:     learn: 0.0278473        total: 13.9s    remaining: 1m 43s
59:     learn: 0.0272398        total: 14.2s    remaining: 1m 43s
60:     learn: 0.0269828        total: 14.5s    remaining: 1m 44s
61:     learn: 0.0264628        total: 14.8s    remaining: 1m 44s
62:     learn: 0.0258635        total: 15.1s    remaining: 1m 44s
63:     learn: 0.0256369        total: 15.4s    remaining: 1m 44s
64:     learn: 0.0250451        total: 15.7s    remaining: 1m 44s
65:     learn: 0.0244187        total: 16s      remaining: 1m 45s
66:     learn: 0.0239400        total: 16.3s    remaining: 1m 45s
67:     learn: 0.0234853        total: 16.6s    remaining: 1m 45s
68:     learn: 0.0229701        total: 16.9s    remaining: 1m 45s
69:     learn: 0.0225650        total: 17.2s    remaining: 1m 45s
70:     learn: 0.0221565        total: 17.5s    remaining: 1m 45s
71:     learn: 0.0218706        total: 17.8s    remaining: 1m 45s
72:     learn: 0.0215014        total: 18.1s    remaining: 1m 45s
73:     learn: 0.0210533        total: 18.4s    remaining: 1m 45s
74:     learn: 0.0207383        total: 18.7s    remaining: 1m 45s
75:     learn: 0.0204417        total: 18.9s    remaining: 1m 45s
76:     learn: 0.0202692        total: 19.2s    remaining: 1m 45s
77:     learn: 0.0201651        total: 19.5s    remaining: 1m 45s
78:     learn: 0.0200065        total: 19.8s    remaining: 1m 45s
79:     learn: 0.0198869        total: 20.1s    remaining: 1m 45s
80:     learn: 0.0197744        total: 20.3s    remaining: 1m 45s
81:     learn: 0.0195124        total: 20.6s    remaining: 1m 45s
82:     learn: 0.0192972        total: 20.9s    remaining: 1m 45s
83:     learn: 0.0190524        total: 21.2s    remaining: 1m 45s
84:     learn: 0.0187621        total: 21.5s    remaining: 1m 44s
85:     learn: 0.0184821        total: 21.8s    remaining: 1m 44s
86:     learn: 0.0183555        total: 22.1s    remaining: 1m 44s
87:     learn: 0.0181094        total: 22.4s    remaining: 1m 44s
88:     learn: 0.0179919        total: 22.7s    remaining: 1m 44s
89:     learn: 0.0177790        total: 23s      remaining: 1m 44s
90:     learn: 0.0176576        total: 23.3s    remaining: 1m 44s
91:     learn: 0.0175027        total: 23.5s    remaining: 1m 44s
92:     learn: 0.0173661        total: 23.8s    remaining: 1m 44s
93:     learn: 0.0173046        total: 24.1s    remaining: 1m 43s
94:     learn: 0.0172128        total: 24.4s    remaining: 1m 43s
95:     learn: 0.0170639        total: 24.7s    remaining: 1m 44s
96:     learn: 0.0169559        total: 25s      remaining: 1m 44s
```

```
97:     learn: 0.0168928        total: 25.4s    remaining: 1m 44s
98:     learn: 0.0167666        total: 25.9s    remaining: 1m 44s
99:     learn: 0.0165357        total: 26.4s    remaining: 1m 45s
100:    learn: 0.0164127        total: 26.8s    remaining: 1m 45s
101:    learn: 0.0163370        total: 27s      remaining: 1m 45s
102:    learn: 0.0161803        total: 27.4s    remaining: 1m 45s
103:    learn: 0.0161366        total: 27.7s    remaining: 1m 45s
104:    learn: 0.0159902        total: 28.1s    remaining: 1m 45s
105:    learn: 0.0158937        total: 28.7s    remaining: 1m 46s
106:    learn: 0.0158032        total: 29s      remaining: 1m 46s
107:    learn: 0.0157450        total: 29.3s    remaining: 1m 46s
108:    learn: 0.0155907        total: 29.6s    remaining: 1m 46s
109:    learn: 0.0154700        total: 29.9s    remaining: 1m 45s
110:    learn: 0.0154062        total: 30.2s    remaining: 1m 45s
111:    learn: 0.0153599        total: 30.4s    remaining: 1m 45s
112:    learn: 0.0152535        total: 30.7s    remaining: 1m 45s
113:    learn: 0.0151402        total: 31s      remaining: 1m 45s
114:    learn: 0.0149583        total: 31.6s    remaining: 1m 45s
115:    learn: 0.0148928        total: 32.1s    remaining: 1m 46s
116:    learn: 0.0147909        total: 32.6s    remaining: 1m 46s
117:    learn: 0.0147104        total: 33s      remaining: 1m 46s
118:    learn: 0.0146455        total: 33.3s    remaining: 1m 46s
119:    learn: 0.0145867        total: 33.7s    remaining: 1m 46s
120:    learn: 0.0145502        total: 33.9s    remaining: 1m 46s
121:    learn: 0.0145147        total: 34.2s    remaining: 1m 45s
122:    learn: 0.0143746        total: 34.5s    remaining: 1m 45s
123:    learn: 0.0142858        total: 34.8s    remaining: 1m 45s
124:    learn: 0.0141895        total: 35s      remaining: 1m 45s
125:    learn: 0.0140932        total: 35.3s    remaining: 1m 44s
126:    learn: 0.0140141        total: 35.6s    remaining: 1m 44s
127:    learn: 0.0138956        total: 35.9s    remaining: 1m 44s
128:    learn: 0.0137942        total: 36.2s    remaining: 1m 44s
129:    learn: 0.0137557        total: 36.5s    remaining: 1m 43s
130:    learn: 0.0136131        total: 36.8s    remaining: 1m 43s
131:    learn: 0.0135207        total: 37.1s    remaining: 1m 43s
132:    learn: 0.0134537        total: 37.4s    remaining: 1m 43s
133:    learn: 0.0133677        total: 37.7s    remaining: 1m 42s
134:    learn: 0.0132932        total: 38s      remaining: 1m 42s
135:    learn: 0.0132439        total: 38.2s    remaining: 1m 42s
136:    learn: 0.0131861        total: 38.5s    remaining: 1m 41s
137:    learn: 0.0131086        total: 38.8s    remaining: 1m 41s
138:    learn: 0.0130537        total: 39.1s    remaining: 1m 41s
139:    learn: 0.0129548        total: 39.4s    remaining: 1m 41s
140:    learn: 0.0128447        total: 39.6s    remaining: 1m 40s
141:    learn: 0.0127503        total: 39.9s    remaining: 1m 40s
142:    learn: 0.0127266        total: 40.2s    remaining: 1m 40s
143:    learn: 0.0126854        total: 40.5s    remaining: 1m 40s
144:    learn: 0.0125422        total: 40.8s    remaining: 1m 39s
```

```
145:    learn: 0.0124932        total: 41.1s    remaining: 1m 39s
146:    learn: 0.0124329        total: 41.4s    remaining: 1m 39s
147:    learn: 0.0123611        total: 41.7s    remaining: 1m 39s
148:    learn: 0.0123030        total: 42s      remaining: 1m 38s
149:    learn: 0.0122243        total: 42.3s    remaining: 1m 38s
150:    learn: 0.0121719        total: 42.6s    remaining: 1m 38s
151:    learn: 0.0121147        total: 42.8s    remaining: 1m 38s
152:    learn: 0.0120649        total: 43.1s    remaining: 1m 37s
153:    learn: 0.0119988        total: 43.4s    remaining: 1m 37s
154:    learn: 0.0119470        total: 43.7s    remaining: 1m 37s
155:    learn: 0.0119016        total: 44s      remaining: 1m 36s
156:    learn: 0.0117691        total: 44.3s    remaining: 1m 36s
157:    learn: 0.0117242        total: 44.6s    remaining: 1m 36s
158:    learn: 0.0116882        total: 44.8s    remaining: 1m 36s
159:    learn: 0.0116254        total: 45.1s    remaining: 1m 35s
160:    learn: 0.0115733        total: 45.4s    remaining: 1m 35s
161:    learn: 0.0115355        total: 45.7s    remaining: 1m 35s
162:    learn: 0.0114250        total: 46s      remaining: 1m 35s
163:    learn: 0.0113522        total: 46.2s    remaining: 1m 34s
164:    learn: 0.0112953        total: 46.5s    remaining: 1m 34s
165:    learn: 0.0112531        total: 46.8s    remaining: 1m 34s
166:    learn: 0.0111742        total: 47.1s    remaining: 1m 33s
167:    learn: 0.0111694        total: 47.4s    remaining: 1m 33s
168:    learn: 0.0111249        total: 47.6s    remaining: 1m 33s
169:    learn: 0.0110393        total: 47.9s    remaining: 1m 33s
170:    learn: 0.0109510        total: 48.2s    remaining: 1m 32s
171:    learn: 0.0108900        total: 48.5s    remaining: 1m 32s
172:    learn: 0.0108675        total: 48.8s    remaining: 1m 32s
173:    learn: 0.0108130        total: 49.1s    remaining: 1m 31s
174:    learn: 0.0107716        total: 49.4s    remaining: 1m 31s
175:    learn: 0.0107198        total: 49.7s    remaining: 1m 31s
176:    learn: 0.0106631        total: 50s      remaining: 1m 31s
177:    learn: 0.0105705        total: 50.3s    remaining: 1m 30s
178:    learn: 0.0105008        total: 50.5s    remaining: 1m 30s
179:    learn: 0.0104590        total: 50.8s    remaining: 1m 30s
180:    learn: 0.0104266        total: 51.1s    remaining: 1m 30s
181:    learn: 0.0103807        total: 51.4s    remaining: 1m 29s
182:    learn: 0.0103548        total: 51.7s    remaining: 1m 29s
183:    learn: 0.0102920        total: 51.9s    remaining: 1m 29s
184:    learn: 0.0102621        total: 52.2s    remaining: 1m 28s
185:    learn: 0.0101750        total: 52.5s    remaining: 1m 28s
186:    learn: 0.0101499        total: 52.8s    remaining: 1m 28s
187:    learn: 0.0101074        total: 53.1s    remaining: 1m 28s
188:    learn: 0.0100639        total: 53.3s    remaining: 1m 27s
189:    learn: 0.0100332        total: 53.6s    remaining: 1m 27s
190:    learn: 0.0099774        total: 53.9s    remaining: 1m 27s
191:    learn: 0.0099312        total: 54.2s    remaining: 1m 26s
192:    learn: 0.0098994        total: 54.5s    remaining: 1m 26s
```

```
193:    learn: 0.0098726        total: 54.7s    remaining: 1m 26s
194:    learn: 0.0098431        total: 55s      remaining: 1m 26s
195:    learn: 0.0098222        total: 55.3s    remaining: 1m 25s
196:    learn: 0.0097631        total: 55.6s    remaining: 1m 25s
197:    learn: 0.0096856        total: 55.9s    remaining: 1m 25s
198:    learn: 0.0096347        total: 56.2s    remaining: 1m 25s
199:    learn: 0.0096120        total: 56.5s    remaining: 1m 24s
200:    learn: 0.0095743        total: 56.8s    remaining: 1m 24s
201:    learn: 0.0095096        total: 57s      remaining: 1m 24s
202:    learn: 0.0094714        total: 57.3s    remaining: 1m 23s
203:    learn: 0.0094330        total: 57.6s    remaining: 1m 23s
204:    learn: 0.0093723        total: 57.9s    remaining: 1m 23s
205:    learn: 0.0093001        total: 58.2s    remaining: 1m 23s
206:    learn: 0.0092544        total: 58.5s    remaining: 1m 22s
207:    learn: 0.0091989        total: 58.9s    remaining: 1m 22s
208:    learn: 0.0091786        total: 59.2s    remaining: 1m 22s
209:    learn: 0.0091326        total: 59.5s    remaining: 1m 22s
210:    learn: 0.0090458        total: 59.8s    remaining: 1m 21s
211:    learn: 0.0090006        total: 1m       remaining: 1m 21s
212:    learn: 0.0089812        total: 1m       remaining: 1m 21s
213:    learn: 0.0089478        total: 1m       remaining: 1m 21s
214:    learn: 0.0089359        total: 1m 1s    remaining: 1m 20s
215:    learn: 0.0088980        total: 1m 1s    remaining: 1m 20s
216:    learn: 0.0088759        total: 1m 1s    remaining: 1m 20s
217:    learn: 0.0088431        total: 1m 2s    remaining: 1m 20s
218:    learn: 0.0087635        total: 1m 2s    remaining: 1m 20s
219:    learn: 0.0087444        total: 1m 2s    remaining: 1m 19s
220:    learn: 0.0086865        total: 1m 3s    remaining: 1m 19s
221:    learn: 0.0086644        total: 1m 3s    remaining: 1m 19s
222:    learn: 0.0086042        total: 1m 3s    remaining: 1m 18s
223:    learn: 0.0085527        total: 1m 3s    remaining: 1m 18s
224:    learn: 0.0085085        total: 1m 4s    remaining: 1m 18s
225:    learn: 0.0084313        total: 1m 4s    remaining: 1m 18s
226:    learn: 0.0083897        total: 1m 4s    remaining: 1m 17s
227:    learn: 0.0083633        total: 1m 4s    remaining: 1m 17s
228:    learn: 0.0083489        total: 1m 5s    remaining: 1m 17s
229:    learn: 0.0083226        total: 1m 5s    remaining: 1m 16s
230:    learn: 0.0082928        total: 1m 5s    remaining: 1m 16s
231:    learn: 0.0082232        total: 1m 6s    remaining: 1m 16s
232:    learn: 0.0081780        total: 1m 6s    remaining: 1m 16s
233:    learn: 0.0081321        total: 1m 6s    remaining: 1m 15s
234:    learn: 0.0081062        total: 1m 6s    remaining: 1m 15s
235:    learn: 0.0080680        total: 1m 7s    remaining: 1m 15s
236:    learn: 0.0080545        total: 1m 7s    remaining: 1m 14s
237:    learn: 0.0080200        total: 1m 7s    remaining: 1m 14s
238:    learn: 0.0080005        total: 1m 8s    remaining: 1m 14s
239:    learn: 0.0079518        total: 1m 8s    remaining: 1m 14s
240:    learn: 0.0079016        total: 1m 8s    remaining: 1m 13s
```

```
241:    learn: 0.0078253        total: 1m 9s    remaining: 1m 13s
242:    learn: 0.0078004        total: 1m 9s    remaining: 1m 13s
243:    learn: 0.0077537        total: 1m 9s    remaining: 1m 13s
244:    learn: 0.0077305        total: 1m 10s   remaining: 1m 13s
245:    learn: 0.0077125        total: 1m 10s   remaining: 1m 12s
246:    learn: 0.0077006        total: 1m 10s   remaining: 1m 12s
247:    learn: 0.0076673        total: 1m 11s   remaining: 1m 12s
248:    learn: 0.0076390        total: 1m 11s   remaining: 1m 12s
249:    learn: 0.0076180        total: 1m 11s   remaining: 1m 11s
250:    learn: 0.0075889        total: 1m 12s   remaining: 1m 11s
251:    learn: 0.0075614        total: 1m 12s   remaining: 1m 11s
252:    learn: 0.0075228        total: 1m 12s   remaining: 1m 10s
253:    learn: 0.0074654        total: 1m 13s   remaining: 1m 10s
254:    learn: 0.0074262        total: 1m 13s   remaining: 1m 10s
255:    learn: 0.0074054        total: 1m 13s   remaining: 1m 10s
256:    learn: 0.0073842        total: 1m 14s   remaining: 1m 10s
257:    learn: 0.0073451        total: 1m 14s   remaining: 1m 9s
258:    learn: 0.0072899        total: 1m 14s   remaining: 1m 9s
259:    learn: 0.0072517        total: 1m 14s   remaining: 1m 9s
260:    learn: 0.0072322        total: 1m 15s   remaining: 1m 8s
261:    learn: 0.0072271        total: 1m 15s   remaining: 1m 8s
262:    learn: 0.0072081        total: 1m 15s   remaining: 1m 8s
263:    learn: 0.0071872        total: 1m 16s   remaining: 1m 8s
264:    learn: 0.0071478        total: 1m 16s   remaining: 1m 7s
265:    learn: 0.0071143        total: 1m 16s   remaining: 1m 7s
266:    learn: 0.0070926        total: 1m 17s   remaining: 1m 7s
267:    learn: 0.0070594        total: 1m 17s   remaining: 1m 6s
268:    learn: 0.0070088        total: 1m 17s   remaining: 1m 6s
269:    learn: 0.0069826        total: 1m 17s   remaining: 1m 6s
270:    learn: 0.0069554        total: 1m 18s   remaining: 1m 6s
271:    learn: 0.0069310        total: 1m 18s   remaining: 1m 5s
272:    learn: 0.0069098        total: 1m 18s   remaining: 1m 5s
273:    learn: 0.0068850        total: 1m 19s   remaining: 1m 5s
274:    learn: 0.0068404        total: 1m 19s   remaining: 1m 5s
275:    learn: 0.0068213        total: 1m 19s   remaining: 1m 4s
276:    learn: 0.0068054        total: 1m 20s   remaining: 1m 4s
277:    learn: 0.0067945        total: 1m 20s   remaining: 1m 4s
278:    learn: 0.0067536        total: 1m 20s   remaining: 1m 3s
279:    learn: 0.0067100        total: 1m 20s   remaining: 1m 3s
280:    learn: 0.0066809        total: 1m 21s   remaining: 1m 3s
281:    learn: 0.0066610        total: 1m 21s   remaining: 1m 3s
282:    learn: 0.0066393        total: 1m 21s   remaining: 1m 2s
283:    learn: 0.0066144        total: 1m 22s   remaining: 1m 2s
284:    learn: 0.0065918        total: 1m 22s   remaining: 1m 2s
285:    learn: 0.0065565        total: 1m 22s   remaining: 1m 1s
286:    learn: 0.0065394        total: 1m 22s   remaining: 1m 1s
287:    learn: 0.0065167        total: 1m 23s   remaining: 1m 1s
288:    learn: 0.0065034        total: 1m 23s   remaining: 1m
```

```
289:    learn: 0.0064766        total: 1m 23s   remaining: 1m
290:    learn: 0.0064336        total: 1m 24s   remaining: 1m
291:    learn: 0.0064054        total: 1m 24s   remaining: 1m
292:    learn: 0.0063896        total: 1m 24s   remaining: 59.9s
293:    learn: 0.0063586        total: 1m 25s   remaining: 59.6s
294:    learn: 0.0063409        total: 1m 25s   remaining: 59.3s
295:    learn: 0.0063193        total: 1m 25s   remaining: 59s
296:    learn: 0.0062760        total: 1m 25s   remaining: 58.7s
297:    learn: 0.0062509        total: 1m 26s   remaining: 58.4s
298:    learn: 0.0062340        total: 1m 26s   remaining: 58.1s
299:    learn: 0.0062232        total: 1m 26s   remaining: 57.9s
300:    learn: 0.0061984        total: 1m 27s   remaining: 57.6s
301:    learn: 0.0061739        total: 1m 27s   remaining: 57.3s
302:    learn: 0.0061566        total: 1m 27s   remaining: 57s
303:    learn: 0.0061412        total: 1m 27s   remaining: 56.7s
304:    learn: 0.0061065        total: 1m 28s   remaining: 56.4s
305:    learn: 0.0060697        total: 1m 28s   remaining: 56.1s
306:    learn: 0.0060572        total: 1m 28s   remaining: 55.9s
307:    learn: 0.0060137        total: 1m 29s   remaining: 55.6s
308:    learn: 0.0059713        total: 1m 29s   remaining: 55.3s
309:    learn: 0.0059405        total: 1m 29s   remaining: 55s
310:    learn: 0.0059123        total: 1m 29s   remaining: 54.7s
311:    learn: 0.0058887        total: 1m 30s   remaining: 54.4s
312:    learn: 0.0058797        total: 1m 30s   remaining: 54.1s
313:    learn: 0.0058520        total: 1m 30s   remaining: 53.8s
314:    learn: 0.0058263        total: 1m 31s   remaining: 53.5s
315:    learn: 0.0058090        total: 1m 31s   remaining: 53.2s
316:    learn: 0.0057853        total: 1m 31s   remaining: 52.9s
317:    learn: 0.0057713        total: 1m 32s   remaining: 52.7s
318:    learn: 0.0057392        total: 1m 32s   remaining: 52.4s
319:    learn: 0.0057124        total: 1m 32s   remaining: 52.1s
320:    learn: 0.0056880        total: 1m 32s   remaining: 51.8s
321:    learn: 0.0056824        total: 1m 33s   remaining: 51.6s
322:    learn: 0.0056702        total: 1m 33s   remaining: 51.3s
323:    learn: 0.0056452        total: 1m 33s   remaining: 51s
324:    learn: 0.0056312        total: 1m 34s   remaining: 50.7s
325:    learn: 0.0056041        total: 1m 34s   remaining: 50.4s
326:    learn: 0.0055746        total: 1m 34s   remaining: 50.1s
327:    learn: 0.0055583        total: 1m 35s   remaining: 49.8s
328:    learn: 0.0055399        total: 1m 35s   remaining: 49.5s
329:    learn: 0.0055170        total: 1m 35s   remaining: 49.3s
330:    learn: 0.0055031        total: 1m 35s   remaining: 49s
331:    learn: 0.0054961        total: 1m 36s   remaining: 48.7s
332:    learn: 0.0054898        total: 1m 36s   remaining: 48.4s
333:    learn: 0.0054653        total: 1m 36s   remaining: 48.1s
334:    learn: 0.0054424        total: 1m 37s   remaining: 47.8s
335:    learn: 0.0054343        total: 1m 37s   remaining: 47.5s
336:    learn: 0.0054090        total: 1m 37s   remaining: 47.2s
```

```
337:    learn: 0.0053899        total: 1m 37s   remaining: 46.9s
338:    learn: 0.0053622        total: 1m 38s   remaining: 46.6s
339:    learn: 0.0053355        total: 1m 38s   remaining: 46.4s
340:    learn: 0.0053271        total: 1m 38s   remaining: 46.1s
341:    learn: 0.0053120        total: 1m 39s   remaining: 45.8s
342:    learn: 0.0053044        total: 1m 39s   remaining: 45.5s
343:    learn: 0.0052743        total: 1m 39s   remaining: 45.2s
344:    learn: 0.0052584        total: 1m 40s   remaining: 44.9s
345:    learn: 0.0052333        total: 1m 40s   remaining: 44.6s
346:    learn: 0.0052048        total: 1m 40s   remaining: 44.4s
347:    learn: 0.0051920        total: 1m 40s   remaining: 44.1s
348:    learn: 0.0051803        total: 1m 41s   remaining: 43.8s
349:    learn: 0.0051667        total: 1m 41s   remaining: 43.5s
350:    learn: 0.0051418        total: 1m 41s   remaining: 43.2s
351:    learn: 0.0051177        total: 1m 42s   remaining: 43s
352:    learn: 0.0050876        total: 1m 42s   remaining: 42.7s
353:    learn: 0.0050557        total: 1m 42s   remaining: 42.4s
354:    learn: 0.0050416        total: 1m 43s   remaining: 42.1s
355:    learn: 0.0050285        total: 1m 43s   remaining: 41.8s
356:    learn: 0.0050177        total: 1m 43s   remaining: 41.5s
357:    learn: 0.0050051        total: 1m 44s   remaining: 41.3s
358:    learn: 0.0049964        total: 1m 44s   remaining: 41s
359:    learn: 0.0049864        total: 1m 44s   remaining: 40.7s
360:    learn: 0.0049719        total: 1m 44s   remaining: 40.3s
361:    learn: 0.0049466        total: 1m 45s   remaining: 40.1s
362:    learn: 0.0049390        total: 1m 45s   remaining: 39.8s
363:    learn: 0.0049273        total: 1m 45s   remaining: 39.5s
364:    learn: 0.0049178        total: 1m 46s   remaining: 39.2s
365:    learn: 0.0049043        total: 1m 46s   remaining: 38.9s
366:    learn: 0.0048732        total: 1m 46s   remaining: 38.6s
367:    learn: 0.0048543        total: 1m 46s   remaining: 38.4s
368:    learn: 0.0048312        total: 1m 47s   remaining: 38.1s
369:    learn: 0.0048213        total: 1m 47s   remaining: 37.8s
370:    learn: 0.0048102        total: 1m 47s   remaining: 37.5s
371:    learn: 0.0047895        total: 1m 48s   remaining: 37.2s
372:    learn: 0.0047795        total: 1m 48s   remaining: 36.9s
373:    learn: 0.0047611        total: 1m 48s   remaining: 36.6s
374:    learn: 0.0047431        total: 1m 48s   remaining: 36.3s
375:    learn: 0.0047258        total: 1m 49s   remaining: 36s
376:    learn: 0.0047113        total: 1m 49s   remaining: 35.7s
377:    learn: 0.0047066        total: 1m 49s   remaining: 35.4s
378:    learn: 0.0046862        total: 1m 49s   remaining: 35.1s
379:    learn: 0.0046643        total: 1m 50s   remaining: 34.8s
380:    learn: 0.0046397        total: 1m 50s   remaining: 34.5s
381:    learn: 0.0046328        total: 1m 50s   remaining: 34.2s
382:    learn: 0.0046120        total: 1m 51s   remaining: 33.9s
383:    learn: 0.0045876        total: 1m 51s   remaining: 33.6s
384:    learn: 0.0045592        total: 1m 51s   remaining: 33.3s
```

```
385:    learn: 0.0045491        total: 1m 51s   remaining: 33s
386:    learn: 0.0045382        total: 1m 52s   remaining: 32.8s
387:    learn: 0.0045306        total: 1m 52s   remaining: 32.5s
388:    learn: 0.0045212        total: 1m 52s   remaining: 32.2s
389:    learn: 0.0045087        total: 1m 53s   remaining: 31.9s
390:    learn: 0.0044994        total: 1m 53s   remaining: 31.6s
391:    learn: 0.0044891        total: 1m 53s   remaining: 31.3s
392:    learn: 0.0044795        total: 1m 53s   remaining: 31s
393:    learn: 0.0044684        total: 1m 54s   remaining: 30.7s
394:    learn: 0.0044566        total: 1m 54s   remaining: 30.5s
395:    learn: 0.0044443        total: 1m 54s   remaining: 30.2s
396:    learn: 0.0044213        total: 1m 55s   remaining: 29.9s
397:    learn: 0.0043943        total: 1m 55s   remaining: 29.6s
398:    learn: 0.0043816        total: 1m 55s   remaining: 29.3s
399:    learn: 0.0043741        total: 1m 56s   remaining: 29s
400:    learn: 0.0043626        total: 1m 56s   remaining: 28.7s
401:    learn: 0.0043551        total: 1m 56s   remaining: 28.4s
402:    learn: 0.0043392        total: 1m 57s   remaining: 28.2s
403:    learn: 0.0043129        total: 1m 57s   remaining: 27.9s
404:    learn: 0.0042942        total: 1m 57s   remaining: 27.6s
405:    learn: 0.0042809        total: 1m 57s   remaining: 27.3s
406:    learn: 0.0042641        total: 1m 58s   remaining: 27s
407:    learn: 0.0042569        total: 1m 58s   remaining: 26.7s
408:    learn: 0.0042463        total: 1m 58s   remaining: 26.4s
409:    learn: 0.0042316        total: 1m 58s   remaining: 26.1s
410:    learn: 0.0042208        total: 1m 59s   remaining: 25.8s
411:    learn: 0.0042063        total: 1m 59s   remaining: 25.5s
412:    learn: 0.0041945        total: 1m 59s   remaining: 25.2s
413:    learn: 0.0041876        total: 2m       remaining: 24.9s
414:    learn: 0.0041811        total: 2m       remaining: 24.7s
415:    learn: 0.0041715        total: 2m       remaining: 24.4s
416:    learn: 0.0041621        total: 2m       remaining: 24.1s
417:    learn: 0.0041491        total: 2m 1s    remaining: 23.8s
418:    learn: 0.0041203        total: 2m 1s    remaining: 23.5s
419:    learn: 0.0041098        total: 2m 1s    remaining: 23.2s
420:    learn: 0.0040944        total: 2m 2s    remaining: 22.9s
421:    learn: 0.0040840        total: 2m 2s    remaining: 22.6s
422:    learn: 0.0040707        total: 2m 2s    remaining: 22.3s
423:    learn: 0.0040584        total: 2m 2s    remaining: 22s
424:    learn: 0.0040488        total: 2m 3s    remaining: 21.7s
425:    learn: 0.0040406        total: 2m 3s    remaining: 21.5s
426:    learn: 0.0040327        total: 2m 3s    remaining: 21.2s
427:    learn: 0.0040234        total: 2m 4s    remaining: 20.9s
428:    learn: 0.0040077        total: 2m 4s    remaining: 20.6s
429:    learn: 0.0039980        total: 2m 4s    remaining: 20.3s
430:    learn: 0.0039820        total: 2m 4s    remaining: 20s
431:    learn: 0.0039753        total: 2m 5s    remaining: 19.7s
432:    learn: 0.0039641        total: 2m 5s    remaining: 19.4s
```

```
433:    learn: 0.0039488        total: 2m 5s    remaining: 19.1s
434:    learn: 0.0039350        total: 2m 6s    remaining: 18.9s
435:    learn: 0.0039269        total: 2m 6s    remaining: 18.6s
436:    learn: 0.0039186        total: 2m 6s    remaining: 18.3s
437:    learn: 0.0039032        total: 2m 7s    remaining: 18s
438:    learn: 0.0038899        total: 2m 7s    remaining: 17.7s
439:    learn: 0.0038778        total: 2m 7s    remaining: 17.4s
440:    learn: 0.0038680        total: 2m 7s    remaining: 17.1s
441:    learn: 0.0038527        total: 2m 8s    remaining: 16.8s
442:    learn: 0.0038448        total: 2m 8s    remaining: 16.5s
443:    learn: 0.0038313        total: 2m 8s    remaining: 16.2s
444:    learn: 0.0038196        total: 2m 9s    remaining: 15.9s
445:    learn: 0.0038114        total: 2m 9s    remaining: 15.7s
446:    learn: 0.0038002        total: 2m 9s    remaining: 15.4s
447:    learn: 0.0037844        total: 2m 9s    remaining: 15.1s
448:    learn: 0.0037661        total: 2m 10s   remaining: 14.8s
449:    learn: 0.0037456        total: 2m 10s   remaining: 14.5s
450:    learn: 0.0037324        total: 2m 10s   remaining: 14.2s
451:    learn: 0.0037240        total: 2m 11s   remaining: 13.9s
452:    learn: 0.0037049        total: 2m 11s   remaining: 13.6s
453:    learn: 0.0036967        total: 2m 11s   remaining: 13.3s
454:    learn: 0.0036870        total: 2m 11s   remaining: 13.1s
455:    learn: 0.0036741        total: 2m 12s   remaining: 12.8s
456:    learn: 0.0036582        total: 2m 12s   remaining: 12.5s
457:    learn: 0.0036481        total: 2m 12s   remaining: 12.2s
458:    learn: 0.0036388        total: 2m 13s   remaining: 11.9s
459:    learn: 0.0036322        total: 2m 13s   remaining: 11.6s
460:    learn: 0.0036202        total: 2m 13s   remaining: 11.3s
461:    learn: 0.0036160        total: 2m 14s   remaining: 11s
462:    learn: 0.0036114        total: 2m 14s   remaining: 10.7s
463:    learn: 0.0035983        total: 2m 14s   remaining: 10.5s
464:    learn: 0.0035907        total: 2m 15s   remaining: 10.2s
465:    learn: 0.0035777        total: 2m 15s   remaining: 9.88s
466:    learn: 0.0035576        total: 2m 15s   remaining: 9.58s
467:    learn: 0.0035469        total: 2m 15s   remaining: 9.29s
468:    learn: 0.0035328        total: 2m 16s   remaining: 9s
469:    learn: 0.0035265        total: 2m 16s   remaining: 8.71s
470:    learn: 0.0035096        total: 2m 16s   remaining: 8.42s
471:    learn: 0.0035048        total: 2m 17s   remaining: 8.13s
472:    learn: 0.0034990        total: 2m 17s   remaining: 7.84s
473:    learn: 0.0034889        total: 2m 17s   remaining: 7.55s
474:    learn: 0.0034760        total: 2m 17s   remaining: 7.26s
475:    learn: 0.0034628        total: 2m 18s   remaining: 6.97s
476:    learn: 0.0034554        total: 2m 18s   remaining: 6.68s
477:    learn: 0.0034429        total: 2m 18s   remaining: 6.39s
478:    learn: 0.0034352        total: 2m 19s   remaining: 6.1s
479:    learn: 0.0034271        total: 2m 19s   remaining: 5.81s
480:    learn: 0.0034093        total: 2m 19s   remaining: 5.52s
```

```
481:	learn: 0.0033911	total: 2m 20s	remaining: 5.23s
482:	learn: 0.0033801	total: 2m 20s	remaining: 4.94s
483:	learn: 0.0033720	total: 2m 20s	remaining: 4.65s
484:	learn: 0.0033605	total: 2m 21s	remaining: 4.36s
485:	learn: 0.0033408	total: 2m 21s	remaining: 4.07s
486:	learn: 0.0033307	total: 2m 21s	remaining: 3.78s
487:	learn: 0.0033135	total: 2m 21s	remaining: 3.49s
488:	learn: 0.0033014	total: 2m 22s	remaining: 3.2s
489:	learn: 0.0032938	total: 2m 22s	remaining: 2.91s
490:	learn: 0.0032793	total: 2m 22s	remaining: 2.62s
491:	learn: 0.0032713	total: 2m 23s	remaining: 2.33s
492:	learn: 0.0032604	total: 2m 23s	remaining: 2.04s
493:	learn: 0.0032510	total: 2m 23s	remaining: 1.74s
494:	learn: 0.0032348	total: 2m 23s	remaining: 1.45s
495:	learn: 0.0032241	total: 2m 24s	remaining: 1.16s
496:	learn: 0.0032154	total: 2m 24s	remaining: 872ms
497:	learn: 0.0032069	total: 2m 24s	remaining: 582ms
498:	learn: 0.0032006	total: 2m 25s	remaining: 291ms
499:	learn: 0.0031890	total: 2m 25s	remaining: 0us
```

3.1.5 Perform cross-validation

We perform cross-validation to ensure our model is not over fitting.

This code takes around 11 minutes to run so it can be skipped. The conclusions are in the section interpretation below

[627]:

```
# Perform cross-validation - takes too long to run, was simply done for␣
 ↪validation purposes
cv_scores = cross_val_score(model, df_categorical, labels, cv=5,␣
 ↪scoring='accuracy')

# Print cross-validation scores
print("Cross-validation scores:", cv_scores)
print("Mean cross-validation score:", cv_scores.mean())

# Mean cross-validation score = 0.9999
```

```
0:	learn: 0.9609453	total: 131ms	remaining: 1m 5s
1:	learn: 0.8538477	total: 241ms	remaining: 1m
2:	learn: 0.7604131	total: 343ms	remaining: 56.7s
3:	learn: 0.6830851	total: 458ms	remaining: 56.8s
4:	learn: 0.6205445	total: 562ms	remaining: 55.6s
5:	learn: 0.5667827	total: 691ms	remaining: 56.9s
6:	learn: 0.5175913	total: 824ms	remaining: 58s
7:	learn: 0.4724244	total: 950ms	remaining: 58.4s
8:	learn: 0.4307875	total: 1.06s	remaining: 57.9s
9:	learn: 0.3965874	total: 1.18s	remaining: 57.9s
10:	learn: 0.3650461	total: 1.33s	remaining: 59.1s
```

```
11:     learn: 0.3344977        total: 1.45s    remaining: 59s
12:     learn: 0.3057596        total: 1.59s    remaining: 59.7s
13:     learn: 0.2812460        total: 1.74s    remaining: 1m
14:     learn: 0.2595487        total: 1.87s    remaining: 1m
15:     learn: 0.2408101        total: 2s       remaining: 1m
16:     learn: 0.2225384        total: 2.22s    remaining: 1m 3s
17:     learn: 0.2031752        total: 2.41s    remaining: 1m 4s
18:     learn: 0.1860500        total: 2.63s    remaining: 1m 6s
19:     learn: 0.1709324        total: 2.87s    remaining: 1m 8s
20:     learn: 0.1571399        total: 3.13s    remaining: 1m 11s
21:     learn: 0.1447375        total: 3.4s     remaining: 1m 13s
22:     learn: 0.1336399        total: 3.69s    remaining: 1m 16s
23:     learn: 0.1240606        total: 3.97s    remaining: 1m 18s
24:     learn: 0.1152412        total: 4.25s    remaining: 1m 20s
25:     learn: 0.1071751        total: 4.53s    remaining: 1m 22s
26:     learn: 0.1000305        total: 4.79s    remaining: 1m 23s
27:     learn: 0.0935409        total: 5.08s    remaining: 1m 25s
28:     learn: 0.0875465        total: 5.37s    remaining: 1m 27s
29:     learn: 0.0821609        total: 5.64s    remaining: 1m 28s
30:     learn: 0.0772571        total: 5.94s    remaining: 1m 29s
31:     learn: 0.0726550        total: 6.19s    remaining: 1m 30s
32:     learn: 0.0685814        total: 6.5s     remaining: 1m 31s
33:     learn: 0.0653457        total: 6.77s    remaining: 1m 32s
34:     learn: 0.0621859        total: 7.01s    remaining: 1m 33s
35:     learn: 0.0590070        total: 7.33s    remaining: 1m 34s
36:     learn: 0.0561267        total: 7.6s     remaining: 1m 35s
37:     learn: 0.0535818        total: 7.91s    remaining: 1m 36s
38:     learn: 0.0510894        total: 8.18s    remaining: 1m 36s
39:     learn: 0.0490559        total: 8.49s    remaining: 1m 37s
40:     learn: 0.0471072        total: 8.79s    remaining: 1m 38s
41:     learn: 0.0452139        total: 9.1s     remaining: 1m 39s
42:     learn: 0.0439288        total: 9.39s    remaining: 1m 39s
43:     learn: 0.0418436        total: 9.67s    remaining: 1m 40s
44:     learn: 0.0404589        total: 9.94s    remaining: 1m 40s
45:     learn: 0.0389766        total: 10.2s    remaining: 1m 41s
46:     learn: 0.0375094        total: 10.5s    remaining: 1m 41s
47:     learn: 0.0363352        total: 10.8s    remaining: 1m 42s
48:     learn: 0.0352272        total: 11.1s    remaining: 1m 42s
49:     learn: 0.0340849        total: 11.4s    remaining: 1m 42s
50:     learn: 0.0332051        total: 11.7s    remaining: 1m 42s
51:     learn: 0.0326030        total: 12s      remaining: 1m 43s
52:     learn: 0.0314578        total: 12.3s    remaining: 1m 43s
53:     learn: 0.0305880        total: 12.6s    remaining: 1m 43s
54:     learn: 0.0295036        total: 12.8s    remaining: 1m 43s
55:     learn: 0.0287889        total: 13.2s    remaining: 1m 44s
56:     learn: 0.0280266        total: 13.5s    remaining: 1m 44s
57:     learn: 0.0273208        total: 13.7s    remaining: 1m 44s
58:     learn: 0.0268725        total: 14.1s    remaining: 1m 45s
```

```
59:     learn: 0.0261099        total: 14.4s    remaining: 1m 45s
60:     learn: 0.0257448        total: 14.6s    remaining: 1m 45s
61:     learn: 0.0251978        total: 14.9s    remaining: 1m 45s
62:     learn: 0.0247881        total: 15.2s    remaining: 1m 45s
63:     learn: 0.0242353        total: 15.5s    remaining: 1m 45s
64:     learn: 0.0239087        total: 15.7s    remaining: 1m 45s
65:     learn: 0.0237231        total: 16s      remaining: 1m 45s
66:     learn: 0.0232928        total: 16.3s    remaining: 1m 45s
67:     learn: 0.0231286        total: 16.6s    remaining: 1m 45s
68:     learn: 0.0226860        total: 16.9s    remaining: 1m 45s
69:     learn: 0.0224795        total: 17.2s    remaining: 1m 45s
70:     learn: 0.0222636        total: 17.5s    remaining: 1m 45s
71:     learn: 0.0218100        total: 17.8s    remaining: 1m 45s
72:     learn: 0.0213742        total: 18.1s    remaining: 1m 45s
73:     learn: 0.0211550        total: 18.3s    remaining: 1m 45s
74:     learn: 0.0208290        total: 18.6s    remaining: 1m 45s
75:     learn: 0.0205756        total: 18.9s    remaining: 1m 45s
76:     learn: 0.0201777        total: 19.2s    remaining: 1m 45s
77:     learn: 0.0200680        total: 19.5s    remaining: 1m 45s
78:     learn: 0.0196615        total: 19.8s    remaining: 1m 45s
79:     learn: 0.0194786        total: 20.1s    remaining: 1m 45s
80:     learn: 0.0191967        total: 20.4s    remaining: 1m 45s
81:     learn: 0.0189348        total: 20.6s    remaining: 1m 45s
82:     learn: 0.0188483        total: 20.9s    remaining: 1m 45s
83:     learn: 0.0186944        total: 21.2s    remaining: 1m 45s
84:     learn: 0.0185207        total: 21.5s    remaining: 1m 45s
85:     learn: 0.0183783        total: 21.8s    remaining: 1m 45s
86:     learn: 0.0181975        total: 22.1s    remaining: 1m 45s
87:     learn: 0.0180615        total: 22.4s    remaining: 1m 44s
88:     learn: 0.0179499        total: 22.7s    remaining: 1m 44s
89:     learn: 0.0178268        total: 23s      remaining: 1m 44s
90:     learn: 0.0177022        total: 23.2s    remaining: 1m 44s
91:     learn: 0.0176135        total: 23.5s    remaining: 1m 44s
92:     learn: 0.0174639        total: 23.8s    remaining: 1m 44s
93:     learn: 0.0172825        total: 24.1s    remaining: 1m 44s
94:     learn: 0.0171336        total: 24.4s    remaining: 1m 43s
95:     learn: 0.0169303        total: 24.7s    remaining: 1m 43s
96:     learn: 0.0167064        total: 25s      remaining: 1m 43s
97:     learn: 0.0165897        total: 25.3s    remaining: 1m 43s
98:     learn: 0.0164740        total: 25.6s    remaining: 1m 43s
99:     learn: 0.0164003        total: 25.9s    remaining: 1m 43s
100:    learn: 0.0163335        total: 26.1s    remaining: 1m 43s
101:    learn: 0.0162158        total: 26.4s    remaining: 1m 43s
102:    learn: 0.0160986        total: 26.7s    remaining: 1m 42s
103:    learn: 0.0159502        total: 27s      remaining: 1m 42s
104:    learn: 0.0157712        total: 27.3s    remaining: 1m 42s
105:    learn: 0.0156581        total: 27.6s    remaining: 1m 42s
106:    learn: 0.0154494        total: 27.9s    remaining: 1m 42s
```

```
107:    learn: 0.0153559        total: 28.1s    remaining: 1m 42s
108:    learn: 0.0152560        total: 28.4s    remaining: 1m 41s
109:    learn: 0.0151515        total: 28.8s    remaining: 1m 42s
110:    learn: 0.0150600        total: 29.1s    remaining: 1m 41s
111:    learn: 0.0149271        total: 29.4s    remaining: 1m 41s
112:    learn: 0.0147990        total: 29.7s    remaining: 1m 41s
113:    learn: 0.0147010        total: 30.1s    remaining: 1m 41s
114:    learn: 0.0146126        total: 30.4s    remaining: 1m 41s
115:    learn: 0.0144623        total: 30.7s    remaining: 1m 41s
116:    learn: 0.0143500        total: 31s      remaining: 1m 41s
117:    learn: 0.0142091        total: 31.3s    remaining: 1m 41s
118:    learn: 0.0140773        total: 31.5s    remaining: 1m 40s
119:    learn: 0.0139603        total: 31.8s    remaining: 1m 40s
120:    learn: 0.0138904        total: 32.1s    remaining: 1m 40s
121:    learn: 0.0138161        total: 32.4s    remaining: 1m 40s
122:    learn: 0.0137667        total: 32.7s    remaining: 1m 40s
123:    learn: 0.0136847        total: 32.9s    remaining: 1m 39s
124:    learn: 0.0135671        total: 33.3s    remaining: 1m 39s
125:    learn: 0.0134560        total: 33.6s    remaining: 1m 39s
126:    learn: 0.0134217        total: 33.8s    remaining: 1m 39s
127:    learn: 0.0133648        total: 34.1s    remaining: 1m 39s
128:    learn: 0.0133250        total: 34.5s    remaining: 1m 39s
129:    learn: 0.0132569        total: 34.7s    remaining: 1m 38s
130:    learn: 0.0131313        total: 35s      remaining: 1m 38s
131:    learn: 0.0129992        total: 35.3s    remaining: 1m 38s
132:    learn: 0.0129246        total: 35.6s    remaining: 1m 38s
133:    learn: 0.0129017        total: 35.9s    remaining: 1m 38s
134:    learn: 0.0128534        total: 36.2s    remaining: 1m 37s
135:    learn: 0.0127936        total: 36.4s    remaining: 1m 37s
136:    learn: 0.0127265        total: 36.7s    remaining: 1m 37s
137:    learn: 0.0126841        total: 37s      remaining: 1m 37s
138:    learn: 0.0125752        total: 37.3s    remaining: 1m 36s
139:    learn: 0.0125086        total: 37.6s    remaining: 1m 36s
140:    learn: 0.0124463        total: 37.8s    remaining: 1m 36s
141:    learn: 0.0123593        total: 38.1s    remaining: 1m 36s
142:    learn: 0.0122736        total: 38.4s    remaining: 1m 35s
143:    learn: 0.0122134        total: 38.6s    remaining: 1m 35s
144:    learn: 0.0121840        total: 38.9s    remaining: 1m 35s
145:    learn: 0.0121152        total: 39.2s    remaining: 1m 35s
146:    learn: 0.0120705        total: 39.5s    remaining: 1m 34s
147:    learn: 0.0120278        total: 39.8s    remaining: 1m 34s
148:    learn: 0.0119346        total: 40s      remaining: 1m 34s
149:    learn: 0.0118368        total: 40.3s    remaining: 1m 34s
150:    learn: 0.0117717        total: 40.6s    remaining: 1m 33s
151:    learn: 0.0116915        total: 40.9s    remaining: 1m 33s
152:    learn: 0.0116397        total: 41.2s    remaining: 1m 33s
153:    learn: 0.0115458        total: 41.5s    remaining: 1m 33s
154:    learn: 0.0114647        total: 41.8s    remaining: 1m 33s
```

```
155:    learn: 0.0113753        total: 42.1s    remaining: 1m 32s
156:    learn: 0.0113371        total: 42.4s    remaining: 1m 32s
157:    learn: 0.0112436        total: 42.6s    remaining: 1m 32s
158:    learn: 0.0112099        total: 42.9s    remaining: 1m 32s
159:    learn: 0.0111674        total: 43.2s    remaining: 1m 31s
160:    learn: 0.0110871        total: 43.5s    remaining: 1m 31s
161:    learn: 0.0110580        total: 43.8s    remaining: 1m 31s
162:    learn: 0.0109869        total: 44.1s    remaining: 1m 31s
163:    learn: 0.0109176        total: 44.4s    remaining: 1m 30s
164:    learn: 0.0108530        total: 44.7s    remaining: 1m 30s
165:    learn: 0.0108089        total: 45s      remaining: 1m 30s
166:    learn: 0.0107861        total: 45.3s    remaining: 1m 30s
167:    learn: 0.0107415        total: 45.5s    remaining: 1m 30s
168:    learn: 0.0106849        total: 45.8s    remaining: 1m 29s
169:    learn: 0.0106155        total: 46.1s    remaining: 1m 29s
170:    learn: 0.0105445        total: 46.4s    remaining: 1m 29s
171:    learn: 0.0104517        total: 46.6s    remaining: 1m 28s
172:    learn: 0.0103617        total: 47s      remaining: 1m 28s
173:    learn: 0.0103139        total: 47.3s    remaining: 1m 28s
174:    learn: 0.0102194        total: 47.6s    remaining: 1m 28s
175:    learn: 0.0101473        total: 47.8s    remaining: 1m 28s
176:    learn: 0.0101060        total: 48.1s    remaining: 1m 27s
177:    learn: 0.0100651        total: 48.4s    remaining: 1m 27s
178:    learn: 0.0100091        total: 48.7s    remaining: 1m 27s
179:    learn: 0.0099498        total: 48.9s    remaining: 1m 27s
180:    learn: 0.0099186        total: 49.2s    remaining: 1m 26s
181:    learn: 0.0098871        total: 49.5s    remaining: 1m 26s
182:    learn: 0.0098620        total: 49.8s    remaining: 1m 26s
183:    learn: 0.0097948        total: 50.1s    remaining: 1m 26s
184:    learn: 0.0097658        total: 50.4s    remaining: 1m 25s
185:    learn: 0.0096880        total: 50.7s    remaining: 1m 25s
186:    learn: 0.0096521        total: 51s      remaining: 1m 25s
187:    learn: 0.0095733        total: 51.3s    remaining: 1m 25s
188:    learn: 0.0095117        total: 51.6s    remaining: 1m 24s
189:    learn: 0.0094639        total: 51.9s    remaining: 1m 24s
190:    learn: 0.0093708        total: 52.2s    remaining: 1m 24s
191:    learn: 0.0093155        total: 52.4s    remaining: 1m 24s
192:    learn: 0.0092696        total: 52.8s    remaining: 1m 23s
193:    learn: 0.0092574        total: 53s      remaining: 1m 23s
194:    learn: 0.0091945        total: 53.3s    remaining: 1m 23s
195:    learn: 0.0091364        total: 53.6s    remaining: 1m 23s
196:    learn: 0.0090968        total: 53.8s    remaining: 1m 22s
197:    learn: 0.0090301        total: 54.1s    remaining: 1m 22s
198:    learn: 0.0089720        total: 54.4s    remaining: 1m 22s
199:    learn: 0.0089336        total: 54.7s    remaining: 1m 22s
200:    learn: 0.0089052        total: 55s      remaining: 1m 21s
201:    learn: 0.0088623        total: 55.3s    remaining: 1m 21s
202:    learn: 0.0088125        total: 55.6s    remaining: 1m 21s
```

```
203:    learn: 0.0087697        total: 55.9s    remaining: 1m 21s
204:    learn: 0.0087318        total: 56.1s    remaining: 1m 20s
205:    learn: 0.0086567        total: 56.4s    remaining: 1m 20s
206:    learn: 0.0086291        total: 56.6s    remaining: 1m 20s
207:    learn: 0.0085912        total: 56.9s    remaining: 1m 19s
208:    learn: 0.0085507        total: 57.2s    remaining: 1m 19s
209:    learn: 0.0084916        total: 57.4s    remaining: 1m 19s
210:    learn: 0.0084322        total: 57.7s    remaining: 1m 19s
211:    learn: 0.0084045        total: 58s      remaining: 1m 18s
212:    learn: 0.0083603        total: 58.3s    remaining: 1m 18s
213:    learn: 0.0083151        total: 58.5s    remaining: 1m 18s
214:    learn: 0.0082608        total: 58.8s    remaining: 1m 18s
215:    learn: 0.0082181        total: 59.2s    remaining: 1m 17s
216:    learn: 0.0081748        total: 59.4s    remaining: 1m 17s
217:    learn: 0.0081464        total: 59.7s    remaining: 1m 17s
218:    learn: 0.0081021        total: 60s      remaining: 1m 16s
219:    learn: 0.0080643        total: 1m       remaining: 1m 16s
220:    learn: 0.0080176        total: 1m       remaining: 1m 16s
221:    learn: 0.0079698        total: 1m       remaining: 1m 16s
222:    learn: 0.0079227        total: 1m 1s    remaining: 1m 15s
223:    learn: 0.0078901        total: 1m 1s    remaining: 1m 15s
224:    learn: 0.0078471        total: 1m 1s    remaining: 1m 15s
225:    learn: 0.0078331        total: 1m 2s    remaining: 1m 15s
226:    learn: 0.0077765        total: 1m 2s    remaining: 1m 14s
227:    learn: 0.0077309        total: 1m 2s    remaining: 1m 14s
228:    learn: 0.0076987        total: 1m 2s    remaining: 1m 14s
229:    learn: 0.0076563        total: 1m 3s    remaining: 1m 14s
230:    learn: 0.0076224        total: 1m 3s    remaining: 1m 13s
231:    learn: 0.0076011        total: 1m 3s    remaining: 1m 13s
232:    learn: 0.0075280        total: 1m 3s    remaining: 1m 13s
233:    learn: 0.0075160        total: 1m 4s    remaining: 1m 13s
234:    learn: 0.0074791        total: 1m 4s    remaining: 1m 12s
235:    learn: 0.0074719        total: 1m 4s    remaining: 1m 12s
236:    learn: 0.0074436        total: 1m 5s    remaining: 1m 12s
237:    learn: 0.0073982        total: 1m 5s    remaining: 1m 11s
238:    learn: 0.0073546        total: 1m 5s    remaining: 1m 11s
239:    learn: 0.0073263        total: 1m 5s    remaining: 1m 11s
240:    learn: 0.0072995        total: 1m 6s    remaining: 1m 11s
241:    learn: 0.0072835        total: 1m 6s    remaining: 1m 10s
242:    learn: 0.0072618        total: 1m 6s    remaining: 1m 10s
243:    learn: 0.0072307        total: 1m 7s    remaining: 1m 10s
244:    learn: 0.0071805        total: 1m 7s    remaining: 1m 10s
245:    learn: 0.0071563        total: 1m 7s    remaining: 1m 10s
246:    learn: 0.0071230        total: 1m 8s    remaining: 1m 9s
247:    learn: 0.0071172        total: 1m 8s    remaining: 1m 9s
248:    learn: 0.0070704        total: 1m 8s    remaining: 1m 9s
249:    learn: 0.0070404        total: 1m 8s    remaining: 1m 8s
250:    learn: 0.0069956        total: 1m 9s    remaining: 1m 8s
```

```
251:    learn: 0.0069643        total: 1m 9s    remaining: 1m 8s
252:    learn: 0.0069388        total: 1m 9s    remaining: 1m 8s
253:    learn: 0.0069066        total: 1m 10s   remaining: 1m 7s
254:    learn: 0.0068948        total: 1m 10s   remaining: 1m 7s
255:    learn: 0.0068658        total: 1m 10s   remaining: 1m 7s
256:    learn: 0.0068330        total: 1m 10s   remaining: 1m 7s
257:    learn: 0.0068051        total: 1m 11s   remaining: 1m 6s
258:    learn: 0.0067683        total: 1m 11s   remaining: 1m 6s
259:    learn: 0.0067490        total: 1m 11s   remaining: 1m 6s
260:    learn: 0.0067252        total: 1m 12s   remaining: 1m 5s
261:    learn: 0.0066907        total: 1m 12s   remaining: 1m 5s
262:    learn: 0.0066768        total: 1m 12s   remaining: 1m 5s
263:    learn: 0.0066375        total: 1m 12s   remaining: 1m 5s
264:    learn: 0.0065868        total: 1m 13s   remaining: 1m 4s
265:    learn: 0.0065524        total: 1m 13s   remaining: 1m 4s
266:    learn: 0.0065230        total: 1m 13s   remaining: 1m 4s
267:    learn: 0.0065093        total: 1m 14s   remaining: 1m 4s
268:    learn: 0.0064756        total: 1m 14s   remaining: 1m 3s
269:    learn: 0.0064553        total: 1m 14s   remaining: 1m 3s
270:    learn: 0.0064361        total: 1m 14s   remaining: 1m 3s
271:    learn: 0.0064137        total: 1m 15s   remaining: 1m 3s
272:    learn: 0.0063820        total: 1m 15s   remaining: 1m 2s
273:    learn: 0.0063545        total: 1m 15s   remaining: 1m 2s
274:    learn: 0.0063347        total: 1m 16s   remaining: 1m 2s
275:    learn: 0.0063011        total: 1m 16s   remaining: 1m 2s
276:    learn: 0.0062805        total: 1m 16s   remaining: 1m 1s
277:    learn: 0.0062659        total: 1m 17s   remaining: 1m 1s
278:    learn: 0.0062542        total: 1m 17s   remaining: 1m 1s
279:    learn: 0.0062258        total: 1m 17s   remaining: 1m
280:    learn: 0.0061949        total: 1m 17s   remaining: 1m
281:    learn: 0.0061791        total: 1m 18s   remaining: 1m
282:    learn: 0.0061628        total: 1m 18s   remaining: 1m
283:    learn: 0.0061358        total: 1m 18s   remaining: 59.9s
284:    learn: 0.0061067        total: 1m 19s   remaining: 59.7s
285:    learn: 0.0060860        total: 1m 19s   remaining: 59.4s
286:    learn: 0.0060613        total: 1m 19s   remaining: 59.1s
287:    learn: 0.0060310        total: 1m 19s   remaining: 58.9s
288:    learn: 0.0059899        total: 1m 20s   remaining: 58.6s
289:    learn: 0.0059669        total: 1m 20s   remaining: 58.3s
290:    learn: 0.0059604        total: 1m 20s   remaining: 58.1s
291:    learn: 0.0059305        total: 1m 21s   remaining: 57.8s
292:    learn: 0.0059099        total: 1m 21s   remaining: 57.5s
293:    learn: 0.0058894        total: 1m 21s   remaining: 57.3s
294:    learn: 0.0058397        total: 1m 22s   remaining: 57s
295:    learn: 0.0058275        total: 1m 22s   remaining: 56.7s
296:    learn: 0.0058105        total: 1m 22s   remaining: 56.5s
297:    learn: 0.0057765        total: 1m 22s   remaining: 56.2s
298:    learn: 0.0057637        total: 1m 23s   remaining: 55.9s
```

```
299:    learn: 0.0057342        total: 1m 23s   remaining: 55.6s
300:    learn: 0.0057075        total: 1m 23s   remaining: 55.4s
301:    learn: 0.0056842        total: 1m 24s   remaining: 55.1s
302:    learn: 0.0056633        total: 1m 24s   remaining: 54.9s
303:    learn: 0.0056402        total: 1m 24s   remaining: 54.6s
304:    learn: 0.0056146        total: 1m 24s   remaining: 54.3s
305:    learn: 0.0055715        total: 1m 25s   remaining: 54.1s
306:    learn: 0.0055457        total: 1m 25s   remaining: 53.8s
307:    learn: 0.0055317        total: 1m 25s   remaining: 53.6s
308:    learn: 0.0055167        total: 1m 26s   remaining: 53.4s
309:    learn: 0.0054917        total: 1m 26s   remaining: 53.1s
310:    learn: 0.0054800        total: 1m 27s   remaining: 52.9s
311:    learn: 0.0054728        total: 1m 27s   remaining: 52.7s
312:    learn: 0.0054623        total: 1m 27s   remaining: 52.4s
313:    learn: 0.0054335        total: 1m 27s   remaining: 52.1s
314:    learn: 0.0054209        total: 1m 28s   remaining: 51.9s
315:    learn: 0.0053988        total: 1m 28s   remaining: 51.6s
316:    learn: 0.0053955        total: 1m 28s   remaining: 51.2s
317:    learn: 0.0053714        total: 1m 29s   remaining: 51s
318:    learn: 0.0053673        total: 1m 29s   remaining: 50.7s
319:    learn: 0.0053457        total: 1m 29s   remaining: 50.5s
320:    learn: 0.0053079        total: 1m 30s   remaining: 50.2s
321:    learn: 0.0052824        total: 1m 30s   remaining: 49.9s
322:    learn: 0.0052683        total: 1m 30s   remaining: 49.7s
323:    learn: 0.0052576        total: 1m 30s   remaining: 49.4s
324:    learn: 0.0052254        total: 1m 31s   remaining: 49.1s
325:    learn: 0.0052120        total: 1m 31s   remaining: 48.9s
326:    learn: 0.0051905        total: 1m 31s   remaining: 48.6s
327:    learn: 0.0051699        total: 1m 32s   remaining: 48.3s
328:    learn: 0.0051559        total: 1m 32s   remaining: 48.1s
329:    learn: 0.0051314        total: 1m 32s   remaining: 47.8s
330:    learn: 0.0051123        total: 1m 33s   remaining: 47.5s
331:    learn: 0.0050881        total: 1m 33s   remaining: 47.3s
332:    learn: 0.0050642        total: 1m 33s   remaining: 47s
333:    learn: 0.0050541        total: 1m 33s   remaining: 46.7s
334:    learn: 0.0050348        total: 1m 34s   remaining: 46.4s
335:    learn: 0.0050283        total: 1m 34s   remaining: 46.2s
336:    learn: 0.0050180        total: 1m 34s   remaining: 45.9s
337:    learn: 0.0050008        total: 1m 35s   remaining: 45.6s
338:    learn: 0.0049879        total: 1m 35s   remaining: 45.4s
339:    learn: 0.0049654        total: 1m 35s   remaining: 45.1s
340:    learn: 0.0049478        total: 1m 36s   remaining: 44.8s
341:    learn: 0.0049302        total: 1m 36s   remaining: 44.5s
342:    learn: 0.0049212        total: 1m 36s   remaining: 44.3s
343:    learn: 0.0049037        total: 1m 36s   remaining: 44s
344:    learn: 0.0048851        total: 1m 37s   remaining: 43.7s
345:    learn: 0.0048730        total: 1m 37s   remaining: 43.4s
346:    learn: 0.0048593        total: 1m 37s   remaining: 43.2s
```

```
347:    learn: 0.0048385        total: 1m 38s   remaining: 42.9s
348:    learn: 0.0048157        total: 1m 38s   remaining: 42.6s
349:    learn: 0.0047994        total: 1m 38s   remaining: 42.3s
350:    learn: 0.0047754        total: 1m 39s   remaining: 42.1s
351:    learn: 0.0047537        total: 1m 39s   remaining: 41.8s
352:    learn: 0.0047447        total: 1m 39s   remaining: 41.5s
353:    learn: 0.0047142        total: 1m 40s   remaining: 41.2s
354:    learn: 0.0046986        total: 1m 40s   remaining: 41s
355:    learn: 0.0046747        total: 1m 40s   remaining: 40.7s
356:    learn: 0.0046621        total: 1m 40s   remaining: 40.4s
357:    learn: 0.0046403        total: 1m 41s   remaining: 40.2s
358:    learn: 0.0046286        total: 1m 41s   remaining: 39.9s
359:    learn: 0.0046135        total: 1m 41s   remaining: 39.6s
360:    learn: 0.0045937        total: 1m 42s   remaining: 39.4s
361:    learn: 0.0045725        total: 1m 42s   remaining: 39.1s
362:    learn: 0.0045578        total: 1m 42s   remaining: 38.8s
363:    learn: 0.0045397        total: 1m 43s   remaining: 38.5s
364:    learn: 0.0045281        total: 1m 43s   remaining: 38.3s
365:    learn: 0.0045162        total: 1m 43s   remaining: 38s
366:    learn: 0.0044987        total: 1m 44s   remaining: 37.7s
367:    learn: 0.0044772        total: 1m 44s   remaining: 37.4s
368:    learn: 0.0044600        total: 1m 44s   remaining: 37.1s
369:    learn: 0.0044502        total: 1m 44s   remaining: 36.9s
370:    learn: 0.0044365        total: 1m 45s   remaining: 36.6s
371:    learn: 0.0044148        total: 1m 45s   remaining: 36.3s
372:    learn: 0.0043938        total: 1m 45s   remaining: 36s
373:    learn: 0.0043831        total: 1m 46s   remaining: 35.8s
374:    learn: 0.0043647        total: 1m 46s   remaining: 35.5s
375:    learn: 0.0043440        total: 1m 46s   remaining: 35.2s
376:    learn: 0.0043343        total: 1m 47s   remaining: 34.9s
377:    learn: 0.0043216        total: 1m 47s   remaining: 34.7s
378:    learn: 0.0043095        total: 1m 47s   remaining: 34.4s
379:    learn: 0.0042888        total: 1m 48s   remaining: 34.1s
380:    learn: 0.0042772        total: 1m 48s   remaining: 33.9s
381:    learn: 0.0042717        total: 1m 48s   remaining: 33.6s
382:    learn: 0.0042675        total: 1m 49s   remaining: 33.3s
383:    learn: 0.0042490        total: 1m 49s   remaining: 33s
384:    learn: 0.0042306        total: 1m 49s   remaining: 32.8s
385:    learn: 0.0042148        total: 1m 49s   remaining: 32.5s
386:    learn: 0.0042093        total: 1m 50s   remaining: 32.2s
387:    learn: 0.0042029        total: 1m 50s   remaining: 31.9s
388:    learn: 0.0041977        total: 1m 50s   remaining: 31.6s
389:    learn: 0.0041779        total: 1m 51s   remaining: 31.3s
390:    learn: 0.0041632        total: 1m 51s   remaining: 31.1s
391:    learn: 0.0041487        total: 1m 51s   remaining: 30.8s
392:    learn: 0.0041315        total: 1m 52s   remaining: 30.5s
393:    learn: 0.0041160        total: 1m 52s   remaining: 30.2s
394:    learn: 0.0041035        total: 1m 52s   remaining: 29.9s
```

```
395:    learn: 0.0040950        total: 1m 52s   remaining: 29.6s
396:    learn: 0.0040789        total: 1m 53s   remaining: 29.4s
397:    learn: 0.0040652        total: 1m 53s   remaining: 29.1s
398:    learn: 0.0040479        total: 1m 53s   remaining: 28.8s
399:    learn: 0.0040400        total: 1m 54s   remaining: 28.5s
400:    learn: 0.0040326        total: 1m 54s   remaining: 28.2s
401:    learn: 0.0040239        total: 1m 54s   remaining: 28s
402:    learn: 0.0040078        total: 1m 54s   remaining: 27.7s
403:    learn: 0.0039898        total: 1m 55s   remaining: 27.4s
404:    learn: 0.0039784        total: 1m 55s   remaining: 27.1s
405:    learn: 0.0039658        total: 1m 55s   remaining: 26.8s
406:    learn: 0.0039460        total: 1m 56s   remaining: 26.5s
407:    learn: 0.0039331        total: 1m 56s   remaining: 26.3s
408:    learn: 0.0039223        total: 1m 56s   remaining: 26s
409:    learn: 0.0039031        total: 1m 57s   remaining: 25.7s
410:    learn: 0.0038856        total: 1m 57s   remaining: 25.4s
411:    learn: 0.0038656        total: 1m 57s   remaining: 25.1s
412:    learn: 0.0038548        total: 1m 57s   remaining: 24.8s
413:    learn: 0.0038468        total: 1m 58s   remaining: 24.5s
414:    learn: 0.0038319        total: 1m 58s   remaining: 24.3s
415:    learn: 0.0038182        total: 1m 58s   remaining: 24s
416:    learn: 0.0038072        total: 1m 58s   remaining: 23.7s
417:    learn: 0.0037945        total: 1m 59s   remaining: 23.4s
418:    learn: 0.0037879        total: 1m 59s   remaining: 23.1s
419:    learn: 0.0037735        total: 1m 59s   remaining: 22.8s
420:    learn: 0.0037668        total: 2m       remaining: 22.5s
421:    learn: 0.0037576        total: 2m       remaining: 22.3s
422:    learn: 0.0037373        total: 2m       remaining: 22s
423:    learn: 0.0037292        total: 2m 1s    remaining: 21.7s
424:    learn: 0.0037149        total: 2m 1s    remaining: 21.4s
425:    learn: 0.0037004        total: 2m 1s    remaining: 21.1s
426:    learn: 0.0036910        total: 2m 1s    remaining: 20.8s
427:    learn: 0.0036825        total: 2m 2s    remaining: 20.6s
428:    learn: 0.0036656        total: 2m 2s    remaining: 20.3s
429:    learn: 0.0036601        total: 2m 2s    remaining: 20s
430:    learn: 0.0036509        total: 2m 3s    remaining: 19.7s
431:    learn: 0.0036380        total: 2m 3s    remaining: 19.4s
432:    learn: 0.0036232        total: 2m 3s    remaining: 19.2s
433:    learn: 0.0036185        total: 2m 4s    remaining: 18.9s
434:    learn: 0.0036086        total: 2m 4s    remaining: 18.6s
435:    learn: 0.0035981        total: 2m 4s    remaining: 18.3s
436:    learn: 0.0035834        total: 2m 5s    remaining: 18s
437:    learn: 0.0035673        total: 2m 5s    remaining: 17.7s
438:    learn: 0.0035510        total: 2m 5s    remaining: 17.5s
439:    learn: 0.0035391        total: 2m 5s    remaining: 17.2s
440:    learn: 0.0035225        total: 2m 6s    remaining: 16.9s
441:    learn: 0.0035131        total: 2m 6s    remaining: 16.6s
442:    learn: 0.0035054        total: 2m 6s    remaining: 16.3s
```

```
443:    learn: 0.0034981        total: 2m 7s    remaining: 16s
444:    learn: 0.0034866        total: 2m 7s    remaining: 15.7s
445:    learn: 0.0034752        total: 2m 7s    remaining: 15.4s
446:    learn: 0.0034659        total: 2m 7s    remaining: 15.2s
447:    learn: 0.0034596        total: 2m 8s    remaining: 14.9s
448:    learn: 0.0034465        total: 2m 8s    remaining: 14.6s
449:    learn: 0.0034395        total: 2m 8s    remaining: 14.3s
450:    learn: 0.0034246        total: 2m 9s    remaining: 14s
451:    learn: 0.0034142        total: 2m 9s    remaining: 13.7s
452:    learn: 0.0034028        total: 2m 9s    remaining: 13.5s
453:    learn: 0.0033892        total: 2m 10s   remaining: 13.2s
454:    learn: 0.0033798        total: 2m 10s   remaining: 12.9s
455:    learn: 0.0033630        total: 2m 10s   remaining: 12.6s
456:    learn: 0.0033514        total: 2m 10s   remaining: 12.3s
457:    learn: 0.0033470        total: 2m 11s   remaining: 12s
458:    learn: 0.0033326        total: 2m 11s   remaining: 11.7s
459:    learn: 0.0033221        total: 2m 11s   remaining: 11.5s
460:    learn: 0.0033151        total: 2m 12s   remaining: 11.2s
461:    learn: 0.0033039        total: 2m 12s   remaining: 10.9s
462:    learn: 0.0032958        total: 2m 12s   remaining: 10.6s
463:    learn: 0.0032807        total: 2m 12s   remaining: 10.3s
464:    learn: 0.0032767        total: 2m 13s   remaining: 10s
465:    learn: 0.0032679        total: 2m 13s   remaining: 9.74s
466:    learn: 0.0032627        total: 2m 13s   remaining: 9.45s
467:    learn: 0.0032501        total: 2m 14s   remaining: 9.16s
468:    learn: 0.0032437        total: 2m 14s   remaining: 8.88s
469:    learn: 0.0032328        total: 2m 14s   remaining: 8.59s
470:    learn: 0.0032266        total: 2m 14s   remaining: 8.3s
471:    learn: 0.0032195        total: 2m 15s   remaining: 8.02s
472:    learn: 0.0031908        total: 2m 15s   remaining: 7.73s
473:    learn: 0.0031856        total: 2m 15s   remaining: 7.45s
474:    learn: 0.0031769        total: 2m 16s   remaining: 7.16s
475:    learn: 0.0031660        total: 2m 16s   remaining: 6.87s
476:    learn: 0.0031557        total: 2m 16s   remaining: 6.59s
477:    learn: 0.0031463        total: 2m 16s   remaining: 6.3s
478:    learn: 0.0031390        total: 2m 17s   remaining: 6.01s
479:    learn: 0.0031321        total: 2m 17s   remaining: 5.73s
480:    learn: 0.0031229        total: 2m 17s   remaining: 5.44s
481:    learn: 0.0031145        total: 2m 18s   remaining: 5.15s
482:    learn: 0.0031080        total: 2m 18s   remaining: 4.87s
483:    learn: 0.0030976        total: 2m 18s   remaining: 4.58s
484:    learn: 0.0030849        total: 2m 18s   remaining: 4.29s
485:    learn: 0.0030791        total: 2m 19s   remaining: 4.01s
486:    learn: 0.0030724        total: 2m 19s   remaining: 3.72s
487:    learn: 0.0030590        total: 2m 19s   remaining: 3.44s
488:    learn: 0.0030474        total: 2m 20s   remaining: 3.15s
489:    learn: 0.0030354        total: 2m 20s   remaining: 2.86s
490:    learn: 0.0030203        total: 2m 20s   remaining: 2.58s
```

```
491:    learn: 0.0030096        total: 2m 20s   remaining: 2.29s
492:    learn: 0.0030000        total: 2m 21s   remaining: 2s
493:    learn: 0.0029946        total: 2m 21s   remaining: 1.72s
494:    learn: 0.0029909        total: 2m 21s   remaining: 1.43s
495:    learn: 0.0029833        total: 2m 21s   remaining: 1.14s
496:    learn: 0.0029727        total: 2m 22s   remaining: 859ms
497:    learn: 0.0029670        total: 2m 22s   remaining: 572ms
498:    learn: 0.0029640        total: 2m 22s   remaining: 286ms
499:    learn: 0.0029574        total: 2m 23s   remaining: 0us
0:      learn: 0.9568851        total: 194ms    remaining: 1m 36s
1:      learn: 0.8451088        total: 491ms    remaining: 2m 2s
2:      learn: 0.7522285        total: 773ms    remaining: 2m 7s
3:      learn: 0.6793363        total: 1.05s    remaining: 2m 10s
4:      learn: 0.6141690        total: 1.36s    remaining: 2m 14s
5:      learn: 0.5510845        total: 1.71s    remaining: 2m 21s
6:      learn: 0.4970896        total: 2s       remaining: 2m 21s
7:      learn: 0.4491691        total: 2.3s     remaining: 2m 21s
8:      learn: 0.4093076        total: 2.7s     remaining: 2m 27s
9:      learn: 0.3734833        total: 2.99s    remaining: 2m 26s
10:     learn: 0.3444579        total: 3.29s    remaining: 2m 26s
11:     learn: 0.3163233        total: 3.58s    remaining: 2m 25s
12:     learn: 0.2915401        total: 3.87s    remaining: 2m 25s
13:     learn: 0.2686471        total: 4.16s    remaining: 2m 24s
14:     learn: 0.2478632        total: 4.43s    remaining: 2m 23s
15:     learn: 0.2283621        total: 4.7s     remaining: 2m 22s
16:     learn: 0.2118629        total: 4.96s    remaining: 2m 21s
17:     learn: 0.1958613        total: 5.23s    remaining: 2m 20s
18:     learn: 0.1818092        total: 5.56s    remaining: 2m 20s
19:     learn: 0.1669039        total: 5.82s    remaining: 2m 19s
20:     learn: 0.1541048        total: 6.09s    remaining: 2m 18s
21:     learn: 0.1419022        total: 6.35s    remaining: 2m 17s
22:     learn: 0.1310431        total: 6.6s     remaining: 2m 16s
23:     learn: 0.1212794        total: 6.84s    remaining: 2m 15s
24:     learn: 0.1125062        total: 7.09s    remaining: 2m 14s
25:     learn: 0.1052124        total: 7.33s    remaining: 2m 13s
26:     learn: 0.0987734        total: 7.62s    remaining: 2m 13s
27:     learn: 0.0921991        total: 7.88s    remaining: 2m 12s
28:     learn: 0.0862879        total: 8.15s    remaining: 2m 12s
29:     learn: 0.0808369        total: 8.41s    remaining: 2m 11s
30:     learn: 0.0758405        total: 8.71s    remaining: 2m 11s
31:     learn: 0.0712514        total: 8.95s    remaining: 2m 10s
32:     learn: 0.0672898        total: 9.26s    remaining: 2m 11s
33:     learn: 0.0639139        total: 9.52s    remaining: 2m 10s
34:     learn: 0.0606475        total: 9.82s    remaining: 2m 10s
35:     learn: 0.0574656        total: 10.1s    remaining: 2m 10s
36:     learn: 0.0544720        total: 10.4s    remaining: 2m 10s
37:     learn: 0.0516065        total: 10.7s    remaining: 2m 10s
38:     learn: 0.0492605        total: 11s      remaining: 2m 10s
```

```
39:     learn: 0.0469662        total: 11.3s    remaining: 2m 9s
40:     learn: 0.0450416        total: 11.6s    remaining: 2m 9s
41:     learn: 0.0429959        total: 11.9s    remaining: 2m 9s
42:     learn: 0.0414693        total: 12.2s    remaining: 2m 9s
43:     learn: 0.0400889        total: 12.5s    remaining: 2m 9s
44:     learn: 0.0387400        total: 12.8s    remaining: 2m 9s
45:     learn: 0.0373167        total: 13.1s    remaining: 2m 8s
46:     learn: 0.0357574        total: 13.4s    remaining: 2m 8s
47:     learn: 0.0345084        total: 13.7s    remaining: 2m 8s
48:     learn: 0.0333169        total: 14s      remaining: 2m 8s
49:     learn: 0.0326050        total: 14.2s    remaining: 2m 8s
50:     learn: 0.0317093        total: 14.5s    remaining: 2m 7s
51:     learn: 0.0309205        total: 14.8s    remaining: 2m 7s
52:     learn: 0.0300616        total: 15.1s    remaining: 2m 7s
53:     learn: 0.0290430        total: 15.4s    remaining: 2m 7s
54:     learn: 0.0282138        total: 15.7s    remaining: 2m 6s
55:     learn: 0.0275641        total: 16s      remaining: 2m 6s
56:     learn: 0.0269046        total: 16.3s    remaining: 2m 6s
57:     learn: 0.0262992        total: 16.6s    remaining: 2m 6s
58:     learn: 0.0258206        total: 16.9s    remaining: 2m 6s
59:     learn: 0.0251201        total: 17.1s    remaining: 2m 5s
60:     learn: 0.0246300        total: 17.4s    remaining: 2m 5s
61:     learn: 0.0241168        total: 17.7s    remaining: 2m 5s
62:     learn: 0.0237970        total: 18s      remaining: 2m 4s
63:     learn: 0.0235365        total: 18.3s    remaining: 2m 4s
64:     learn: 0.0231297        total: 18.6s    remaining: 2m 4s
65:     learn: 0.0228621        total: 18.9s    remaining: 2m 4s
66:     learn: 0.0224374        total: 19.2s    remaining: 2m 3s
67:     learn: 0.0220935        total: 19.5s    remaining: 2m 3s
68:     learn: 0.0219222        total: 19.7s    remaining: 2m 3s
69:     learn: 0.0217011        total: 20s      remaining: 2m 2s
70:     learn: 0.0214294        total: 20.3s    remaining: 2m 2s
71:     learn: 0.0210343        total: 20.6s    remaining: 2m 2s
72:     learn: 0.0206492        total: 20.9s    remaining: 2m 2s
73:     learn: 0.0203818        total: 21.2s    remaining: 2m 2s
74:     learn: 0.0199555        total: 21.5s    remaining: 2m 2s
75:     learn: 0.0196339        total: 21.8s    remaining: 2m 1s
76:     learn: 0.0194957        total: 22.2s    remaining: 2m 1s
77:     learn: 0.0193141        total: 22.5s    remaining: 2m 1s
78:     learn: 0.0190639        total: 22.8s    remaining: 2m 1s
79:     learn: 0.0187859        total: 23.1s    remaining: 2m 1s
80:     learn: 0.0183823        total: 23.4s    remaining: 2m 1s
81:     learn: 0.0180045        total: 23.7s    remaining: 2m
82:     learn: 0.0178837        total: 24s      remaining: 2m
83:     learn: 0.0176034        total: 24.3s    remaining: 2m
84:     learn: 0.0173981        total: 24.6s    remaining: 2m
85:     learn: 0.0171066        total: 24.9s    remaining: 1m 59s
86:     learn: 0.0168718        total: 25.2s    remaining: 1m 59s
```

```
87:     learn: 0.0166536        total: 25.5s    remaining: 1m 59s
88:     learn: 0.0165591        total: 25.8s    remaining: 1m 59s
89:     learn: 0.0162736        total: 26.1s    remaining: 1m 59s
90:     learn: 0.0160416        total: 26.4s    remaining: 1m 58s
91:     learn: 0.0159524        total: 26.7s    remaining: 1m 58s
92:     learn: 0.0157946        total: 27s      remaining: 1m 58s
93:     learn: 0.0156162        total: 27.3s    remaining: 1m 57s
94:     learn: 0.0154868        total: 27.5s    remaining: 1m 57s
95:     learn: 0.0153195        total: 27.9s    remaining: 1m 57s
96:     learn: 0.0152243        total: 28.2s    remaining: 1m 56s
97:     learn: 0.0150813        total: 28.4s    remaining: 1m 56s
98:     learn: 0.0149722        total: 28.7s    remaining: 1m 56s
99:     learn: 0.0148357        total: 29.1s    remaining: 1m 56s
100:    learn: 0.0147473        total: 29.3s    remaining: 1m 55s
101:    learn: 0.0146174        total: 29.6s    remaining: 1m 55s
102:    learn: 0.0145097        total: 29.9s    remaining: 1m 55s
103:    learn: 0.0143843        total: 30.2s    remaining: 1m 54s
104:    learn: 0.0142433        total: 30.5s    remaining: 1m 54s
105:    learn: 0.0140882        total: 30.8s    remaining: 1m 54s
106:    learn: 0.0140314        total: 31.1s    remaining: 1m 54s
107:    learn: 0.0139732        total: 31.4s    remaining: 1m 53s
108:    learn: 0.0138087        total: 31.7s    remaining: 1m 53s
109:    learn: 0.0136995        total: 32s      remaining: 1m 53s
110:    learn: 0.0136129        total: 32.2s    remaining: 1m 52s
111:    learn: 0.0134701        total: 32.6s    remaining: 1m 52s
112:    learn: 0.0133851        total: 32.8s    remaining: 1m 52s
113:    learn: 0.0133056        total: 33.2s    remaining: 1m 52s
114:    learn: 0.0131529        total: 33.5s    remaining: 1m 52s
115:    learn: 0.0131119        total: 33.7s    remaining: 1m 51s
116:    learn: 0.0129847        total: 34s      remaining: 1m 51s
117:    learn: 0.0129260        total: 34.3s    remaining: 1m 51s
118:    learn: 0.0128358        total: 34.6s    remaining: 1m 50s
119:    learn: 0.0127588        total: 34.9s    remaining: 1m 50s
120:    learn: 0.0126759        total: 35.2s    remaining: 1m 50s
121:    learn: 0.0125875        total: 35.4s    remaining: 1m 49s
122:    learn: 0.0125070        total: 35.7s    remaining: 1m 49s
123:    learn: 0.0124040        total: 36s      remaining: 1m 49s
124:    learn: 0.0123378        total: 36.3s    remaining: 1m 48s
125:    learn: 0.0122476        total: 36.6s    remaining: 1m 48s
126:    learn: 0.0121334        total: 36.9s    remaining: 1m 48s
127:    learn: 0.0120318        total: 37.1s    remaining: 1m 47s
128:    learn: 0.0119657        total: 37.4s    remaining: 1m 47s
129:    learn: 0.0119029        total: 37.7s    remaining: 1m 47s
130:    learn: 0.0118748        total: 38s      remaining: 1m 46s
131:    learn: 0.0118110        total: 38.2s    remaining: 1m 46s
132:    learn: 0.0117188        total: 38.5s    remaining: 1m 46s
133:    learn: 0.0116832        total: 38.8s    remaining: 1m 46s
134:    learn: 0.0115996        total: 39.1s    remaining: 1m 45s
```

```
135:    learn: 0.0115289        total: 39.4s    remaining: 1m 45s
136:    learn: 0.0114979        total: 39.7s    remaining: 1m 45s
137:    learn: 0.0114327        total: 40s      remaining: 1m 44s
138:    learn: 0.0113826        total: 40.3s    remaining: 1m 44s
139:    learn: 0.0113010        total: 40.6s    remaining: 1m 44s
140:    learn: 0.0112550        total: 40.9s    remaining: 1m 44s
141:    learn: 0.0112112        total: 41.2s    remaining: 1m 43s
142:    learn: 0.0111395        total: 41.6s    remaining: 1m 43s
143:    learn: 0.0110791        total: 41.9s    remaining: 1m 43s
144:    learn: 0.0110409        total: 42.2s    remaining: 1m 43s
145:    learn: 0.0109684        total: 42.5s    remaining: 1m 43s
146:    learn: 0.0109264        total: 42.8s    remaining: 1m 42s
147:    learn: 0.0108521        total: 43s      remaining: 1m 42s
148:    learn: 0.0108085        total: 43.3s    remaining: 1m 42s
149:    learn: 0.0107366        total: 43.6s    remaining: 1m 41s
150:    learn: 0.0106745        total: 43.9s    remaining: 1m 41s
151:    learn: 0.0106230        total: 44.2s    remaining: 1m 41s
152:    learn: 0.0105880        total: 44.5s    remaining: 1m 40s
153:    learn: 0.0105477        total: 44.8s    remaining: 1m 40s
154:    learn: 0.0105218        total: 45.1s    remaining: 1m 40s
155:    learn: 0.0104212        total: 45.4s    remaining: 1m 40s
156:    learn: 0.0103510        total: 45.6s    remaining: 1m 39s
157:    learn: 0.0102919        total: 45.9s    remaining: 1m 39s
158:    learn: 0.0102430        total: 46.2s    remaining: 1m 39s
159:    learn: 0.0101876        total: 46.5s    remaining: 1m 38s
160:    learn: 0.0101420        total: 46.8s    remaining: 1m 38s
161:    learn: 0.0100883        total: 47.1s    remaining: 1m 38s
162:    learn: 0.0100284        total: 47.4s    remaining: 1m 37s
163:    learn: 0.0099873        total: 47.7s    remaining: 1m 37s
164:    learn: 0.0099459        total: 48s      remaining: 1m 37s
165:    learn: 0.0099082        total: 48.3s    remaining: 1m 37s
166:    learn: 0.0098673        total: 48.6s    remaining: 1m 36s
167:    learn: 0.0098064        total: 48.9s    remaining: 1m 36s
168:    learn: 0.0097911        total: 49.1s    remaining: 1m 36s
169:    learn: 0.0097383        total: 49.4s    remaining: 1m 35s
170:    learn: 0.0096936        total: 49.7s    remaining: 1m 35s
171:    learn: 0.0096380        total: 49.9s    remaining: 1m 35s
172:    learn: 0.0095849        total: 50.3s    remaining: 1m 34s
173:    learn: 0.0095547        total: 50.5s    remaining: 1m 34s
174:    learn: 0.0094981        total: 50.8s    remaining: 1m 34s
175:    learn: 0.0094305        total: 51.1s    remaining: 1m 34s
176:    learn: 0.0093093        total: 51.4s    remaining: 1m 33s
177:    learn: 0.0092725        total: 51.7s    remaining: 1m 33s
178:    learn: 0.0092068        total: 52s      remaining: 1m 33s
179:    learn: 0.0091675        total: 52.3s    remaining: 1m 32s
180:    learn: 0.0091159        total: 52.6s    remaining: 1m 32s
181:    learn: 0.0090922        total: 52.8s    remaining: 1m 32s
182:    learn: 0.0090531        total: 53.1s    remaining: 1m 32s
```

```
183:    learn: 0.0090144        total: 53.4s    remaining: 1m 31s
184:    learn: 0.0089546        total: 53.7s    remaining: 1m 31s
185:    learn: 0.0089157        total: 54s      remaining: 1m 31s
186:    learn: 0.0088883        total: 54.3s    remaining: 1m 30s
187:    learn: 0.0088482        total: 54.5s    remaining: 1m 30s
188:    learn: 0.0088167        total: 54.8s    remaining: 1m 30s
189:    learn: 0.0087874        total: 55.2s    remaining: 1m 30s
190:    learn: 0.0087393        total: 55.5s    remaining: 1m 29s
191:    learn: 0.0086925        total: 55.7s    remaining: 1m 29s
192:    learn: 0.0086483        total: 56s      remaining: 1m 29s
193:    learn: 0.0086208        total: 56.3s    remaining: 1m 28s
194:    learn: 0.0085887        total: 56.6s    remaining: 1m 28s
195:    learn: 0.0085548        total: 56.9s    remaining: 1m 28s
196:    learn: 0.0085172        total: 57.2s    remaining: 1m 27s
197:    learn: 0.0084680        total: 57.5s    remaining: 1m 27s
198:    learn: 0.0084230        total: 57.8s    remaining: 1m 27s
199:    learn: 0.0083771        total: 58s      remaining: 1m 26s
200:    learn: 0.0083432        total: 58.2s    remaining: 1m 26s
201:    learn: 0.0083121        total: 58.5s    remaining: 1m 26s
202:    learn: 0.0082748        total: 58.8s    remaining: 1m 26s
203:    learn: 0.0082430        total: 59.1s    remaining: 1m 25s
204:    learn: 0.0082177        total: 59.4s    remaining: 1m 25s
205:    learn: 0.0081707        total: 59.7s    remaining: 1m 25s
206:    learn: 0.0081335        total: 60s      remaining: 1m 24s
207:    learn: 0.0081226        total: 1m       remaining: 1m 24s
208:    learn: 0.0080965        total: 1m       remaining: 1m 24s
209:    learn: 0.0080701        total: 1m       remaining: 1m 23s
210:    learn: 0.0080298        total: 1m 1s    remaining: 1m 23s
211:    learn: 0.0079989        total: 1m 1s    remaining: 1m 23s
212:    learn: 0.0079689        total: 1m 1s    remaining: 1m 23s
213:    learn: 0.0079498        total: 1m 2s    remaining: 1m 22s
214:    learn: 0.0079243        total: 1m 2s    remaining: 1m 22s
215:    learn: 0.0078946        total: 1m 2s    remaining: 1m 22s
216:    learn: 0.0078694        total: 1m 2s    remaining: 1m 22s
217:    learn: 0.0078235        total: 1m 3s    remaining: 1m 21s
218:    learn: 0.0077662        total: 1m 3s    remaining: 1m 21s
219:    learn: 0.0077242        total: 1m 4s    remaining: 1m 21s
220:    learn: 0.0076794        total: 1m 4s    remaining: 1m 21s
221:    learn: 0.0076408        total: 1m 4s    remaining: 1m 21s
222:    learn: 0.0075484        total: 1m 5s    remaining: 1m 21s
223:    learn: 0.0075175        total: 1m 5s    remaining: 1m 20s
224:    learn: 0.0074760        total: 1m 5s    remaining: 1m 20s
225:    learn: 0.0074342        total: 1m 6s    remaining: 1m 20s
226:    learn: 0.0074087        total: 1m 6s    remaining: 1m 19s
227:    learn: 0.0073870        total: 1m 6s    remaining: 1m 19s
228:    learn: 0.0073535        total: 1m 7s    remaining: 1m 19s
229:    learn: 0.0073137        total: 1m 7s    remaining: 1m 19s
230:    learn: 0.0072848        total: 1m 7s    remaining: 1m 18s
```

```
231:    learn: 0.0072450        total: 1m 8s    remaining: 1m 18s
232:    learn: 0.0071893        total: 1m 8s    remaining: 1m 18s
233:    learn: 0.0071624        total: 1m 8s    remaining: 1m 18s
234:    learn: 0.0071318        total: 1m 8s    remaining: 1m 17s
235:    learn: 0.0070939        total: 1m 9s    remaining: 1m 17s
236:    learn: 0.0070735        total: 1m 9s    remaining: 1m 17s
237:    learn: 0.0070433        total: 1m 9s    remaining: 1m 16s
238:    learn: 0.0070039        total: 1m 10s   remaining: 1m 16s
239:    learn: 0.0069812        total: 1m 10s   remaining: 1m 16s
240:    learn: 0.0069457        total: 1m 10s   remaining: 1m 15s
241:    learn: 0.0069240        total: 1m 10s   remaining: 1m 15s
242:    learn: 0.0068707        total: 1m 11s   remaining: 1m 15s
243:    learn: 0.0068226        total: 1m 11s   remaining: 1m 15s
244:    learn: 0.0068042        total: 1m 11s   remaining: 1m 14s
245:    learn: 0.0067793        total: 1m 12s   remaining: 1m 14s
246:    learn: 0.0067432        total: 1m 12s   remaining: 1m 14s
247:    learn: 0.0066913        total: 1m 12s   remaining: 1m 13s
248:    learn: 0.0066562        total: 1m 13s   remaining: 1m 13s
249:    learn: 0.0065926        total: 1m 13s   remaining: 1m 13s
250:    learn: 0.0065647        total: 1m 13s   remaining: 1m 13s
251:    learn: 0.0065471        total: 1m 13s   remaining: 1m 12s
252:    learn: 0.0065195        total: 1m 14s   remaining: 1m 12s
253:    learn: 0.0064971        total: 1m 14s   remaining: 1m 12s
254:    learn: 0.0064680        total: 1m 14s   remaining: 1m 11s
255:    learn: 0.0064261        total: 1m 15s   remaining: 1m 11s
256:    learn: 0.0064197        total: 1m 15s   remaining: 1m 11s
257:    learn: 0.0063695        total: 1m 15s   remaining: 1m 10s
258:    learn: 0.0063508        total: 1m 15s   remaining: 1m 10s
259:    learn: 0.0063162        total: 1m 16s   remaining: 1m 10s
260:    learn: 0.0062678        total: 1m 16s   remaining: 1m 10s
261:    learn: 0.0062499        total: 1m 16s   remaining: 1m 9s
262:    learn: 0.0062153        total: 1m 17s   remaining: 1m 9s
263:    learn: 0.0061887        total: 1m 17s   remaining: 1m 9s
264:    learn: 0.0061564        total: 1m 17s   remaining: 1m 8s
265:    learn: 0.0061307        total: 1m 18s   remaining: 1m 8s
266:    learn: 0.0060967        total: 1m 18s   remaining: 1m 8s
267:    learn: 0.0060788        total: 1m 18s   remaining: 1m 8s
268:    learn: 0.0060519        total: 1m 18s   remaining: 1m 7s
269:    learn: 0.0060192        total: 1m 19s   remaining: 1m 7s
270:    learn: 0.0059992        total: 1m 19s   remaining: 1m 7s
271:    learn: 0.0059757        total: 1m 19s   remaining: 1m 6s
272:    learn: 0.0059466        total: 1m 20s   remaining: 1m 6s
273:    learn: 0.0059238        total: 1m 20s   remaining: 1m 6s
274:    learn: 0.0059034        total: 1m 20s   remaining: 1m 6s
275:    learn: 0.0058832        total: 1m 21s   remaining: 1m 5s
276:    learn: 0.0058632        total: 1m 21s   remaining: 1m 5s
277:    learn: 0.0058335        total: 1m 21s   remaining: 1m 5s
278:    learn: 0.0058059        total: 1m 21s   remaining: 1m 4s
```

```
279:    learn: 0.0057925        total: 1m 22s   remaining: 1m 4s
280:    learn: 0.0057721        total: 1m 22s   remaining: 1m 4s
281:    learn: 0.0057489        total: 1m 22s   remaining: 1m 4s
282:    learn: 0.0057293        total: 1m 23s   remaining: 1m 3s
283:    learn: 0.0057013        total: 1m 23s   remaining: 1m 3s
284:    learn: 0.0056668        total: 1m 23s   remaining: 1m 3s
285:    learn: 0.0056471        total: 1m 23s   remaining: 1m 2s
286:    learn: 0.0056374        total: 1m 24s   remaining: 1m 2s
287:    learn: 0.0056041        total: 1m 24s   remaining: 1m 2s
288:    learn: 0.0055856        total: 1m 24s   remaining: 1m 1s
289:    learn: 0.0055660        total: 1m 25s   remaining: 1m 1s
290:    learn: 0.0055375        total: 1m 25s   remaining: 1m 1s
291:    learn: 0.0055232        total: 1m 25s   remaining: 1m 1s
292:    learn: 0.0055061        total: 1m 25s   remaining: 1m
293:    learn: 0.0054914        total: 1m 26s   remaining: 1m
294:    learn: 0.0054731        total: 1m 26s   remaining: 1m
295:    learn: 0.0054337        total: 1m 26s   remaining: 59.8s
296:    learn: 0.0054026        total: 1m 27s   remaining: 59.5s
297:    learn: 0.0053761        total: 1m 27s   remaining: 59.2s
298:    learn: 0.0053542        total: 1m 27s   remaining: 58.9s
299:    learn: 0.0053232        total: 1m 27s   remaining: 58.7s
300:    learn: 0.0053067        total: 1m 28s   remaining: 58.4s
301:    learn: 0.0052928        total: 1m 28s   remaining: 58.1s
302:    learn: 0.0052602        total: 1m 28s   remaining: 57.7s
303:    learn: 0.0052422        total: 1m 29s   remaining: 57.4s
304:    learn: 0.0052194        total: 1m 29s   remaining: 57.1s
305:    learn: 0.0051998        total: 1m 29s   remaining: 56.8s
306:    learn: 0.0051685        total: 1m 29s   remaining: 56.5s
307:    learn: 0.0051521        total: 1m 30s   remaining: 56.2s
308:    learn: 0.0051333        total: 1m 30s   remaining: 56s
309:    learn: 0.0051138        total: 1m 30s   remaining: 55.7s
310:    learn: 0.0050991        total: 1m 31s   remaining: 55.4s
311:    learn: 0.0050803        total: 1m 31s   remaining: 55.1s
312:    learn: 0.0050551        total: 1m 31s   remaining: 54.8s
313:    learn: 0.0050244        total: 1m 31s   remaining: 54.5s
314:    learn: 0.0050087        total: 1m 32s   remaining: 54.2s
315:    learn: 0.0049906        total: 1m 32s   remaining: 53.9s
316:    learn: 0.0049703        total: 1m 32s   remaining: 53.6s
317:    learn: 0.0049478        total: 1m 33s   remaining: 53.3s
318:    learn: 0.0049294        total: 1m 33s   remaining: 53s
319:    learn: 0.0048992        total: 1m 33s   remaining: 52.7s
320:    learn: 0.0048883        total: 1m 33s   remaining: 52.4s
321:    learn: 0.0048743        total: 1m 34s   remaining: 52.1s
322:    learn: 0.0048558        total: 1m 34s   remaining: 51.8s
323:    learn: 0.0048355        total: 1m 34s   remaining: 51.5s
324:    learn: 0.0048232        total: 1m 35s   remaining: 51.2s
325:    learn: 0.0048055        total: 1m 35s   remaining: 50.9s
326:    learn: 0.0047833        total: 1m 35s   remaining: 50.6s
```

```
327:    learn: 0.0047609        total: 1m 35s   remaining: 50.3s
328:    learn: 0.0047513        total: 1m 36s   remaining: 50s
329:    learn: 0.0047320        total: 1m 36s   remaining: 49.7s
330:    learn: 0.0047157        total: 1m 36s   remaining: 49.5s
331:    learn: 0.0046965        total: 1m 37s   remaining: 49.2s
332:    learn: 0.0046835        total: 1m 37s   remaining: 48.9s
333:    learn: 0.0046689        total: 1m 37s   remaining: 48.6s
334:    learn: 0.0046462        total: 1m 38s   remaining: 48.3s
335:    learn: 0.0046257        total: 1m 38s   remaining: 48s
336:    learn: 0.0046114        total: 1m 38s   remaining: 47.7s
337:    learn: 0.0045886        total: 1m 38s   remaining: 47.4s
338:    learn: 0.0045859        total: 1m 39s   remaining: 47.1s
339:    learn: 0.0045736        total: 1m 39s   remaining: 46.8s
340:    learn: 0.0045491        total: 1m 39s   remaining: 46.5s
341:    learn: 0.0045181        total: 1m 40s   remaining: 46.2s
342:    learn: 0.0044987        total: 1m 40s   remaining: 45.9s
343:    learn: 0.0044919        total: 1m 40s   remaining: 45.6s
344:    learn: 0.0044851        total: 1m 40s   remaining: 45.3s
345:    learn: 0.0044718        total: 1m 41s   remaining: 45s
346:    learn: 0.0044610        total: 1m 41s   remaining: 44.7s
347:    learn: 0.0044517        total: 1m 41s   remaining: 44.4s
348:    learn: 0.0044391        total: 1m 42s   remaining: 44.1s
349:    learn: 0.0044237        total: 1m 42s   remaining: 43.8s
350:    learn: 0.0044170        total: 1m 42s   remaining: 43.6s
351:    learn: 0.0044031        total: 1m 42s   remaining: 43.3s
352:    learn: 0.0043849        total: 1m 43s   remaining: 43s
353:    learn: 0.0043694        total: 1m 43s   remaining: 42.7s
354:    learn: 0.0043605        total: 1m 43s   remaining: 42.4s
355:    learn: 0.0043530        total: 1m 44s   remaining: 42.1s
356:    learn: 0.0043374        total: 1m 44s   remaining: 41.8s
357:    learn: 0.0043197        total: 1m 44s   remaining: 41.5s
358:    learn: 0.0043082        total: 1m 44s   remaining: 41.2s
359:    learn: 0.0042977        total: 1m 45s   remaining: 40.9s
360:    learn: 0.0042838        total: 1m 45s   remaining: 40.6s
361:    learn: 0.0042640        total: 1m 45s   remaining: 40.4s
362:    learn: 0.0042389        total: 1m 46s   remaining: 40.1s
363:    learn: 0.0042236        total: 1m 46s   remaining: 39.8s
364:    learn: 0.0042087        total: 1m 46s   remaining: 39.5s
365:    learn: 0.0041905        total: 1m 47s   remaining: 39.2s
366:    learn: 0.0041780        total: 1m 47s   remaining: 38.9s
367:    learn: 0.0041680        total: 1m 47s   remaining: 38.6s
368:    learn: 0.0041472        total: 1m 47s   remaining: 38.3s
369:    learn: 0.0041303        total: 1m 48s   remaining: 38s
370:    learn: 0.0041262        total: 1m 48s   remaining: 37.7s
371:    learn: 0.0041114        total: 1m 48s   remaining: 37.5s
372:    learn: 0.0041049        total: 1m 49s   remaining: 37.2s
373:    learn: 0.0040919        total: 1m 49s   remaining: 36.9s
374:    learn: 0.0040804        total: 1m 49s   remaining: 36.6s
```

```
375:    learn: 0.0040573        total: 1m 50s   remaining: 36.3s
376:    learn: 0.0040397        total: 1m 50s   remaining: 36s
377:    learn: 0.0040240        total: 1m 50s   remaining: 35.7s
378:    learn: 0.0040106        total: 1m 50s   remaining: 35.4s
379:    learn: 0.0039977        total: 1m 51s   remaining: 35.1s
380:    learn: 0.0039881        total: 1m 51s   remaining: 34.8s
381:    learn: 0.0039671        total: 1m 51s   remaining: 34.5s
382:    learn: 0.0039568        total: 1m 52s   remaining: 34.3s
383:    learn: 0.0039429        total: 1m 52s   remaining: 34s
384:    learn: 0.0039033        total: 1m 52s   remaining: 33.7s
385:    learn: 0.0038870        total: 1m 52s   remaining: 33.4s
386:    learn: 0.0038820        total: 1m 53s   remaining: 33.1s
387:    learn: 0.0038799        total: 1m 53s   remaining: 32.8s
388:    learn: 0.0038645        total: 1m 53s   remaining: 32.5s
389:    learn: 0.0038417        total: 1m 54s   remaining: 32.2s
390:    learn: 0.0038332        total: 1m 54s   remaining: 31.8s
391:    learn: 0.0038207        total: 1m 54s   remaining: 31.6s
392:    learn: 0.0038125        total: 1m 54s   remaining: 31.3s
393:    learn: 0.0038043        total: 1m 55s   remaining: 31s
394:    learn: 0.0037889        total: 1m 55s   remaining: 30.7s
395:    learn: 0.0037674        total: 1m 55s   remaining: 30.4s
396:    learn: 0.0037541        total: 1m 55s   remaining: 30.1s
397:    learn: 0.0037403        total: 1m 56s   remaining: 29.8s
398:    learn: 0.0037319        total: 1m 56s   remaining: 29.5s
399:    learn: 0.0037200        total: 1m 56s   remaining: 29.2s
400:    learn: 0.0037038        total: 1m 57s   remaining: 28.9s
401:    learn: 0.0036914        total: 1m 57s   remaining: 28.6s
402:    learn: 0.0036813        total: 1m 57s   remaining: 28.3s
403:    learn: 0.0036715        total: 1m 57s   remaining: 28s
404:    learn: 0.0036658        total: 1m 58s   remaining: 27.7s
405:    learn: 0.0036528        total: 1m 58s   remaining: 27.4s
406:    learn: 0.0036413        total: 1m 58s   remaining: 27.2s
407:    learn: 0.0036340        total: 1m 59s   remaining: 26.9s
408:    learn: 0.0036238        total: 1m 59s   remaining: 26.6s
409:    learn: 0.0036095        total: 1m 59s   remaining: 26.3s
410:    learn: 0.0036029        total: 2m       remaining: 26s
411:    learn: 0.0035950        total: 2m       remaining: 25.7s
412:    learn: 0.0035848        total: 2m       remaining: 25.4s
413:    learn: 0.0035762        total: 2m       remaining: 25.1s
414:    learn: 0.0035634        total: 2m 1s    remaining: 24.8s
415:    learn: 0.0035544        total: 2m 1s    remaining: 24.5s
416:    learn: 0.0035414        total: 2m 1s    remaining: 24.2s
417:    learn: 0.0035333        total: 2m 2s    remaining: 23.9s
418:    learn: 0.0035222        total: 2m 2s    remaining: 23.7s
419:    learn: 0.0035145        total: 2m 2s    remaining: 23.4s
420:    learn: 0.0035075        total: 2m 2s    remaining: 23.1s
421:    learn: 0.0034945        total: 2m 3s    remaining: 22.8s
422:    learn: 0.0034811        total: 2m 3s    remaining: 22.5s
```

```
423:    learn: 0.0034679        total: 2m 3s    remaining: 22.2s
424:    learn: 0.0034591        total: 2m 4s    remaining: 21.9s
425:    learn: 0.0034499        total: 2m 4s    remaining: 21.6s
426:    learn: 0.0034314        total: 2m 4s    remaining: 21.3s
427:    learn: 0.0034232        total: 2m 5s    remaining: 21s
428:    learn: 0.0034106        total: 2m 5s    remaining: 20.7s
429:    learn: 0.0033947        total: 2m 5s    remaining: 20.4s
430:    learn: 0.0033917        total: 2m 5s    remaining: 20.1s
431:    learn: 0.0033837        total: 2m 6s    remaining: 19.9s
432:    learn: 0.0033612        total: 2m 6s    remaining: 19.6s
433:    learn: 0.0033568        total: 2m 6s    remaining: 19.3s
434:    learn: 0.0033492        total: 2m 7s    remaining: 19s
435:    learn: 0.0033451        total: 2m 7s    remaining: 18.7s
436:    learn: 0.0033348        total: 2m 7s    remaining: 18.4s
437:    learn: 0.0033248        total: 2m 7s    remaining: 18.1s
438:    learn: 0.0033196        total: 2m 8s    remaining: 17.8s
439:    learn: 0.0033068        total: 2m 8s    remaining: 17.5s
440:    learn: 0.0032958        total: 2m 8s    remaining: 17.2s
441:    learn: 0.0032889        total: 2m 9s    remaining: 16.9s
442:    learn: 0.0032775        total: 2m 9s    remaining: 16.6s
443:    learn: 0.0032685        total: 2m 9s    remaining: 16.4s
444:    learn: 0.0032632        total: 2m 9s    remaining: 16.1s
445:    learn: 0.0032525        total: 2m 10s   remaining: 15.8s
446:    learn: 0.0032438        total: 2m 10s   remaining: 15.5s
447:    learn: 0.0032323        total: 2m 10s   remaining: 15.2s
448:    learn: 0.0032113        total: 2m 11s   remaining: 14.9s
449:    learn: 0.0031998        total: 2m 11s   remaining: 14.6s
450:    learn: 0.0031926        total: 2m 11s   remaining: 14.3s
451:    learn: 0.0031852        total: 2m 11s   remaining: 14s
452:    learn: 0.0031703        total: 2m 12s   remaining: 13.7s
453:    learn: 0.0031597        total: 2m 12s   remaining: 13.4s
454:    learn: 0.0031473        total: 2m 12s   remaining: 13.1s
455:    learn: 0.0031374        total: 2m 13s   remaining: 12.8s
456:    learn: 0.0031272        total: 2m 13s   remaining: 12.6s
457:    learn: 0.0031148        total: 2m 13s   remaining: 12.3s
458:    learn: 0.0031037        total: 2m 13s   remaining: 12s
459:    learn: 0.0030941        total: 2m 14s   remaining: 11.7s
460:    learn: 0.0030836        total: 2m 14s   remaining: 11.4s
461:    learn: 0.0030783        total: 2m 14s   remaining: 11.1s
462:    learn: 0.0030611        total: 2m 15s   remaining: 10.8s
463:    learn: 0.0030468        total: 2m 15s   remaining: 10.5s
464:    learn: 0.0030340        total: 2m 15s   remaining: 10.2s
465:    learn: 0.0030254        total: 2m 15s   remaining: 9.92s
466:    learn: 0.0030134        total: 2m 16s   remaining: 9.63s
467:    learn: 0.0030017        total: 2m 16s   remaining: 9.34s
468:    learn: 0.0029939        total: 2m 16s   remaining: 9.04s
469:    learn: 0.0029759        total: 2m 17s   remaining: 8.75s
470:    learn: 0.0029737        total: 2m 17s   remaining: 8.46s
```

```
471:    learn: 0.0029640        total: 2m 17s   remaining: 8.17s
472:    learn: 0.0029550        total: 2m 17s   remaining: 7.88s
473:    learn: 0.0029482        total: 2m 18s   remaining: 7.58s
474:    learn: 0.0029426        total: 2m 18s   remaining: 7.29s
475:    learn: 0.0029365        total: 2m 18s   remaining: 7s
476:    learn: 0.0029303        total: 2m 19s   remaining: 6.71s
477:    learn: 0.0029192        total: 2m 19s   remaining: 6.42s
478:    learn: 0.0029119        total: 2m 19s   remaining: 6.12s
479:    learn: 0.0029006        total: 2m 19s   remaining: 5.83s
480:    learn: 0.0028910        total: 2m 20s   remaining: 5.54s
481:    learn: 0.0028842        total: 2m 20s   remaining: 5.25s
482:    learn: 0.0028690        total: 2m 20s   remaining: 4.96s
483:    learn: 0.0028622        total: 2m 21s   remaining: 4.67s
484:    learn: 0.0028505        total: 2m 21s   remaining: 4.38s
485:    learn: 0.0028396        total: 2m 21s   remaining: 4.08s
486:    learn: 0.0028279        total: 2m 22s   remaining: 3.79s
487:    learn: 0.0028249        total: 2m 22s   remaining: 3.5s
488:    learn: 0.0028138        total: 2m 22s   remaining: 3.21s
489:    learn: 0.0028051        total: 2m 22s   remaining: 2.92s
490:    learn: 0.0027996        total: 2m 23s   remaining: 2.63s
491:    learn: 0.0027870        total: 2m 23s   remaining: 2.33s
492:    learn: 0.0027754        total: 2m 23s   remaining: 2.04s
493:    learn: 0.0027703        total: 2m 24s   remaining: 1.75s
494:    learn: 0.0027621        total: 2m 24s   remaining: 1.46s
495:    learn: 0.0027515        total: 2m 24s   remaining: 1.17s
496:    learn: 0.0027447        total: 2m 25s   remaining: 876ms
497:    learn: 0.0027344        total: 2m 25s   remaining: 584ms
498:    learn: 0.0027280        total: 2m 25s   remaining: 292ms
499:    learn: 0.0027205        total: 2m 25s   remaining: 0us
0:      learn: 0.9593295        total: 219ms    remaining: 1m 49s
1:      learn: 0.8455813        total: 498ms    remaining: 2m 4s
2:      learn: 0.7559217        total: 784ms    remaining: 2m 9s
3:      learn: 0.6692340        total: 1.07s    remaining: 2m 12s
4:      learn: 0.5984585        total: 1.44s    remaining: 2m 22s
5:      learn: 0.5394063        total: 1.78s    remaining: 2m 26s
6:      learn: 0.4900241        total: 2.08s    remaining: 2m 26s
7:      learn: 0.4434525        total: 2.38s    remaining: 2m 26s
8:      learn: 0.4031443        total: 2.66s    remaining: 2m 25s
9:      learn: 0.3681809        total: 2.95s    remaining: 2m 24s
10:     learn: 0.3355321        total: 3.24s    remaining: 2m 24s
11:     learn: 0.3032481        total: 3.49s    remaining: 2m 22s
12:     learn: 0.2748559        total: 3.75s    remaining: 2m 20s
13:     learn: 0.2497911        total: 4s       remaining: 2m 19s
14:     learn: 0.2283837        total: 4.25s    remaining: 2m 17s
15:     learn: 0.2088453        total: 4.52s    remaining: 2m 16s
16:     learn: 0.1925385        total: 4.79s    remaining: 2m 16s
17:     learn: 0.1769017        total: 5.05s    remaining: 2m 15s
18:     learn: 0.1630361        total: 5.3s     remaining: 2m 14s
```

```
19:     learn: 0.1514862        total: 5.58s    remaining: 2m 14s
20:     learn: 0.1398354        total: 5.83s    remaining: 2m 13s
21:     learn: 0.1298102        total: 6.09s    remaining: 2m 12s
22:     learn: 0.1204080        total: 6.37s    remaining: 2m 12s
23:     learn: 0.1119994        total: 6.64s    remaining: 2m 11s
24:     learn: 0.1044955        total: 6.9s     remaining: 2m 11s
25:     learn: 0.0975524        total: 7.16s    remaining: 2m 10s
26:     learn: 0.0915566        total: 7.45s    remaining: 2m 10s
27:     learn: 0.0861048        total: 7.73s    remaining: 2m 10s
28:     learn: 0.0815264        total: 7.99s    remaining: 2m 9s
29:     learn: 0.0768739        total: 8.26s    remaining: 2m 9s
30:     learn: 0.0722860        total: 8.53s    remaining: 2m 9s
31:     learn: 0.0685459        total: 8.81s    remaining: 2m 8s
32:     learn: 0.0648463        total: 9.11s    remaining: 2m 8s
33:     learn: 0.0616368        total: 9.39s    remaining: 2m 8s
34:     learn: 0.0586667        total: 9.66s    remaining: 2m 8s
35:     learn: 0.0558527        total: 9.96s    remaining: 2m 8s
36:     learn: 0.0532712        total: 10.2s    remaining: 2m 8s
37:     learn: 0.0506299        total: 10.5s    remaining: 2m 7s
38:     learn: 0.0486002        total: 10.8s    remaining: 2m 7s
39:     learn: 0.0464540        total: 11.1s    remaining: 2m 7s
40:     learn: 0.0445658        total: 11.4s    remaining: 2m 7s
41:     learn: 0.0426102        total: 11.8s    remaining: 2m 8s
42:     learn: 0.0409019        total: 12.1s    remaining: 2m 8s
43:     learn: 0.0394256        total: 12.4s    remaining: 2m 8s
44:     learn: 0.0378785        total: 12.7s    remaining: 2m 8s
45:     learn: 0.0365909        total: 13s      remaining: 2m 8s
46:     learn: 0.0353727        total: 13.3s    remaining: 2m 8s
47:     learn: 0.0342549        total: 13.6s    remaining: 2m 8s
48:     learn: 0.0334961        total: 13.9s    remaining: 2m 7s
49:     learn: 0.0323479        total: 14.2s    remaining: 2m 7s
50:     learn: 0.0314733        total: 14.5s    remaining: 2m 7s
51:     learn: 0.0305144        total: 14.8s    remaining: 2m 7s
52:     learn: 0.0298786        total: 15s      remaining: 2m 6s
53:     learn: 0.0291703        total: 15.4s    remaining: 2m 6s
54:     learn: 0.0283882        total: 15.6s    remaining: 2m 6s
55:     learn: 0.0277186        total: 15.9s    remaining: 2m 6s
56:     learn: 0.0269719        total: 16.2s    remaining: 2m 5s
57:     learn: 0.0264202        total: 16.5s    remaining: 2m 5s
58:     learn: 0.0259941        total: 16.8s    remaining: 2m 5s
59:     learn: 0.0253140        total: 17.1s    remaining: 2m 5s
60:     learn: 0.0250838        total: 17.3s    remaining: 2m 4s
61:     learn: 0.0248331        total: 17.5s    remaining: 2m 3s
62:     learn: 0.0245266        total: 17.8s    remaining: 2m 3s
63:     learn: 0.0240583        total: 18.1s    remaining: 2m 3s
64:     learn: 0.0234479        total: 18.4s    remaining: 2m 3s
65:     learn: 0.0230907        total: 18.6s    remaining: 2m 2s
66:     learn: 0.0226567        total: 18.9s    remaining: 2m 2s
```

```
67:     learn: 0.0222759        total: 19.2s    remaining: 2m 2s
68:     learn: 0.0219138        total: 19.5s    remaining: 2m 1s
69:     learn: 0.0215689        total: 19.8s    remaining: 2m 1s
70:     learn: 0.0212712        total: 20.1s    remaining: 2m 1s
71:     learn: 0.0210482        total: 20.4s    remaining: 2m 1s
72:     learn: 0.0207444        total: 20.7s    remaining: 2m
73:     learn: 0.0203467        total: 21s      remaining: 2m
74:     learn: 0.0200351        total: 21.2s    remaining: 2m
75:     learn: 0.0197814        total: 21.5s    remaining: 2m
76:     learn: 0.0195714        total: 21.8s    remaining: 1m 59s
77:     learn: 0.0193869        total: 22.1s    remaining: 1m 59s
78:     learn: 0.0191087        total: 22.4s    remaining: 1m 59s
79:     learn: 0.0189322        total: 22.7s    remaining: 1m 59s
80:     learn: 0.0188715        total: 22.9s    remaining: 1m 58s
81:     learn: 0.0186450        total: 23.2s    remaining: 1m 58s
82:     learn: 0.0183488        total: 23.5s    remaining: 1m 58s
83:     learn: 0.0182058        total: 23.8s    remaining: 1m 57s
84:     learn: 0.0180838        total: 24.1s    remaining: 1m 57s
85:     learn: 0.0178476        total: 24.4s    remaining: 1m 57s
86:     learn: 0.0178090        total: 24.6s    remaining: 1m 56s
87:     learn: 0.0175783        total: 24.9s    remaining: 1m 56s
88:     learn: 0.0174472        total: 25.2s    remaining: 1m 56s
89:     learn: 0.0173256        total: 25.5s    remaining: 1m 56s
90:     learn: 0.0171229        total: 25.8s    remaining: 1m 55s
91:     learn: 0.0170071        total: 26.1s    remaining: 1m 55s
92:     learn: 0.0169431        total: 26.3s    remaining: 1m 55s
93:     learn: 0.0168296        total: 26.6s    remaining: 1m 54s
94:     learn: 0.0167795        total: 26.9s    remaining: 1m 54s
95:     learn: 0.0165038        total: 27.2s    remaining: 1m 54s
96:     learn: 0.0163925        total: 27.5s    remaining: 1m 54s
97:     learn: 0.0161756        total: 27.8s    remaining: 1m 53s
98:     learn: 0.0159652        total: 28s      remaining: 1m 53s
99:     learn: 0.0158093        total: 28.3s    remaining: 1m 53s
100:    learn: 0.0156900        total: 28.6s    remaining: 1m 52s
101:    learn: 0.0155929        total: 28.9s    remaining: 1m 52s
102:    learn: 0.0154322        total: 29.2s    remaining: 1m 52s
103:    learn: 0.0153530        total: 29.5s    remaining: 1m 52s
104:    learn: 0.0151779        total: 29.8s    remaining: 1m 52s
105:    learn: 0.0150749        total: 30.1s    remaining: 1m 51s
106:    learn: 0.0147688        total: 30.4s    remaining: 1m 51s
107:    learn: 0.0146848        total: 30.6s    remaining: 1m 51s
108:    learn: 0.0145557        total: 30.9s    remaining: 1m 50s
109:    learn: 0.0143436        total: 31.2s    remaining: 1m 50s
110:    learn: 0.0142149        total: 31.5s    remaining: 1m 50s
111:    learn: 0.0141079        total: 31.8s    remaining: 1m 50s
112:    learn: 0.0139909        total: 32.1s    remaining: 1m 49s
113:    learn: 0.0138612        total: 32.3s    remaining: 1m 49s
114:    learn: 0.0137806        total: 32.6s    remaining: 1m 49s
```

```
115:    learn: 0.0136555        total: 32.9s    remaining: 1m 48s
116:    learn: 0.0135570        total: 33.2s    remaining: 1m 48s
117:    learn: 0.0134441        total: 33.5s    remaining: 1m 48s
118:    learn: 0.0133631        total: 33.8s    remaining: 1m 48s
119:    learn: 0.0132517        total: 34.1s    remaining: 1m 47s
120:    learn: 0.0131554        total: 34.3s    remaining: 1m 47s
121:    learn: 0.0130716        total: 34.6s    remaining: 1m 47s
122:    learn: 0.0129506        total: 34.9s    remaining: 1m 47s
123:    learn: 0.0129062        total: 35.2s    remaining: 1m 46s
124:    learn: 0.0127673        total: 35.5s    remaining: 1m 46s
125:    learn: 0.0127075        total: 35.7s    remaining: 1m 46s
126:    learn: 0.0126600        total: 36s      remaining: 1m 45s
127:    learn: 0.0125812        total: 36.3s    remaining: 1m 45s
128:    learn: 0.0125041        total: 36.6s    remaining: 1m 45s
129:    learn: 0.0124663        total: 36.8s    remaining: 1m 44s
130:    learn: 0.0123361        total: 37.2s    remaining: 1m 44s
131:    learn: 0.0122806        total: 37.4s    remaining: 1m 44s
132:    learn: 0.0121390        total: 37.7s    remaining: 1m 44s
133:    learn: 0.0120666        total: 38s      remaining: 1m 43s
134:    learn: 0.0120089        total: 38.3s    remaining: 1m 43s
135:    learn: 0.0118901        total: 38.6s    remaining: 1m 43s
136:    learn: 0.0118421        total: 38.9s    remaining: 1m 42s
137:    learn: 0.0117784        total: 39.2s    remaining: 1m 42s
138:    learn: 0.0116701        total: 39.5s    remaining: 1m 42s
139:    learn: 0.0116037        total: 39.8s    remaining: 1m 42s
140:    learn: 0.0115326        total: 40.1s    remaining: 1m 42s
141:    learn: 0.0113977        total: 40.4s    remaining: 1m 41s
142:    learn: 0.0113059        total: 40.6s    remaining: 1m 41s
143:    learn: 0.0112594        total: 40.9s    remaining: 1m 41s
144:    learn: 0.0111827        total: 41.2s    remaining: 1m 40s
145:    learn: 0.0111264        total: 41.5s    remaining: 1m 40s
146:    learn: 0.0110562        total: 41.7s    remaining: 1m 40s
147:    learn: 0.0110033        total: 42s      remaining: 1m 39s
148:    learn: 0.0109566        total: 42.3s    remaining: 1m 39s
149:    learn: 0.0108891        total: 42.6s    remaining: 1m 39s
150:    learn: 0.0108230        total: 42.9s    remaining: 1m 39s
151:    learn: 0.0107789        total: 43.1s    remaining: 1m 38s
152:    learn: 0.0107522        total: 43.4s    remaining: 1m 38s
153:    learn: 0.0106701        total: 43.7s    remaining: 1m 38s
154:    learn: 0.0106332        total: 43.9s    remaining: 1m 37s
155:    learn: 0.0105581        total: 44.2s    remaining: 1m 37s
156:    learn: 0.0105318        total: 44.5s    remaining: 1m 37s
157:    learn: 0.0104918        total: 44.8s    remaining: 1m 36s
158:    learn: 0.0104542        total: 45.1s    remaining: 1m 36s
159:    learn: 0.0104367        total: 45.3s    remaining: 1m 36s
160:    learn: 0.0104289        total: 45.6s    remaining: 1m 36s
161:    learn: 0.0103733        total: 45.9s    remaining: 1m 35s
162:    learn: 0.0103037        total: 46.2s    remaining: 1m 35s
```

```
163:    learn: 0.0102593        total: 46.5s    remaining: 1m 35s
164:    learn: 0.0101511        total: 46.8s    remaining: 1m 35s
165:    learn: 0.0101131        total: 47.1s    remaining: 1m 34s
166:    learn: 0.0100587        total: 47.4s    remaining: 1m 34s
167:    learn: 0.0099923        total: 47.7s    remaining: 1m 34s
168:    learn: 0.0099211        total: 48s      remaining: 1m 33s
169:    learn: 0.0098974        total: 48.3s    remaining: 1m 33s
170:    learn: 0.0098695        total: 48.6s    remaining: 1m 33s
171:    learn: 0.0097711        total: 48.9s    remaining: 1m 33s
172:    learn: 0.0097147        total: 49.2s    remaining: 1m 32s
173:    learn: 0.0096823        total: 49.5s    remaining: 1m 32s
174:    learn: 0.0095939        total: 49.8s    remaining: 1m 32s
175:    learn: 0.0094997        total: 50.1s    remaining: 1m 32s
176:    learn: 0.0094045        total: 50.4s    remaining: 1m 31s
177:    learn: 0.0093632        total: 50.8s    remaining: 1m 31s
178:    learn: 0.0092700        total: 51.1s    remaining: 1m 31s
179:    learn: 0.0092344        total: 51.4s    remaining: 1m 31s
180:    learn: 0.0092017        total: 51.7s    remaining: 1m 31s
181:    learn: 0.0091527        total: 51.9s    remaining: 1m 30s
182:    learn: 0.0091278        total: 52.3s    remaining: 1m 30s
183:    learn: 0.0090624        total: 52.5s    remaining: 1m 30s
184:    learn: 0.0090410        total: 52.8s    remaining: 1m 29s
185:    learn: 0.0089962        total: 53.2s    remaining: 1m 29s
186:    learn: 0.0089462        total: 53.4s    remaining: 1m 29s
187:    learn: 0.0088990        total: 53.8s    remaining: 1m 29s
188:    learn: 0.0088593        total: 54s      remaining: 1m 28s
189:    learn: 0.0087717        total: 54.3s    remaining: 1m 28s
190:    learn: 0.0087458        total: 54.7s    remaining: 1m 28s
191:    learn: 0.0086603        total: 55s      remaining: 1m 28s
192:    learn: 0.0086253        total: 55.3s    remaining: 1m 27s
193:    learn: 0.0085544        total: 55.6s    remaining: 1m 27s
194:    learn: 0.0084997        total: 55.9s    remaining: 1m 27s
195:    learn: 0.0084889        total: 56.2s    remaining: 1m 27s
196:    learn: 0.0084333        total: 56.5s    remaining: 1m 26s
197:    learn: 0.0083997        total: 56.8s    remaining: 1m 26s
198:    learn: 0.0083533        total: 57.1s    remaining: 1m 26s
199:    learn: 0.0083177        total: 57.4s    remaining: 1m 26s
200:    learn: 0.0082984        total: 57.7s    remaining: 1m 25s
201:    learn: 0.0082492        total: 58s      remaining: 1m 25s
202:    learn: 0.0082069        total: 58.2s    remaining: 1m 25s
203:    learn: 0.0081984        total: 58.5s    remaining: 1m 24s
204:    learn: 0.0081345        total: 58.8s    remaining: 1m 24s
205:    learn: 0.0081194        total: 59s      remaining: 1m 24s
206:    learn: 0.0080809        total: 59.3s    remaining: 1m 24s
207:    learn: 0.0080362        total: 59.7s    remaining: 1m 23s
208:    learn: 0.0080032        total: 59.9s    remaining: 1m 23s
209:    learn: 0.0079692        total: 1m       remaining: 1m 23s
210:    learn: 0.0079369        total: 1m       remaining: 1m 22s
```

```
211:    learn: 0.0079082        total: 1m       remaining: 1m 22s
212:    learn: 0.0078902        total: 1m 1s    remaining: 1m 22s
213:    learn: 0.0078626        total: 1m 1s    remaining: 1m 21s
214:    learn: 0.0078146        total: 1m 1s    remaining: 1m 21s
215:    learn: 0.0077939        total: 1m 1s    remaining: 1m 21s
216:    learn: 0.0077285        total: 1m 2s    remaining: 1m 21s
217:    learn: 0.0076981        total: 1m 2s    remaining: 1m 20s
218:    learn: 0.0076485        total: 1m 2s    remaining: 1m 20s
219:    learn: 0.0076217        total: 1m 3s    remaining: 1m 20s
220:    learn: 0.0075889        total: 1m 3s    remaining: 1m 20s
221:    learn: 0.0075379        total: 1m 3s    remaining: 1m 19s
222:    learn: 0.0075234        total: 1m 3s    remaining: 1m 19s
223:    learn: 0.0075110        total: 1m 4s    remaining: 1m 19s
224:    learn: 0.0074845        total: 1m 4s    remaining: 1m 18s
225:    learn: 0.0074489        total: 1m 4s    remaining: 1m 18s
226:    learn: 0.0073931        total: 1m 5s    remaining: 1m 18s
227:    learn: 0.0073789        total: 1m 5s    remaining: 1m 17s
228:    learn: 0.0073492        total: 1m 5s    remaining: 1m 17s
229:    learn: 0.0072960        total: 1m 5s    remaining: 1m 17s
230:    learn: 0.0072362        total: 1m 6s    remaining: 1m 17s
231:    learn: 0.0072005        total: 1m 6s    remaining: 1m 16s
232:    learn: 0.0071764        total: 1m 6s    remaining: 1m 16s
233:    learn: 0.0071502        total: 1m 7s    remaining: 1m 16s
234:    learn: 0.0071027        total: 1m 7s    remaining: 1m 15s
235:    learn: 0.0070824        total: 1m 7s    remaining: 1m 15s
236:    learn: 0.0070706        total: 1m 7s    remaining: 1m 15s
237:    learn: 0.0070284        total: 1m 8s    remaining: 1m 15s
238:    learn: 0.0070024        total: 1m 8s    remaining: 1m 14s
239:    learn: 0.0069542        total: 1m 8s    remaining: 1m 14s
240:    learn: 0.0069300        total: 1m 9s    remaining: 1m 14s
241:    learn: 0.0069067        total: 1m 9s    remaining: 1m 13s
242:    learn: 0.0068755        total: 1m 9s    remaining: 1m 13s
243:    learn: 0.0068585        total: 1m 9s    remaining: 1m 13s
244:    learn: 0.0068423        total: 1m 10s   remaining: 1m 13s
245:    learn: 0.0068061        total: 1m 10s   remaining: 1m 12s
246:    learn: 0.0067760        total: 1m 10s   remaining: 1m 12s
247:    learn: 0.0067497        total: 1m 11s   remaining: 1m 12s
248:    learn: 0.0067396        total: 1m 11s   remaining: 1m 11s
249:    learn: 0.0067132        total: 1m 11s   remaining: 1m 11s
250:    learn: 0.0067059        total: 1m 11s   remaining: 1m 11s
251:    learn: 0.0066759        total: 1m 12s   remaining: 1m 11s
252:    learn: 0.0066381        total: 1m 12s   remaining: 1m 10s
253:    learn: 0.0066051        total: 1m 12s   remaining: 1m 10s
254:    learn: 0.0065671        total: 1m 13s   remaining: 1m 10s
255:    learn: 0.0065396        total: 1m 13s   remaining: 1m 9s
256:    learn: 0.0065271        total: 1m 13s   remaining: 1m 9s
257:    learn: 0.0064840        total: 1m 13s   remaining: 1m 9s
258:    learn: 0.0064686        total: 1m 14s   remaining: 1m 9s
```

```
259:    learn: 0.0064187        total: 1m 14s   remaining: 1m 8s
260:    learn: 0.0063917        total: 1m 14s   remaining: 1m 8s
261:    learn: 0.0063622        total: 1m 15s   remaining: 1m 8s
262:    learn: 0.0063378        total: 1m 15s   remaining: 1m 7s
263:    learn: 0.0063104        total: 1m 15s   remaining: 1m 7s
264:    learn: 0.0063001        total: 1m 15s   remaining: 1m 7s
265:    learn: 0.0062824        total: 1m 16s   remaining: 1m 7s
266:    learn: 0.0062499        total: 1m 16s   remaining: 1m 6s
267:    learn: 0.0062173        total: 1m 17s   remaining: 1m 6s
268:    learn: 0.0061948        total: 1m 17s   remaining: 1m 6s
269:    learn: 0.0061520        total: 1m 17s   remaining: 1m 6s
270:    learn: 0.0061235        total: 1m 18s   remaining: 1m 6s
271:    learn: 0.0060938        total: 1m 18s   remaining: 1m 6s
272:    learn: 0.0060712        total: 1m 19s   remaining: 1m 5s
273:    learn: 0.0060648        total: 1m 19s   remaining: 1m 5s
274:    learn: 0.0060352        total: 1m 19s   remaining: 1m 5s
275:    learn: 0.0060119        total: 1m 19s   remaining: 1m 4s
276:    learn: 0.0059874        total: 1m 20s   remaining: 1m 4s
277:    learn: 0.0059670        total: 1m 20s   remaining: 1m 4s
278:    learn: 0.0059227        total: 1m 20s   remaining: 1m 4s
279:    learn: 0.0059037        total: 1m 21s   remaining: 1m 3s
280:    learn: 0.0058760        total: 1m 21s   remaining: 1m 3s
281:    learn: 0.0058456        total: 1m 21s   remaining: 1m 3s
282:    learn: 0.0058125        total: 1m 22s   remaining: 1m 2s
283:    learn: 0.0057643        total: 1m 22s   remaining: 1m 2s
284:    learn: 0.0057334        total: 1m 22s   remaining: 1m 2s
285:    learn: 0.0057160        total: 1m 22s   remaining: 1m 1s
286:    learn: 0.0056932        total: 1m 23s   remaining: 1m 1s
287:    learn: 0.0056663        total: 1m 23s   remaining: 1m 1s
288:    learn: 0.0056509        total: 1m 23s   remaining: 1m 1s
289:    learn: 0.0056266        total: 1m 24s   remaining: 1m
290:    learn: 0.0055931        total: 1m 24s   remaining: 1m
291:    learn: 0.0055735        total: 1m 24s   remaining: 1m
292:    learn: 0.0055608        total: 1m 24s   remaining: 60s
293:    learn: 0.0055512        total: 1m 25s   remaining: 59.7s
294:    learn: 0.0055391        total: 1m 25s   remaining: 59.4s
295:    learn: 0.0055059        total: 1m 25s   remaining: 59.1s
296:    learn: 0.0054827        total: 1m 26s   remaining: 58.8s
297:    learn: 0.0054707        total: 1m 26s   remaining: 58.5s
298:    learn: 0.0054558        total: 1m 26s   remaining: 58.2s
299:    learn: 0.0054374        total: 1m 26s   remaining: 57.9s
300:    learn: 0.0054116        total: 1m 27s   remaining: 57.6s
301:    learn: 0.0053939        total: 1m 27s   remaining: 57.3s
302:    learn: 0.0053829        total: 1m 27s   remaining: 57s
303:    learn: 0.0053581        total: 1m 27s   remaining: 56.7s
304:    learn: 0.0053427        total: 1m 28s   remaining: 56.4s
305:    learn: 0.0053333        total: 1m 28s   remaining: 56.1s
306:    learn: 0.0053059        total: 1m 28s   remaining: 55.8s
```

```
307:    learn: 0.0052926        total: 1m 29s   remaining: 55.5s
308:    learn: 0.0052775        total: 1m 29s   remaining: 55.3s
309:    learn: 0.0052721        total: 1m 29s   remaining: 55s
310:    learn: 0.0052628        total: 1m 30s   remaining: 54.7s
311:    learn: 0.0052485        total: 1m 30s   remaining: 54.4s
312:    learn: 0.0052318        total: 1m 30s   remaining: 54.1s
313:    learn: 0.0052090        total: 1m 30s   remaining: 53.8s
314:    learn: 0.0051850        total: 1m 31s   remaining: 53.5s
315:    learn: 0.0051436        total: 1m 31s   remaining: 53.2s
316:    learn: 0.0051268        total: 1m 31s   remaining: 52.9s
317:    learn: 0.0051093        total: 1m 31s   remaining: 52.6s
318:    learn: 0.0050832        total: 1m 32s   remaining: 52.3s
319:    learn: 0.0050651        total: 1m 32s   remaining: 52s
320:    learn: 0.0050449        total: 1m 32s   remaining: 51.8s
321:    learn: 0.0050065        total: 1m 33s   remaining: 51.4s
322:    learn: 0.0049842        total: 1m 33s   remaining: 51.1s
323:    learn: 0.0049635        total: 1m 33s   remaining: 50.8s
324:    learn: 0.0049488        total: 1m 33s   remaining: 50.5s
325:    learn: 0.0049437        total: 1m 34s   remaining: 50.2s
326:    learn: 0.0049245        total: 1m 34s   remaining: 50s
327:    learn: 0.0049038        total: 1m 34s   remaining: 49.7s
328:    learn: 0.0048933        total: 1m 35s   remaining: 49.4s
329:    learn: 0.0048779        total: 1m 35s   remaining: 49.1s
330:    learn: 0.0048669        total: 1m 35s   remaining: 48.8s
331:    learn: 0.0048389        total: 1m 35s   remaining: 48.5s
332:    learn: 0.0048297        total: 1m 36s   remaining: 48.2s
333:    learn: 0.0048088        total: 1m 36s   remaining: 48s
334:    learn: 0.0047943        total: 1m 36s   remaining: 47.7s
335:    learn: 0.0047718        total: 1m 37s   remaining: 47.4s
336:    learn: 0.0047540        total: 1m 37s   remaining: 47.1s
337:    learn: 0.0047368        total: 1m 37s   remaining: 46.8s
338:    learn: 0.0047191        total: 1m 37s   remaining: 46.5s
339:    learn: 0.0046937        total: 1m 38s   remaining: 46.2s
340:    learn: 0.0046568        total: 1m 38s   remaining: 46s
341:    learn: 0.0046450        total: 1m 38s   remaining: 45.7s
342:    learn: 0.0046301        total: 1m 39s   remaining: 45.4s
343:    learn: 0.0046096        total: 1m 39s   remaining: 45.1s
344:    learn: 0.0046013        total: 1m 39s   remaining: 44.8s
345:    learn: 0.0045834        total: 1m 40s   remaining: 44.5s
346:    learn: 0.0045723        total: 1m 40s   remaining: 44.2s
347:    learn: 0.0045539        total: 1m 40s   remaining: 43.9s
348:    learn: 0.0045420        total: 1m 40s   remaining: 43.6s
349:    learn: 0.0045311        total: 1m 41s   remaining: 43.3s
350:    learn: 0.0045158        total: 1m 41s   remaining: 43s
351:    learn: 0.0045011        total: 1m 41s   remaining: 42.8s
352:    learn: 0.0044793        total: 1m 41s   remaining: 42.5s
353:    learn: 0.0044616        total: 1m 42s   remaining: 42.2s
354:    learn: 0.0044482        total: 1m 42s   remaining: 41.9s
```

```
355:    learn: 0.0044250        total: 1m 42s   remaining: 41.6s
356:    learn: 0.0044151        total: 1m 43s   remaining: 41.3s
357:    learn: 0.0044028        total: 1m 43s   remaining: 41s
358:    learn: 0.0043828        total: 1m 43s   remaining: 40.7s
359:    learn: 0.0043722        total: 1m 43s   remaining: 40.4s
360:    learn: 0.0043571        total: 1m 44s   remaining: 40.1s
361:    learn: 0.0043373        total: 1m 44s   remaining: 39.8s
362:    learn: 0.0043283        total: 1m 44s   remaining: 39.5s
363:    learn: 0.0043249        total: 1m 44s   remaining: 39.2s
364:    learn: 0.0043173        total: 1m 45s   remaining: 38.9s
365:    learn: 0.0043038        total: 1m 45s   remaining: 38.7s
366:    learn: 0.0042892        total: 1m 45s   remaining: 38.4s
367:    learn: 0.0042588        total: 1m 46s   remaining: 38.1s
368:    learn: 0.0042493        total: 1m 46s   remaining: 37.8s
369:    learn: 0.0042294        total: 1m 46s   remaining: 37.5s
370:    learn: 0.0042195        total: 1m 47s   remaining: 37.2s
371:    learn: 0.0041968        total: 1m 47s   remaining: 36.9s
372:    learn: 0.0041833        total: 1m 47s   remaining: 36.7s
373:    learn: 0.0041729        total: 1m 48s   remaining: 36.4s
374:    learn: 0.0041541        total: 1m 48s   remaining: 36.1s
375:    learn: 0.0041429        total: 1m 48s   remaining: 35.8s
376:    learn: 0.0041244        total: 1m 48s   remaining: 35.5s
377:    learn: 0.0041003        total: 1m 49s   remaining: 35.2s
378:    learn: 0.0040877        total: 1m 49s   remaining: 34.9s
379:    learn: 0.0040669        total: 1m 49s   remaining: 34.6s
380:    learn: 0.0040375        total: 1m 49s   remaining: 34.3s
381:    learn: 0.0040318        total: 1m 50s   remaining: 34.1s
382:    learn: 0.0040232        total: 1m 50s   remaining: 33.8s
383:    learn: 0.0040107        total: 1m 50s   remaining: 33.5s
384:    learn: 0.0039970        total: 1m 51s   remaining: 33.2s
385:    learn: 0.0039893        total: 1m 51s   remaining: 32.9s
386:    learn: 0.0039734        total: 1m 51s   remaining: 32.6s
387:    learn: 0.0039632        total: 1m 51s   remaining: 32.3s
388:    learn: 0.0039569        total: 1m 52s   remaining: 32s
389:    learn: 0.0039495        total: 1m 52s   remaining: 31.8s
390:    learn: 0.0039359        total: 1m 52s   remaining: 31.5s
391:    learn: 0.0039276        total: 1m 53s   remaining: 31.2s
392:    learn: 0.0039117        total: 1m 53s   remaining: 30.9s
393:    learn: 0.0039025        total: 1m 53s   remaining: 30.6s
394:    learn: 0.0038807        total: 1m 53s   remaining: 30.3s
395:    learn: 0.0038581        total: 1m 54s   remaining: 30s
396:    learn: 0.0038427        total: 1m 54s   remaining: 29.7s
397:    learn: 0.0038195        total: 1m 54s   remaining: 29.4s
398:    learn: 0.0038104        total: 1m 55s   remaining: 29.2s
399:    learn: 0.0037900        total: 1m 55s   remaining: 28.9s
400:    learn: 0.0037828        total: 1m 55s   remaining: 28.6s
401:    learn: 0.0037663        total: 1m 56s   remaining: 28.3s
402:    learn: 0.0037375        total: 1m 56s   remaining: 28s
```

```
403:    learn: 0.0037315        total: 1m 56s   remaining: 27.8s
404:    learn: 0.0037167        total: 1m 57s   remaining: 27.5s
405:    learn: 0.0037043        total: 1m 57s   remaining: 27.2s
406:    learn: 0.0036874        total: 1m 58s   remaining: 27s
407:    learn: 0.0036768        total: 1m 58s   remaining: 26.7s
408:    learn: 0.0036665        total: 1m 58s   remaining: 26.4s
409:    learn: 0.0036598        total: 1m 59s   remaining: 26.2s
410:    learn: 0.0036466        total: 1m 59s   remaining: 25.9s
411:    learn: 0.0036300        total: 2m       remaining: 25.7s
412:    learn: 0.0035992        total: 2m       remaining: 25.4s
413:    learn: 0.0035924        total: 2m       remaining: 25.1s
414:    learn: 0.0035835        total: 2m 1s    remaining: 24.8s
415:    learn: 0.0035721        total: 2m 1s    remaining: 24.5s
416:    learn: 0.0035679        total: 2m 1s    remaining: 24.2s
417:    learn: 0.0035585        total: 2m 1s    remaining: 23.9s
418:    learn: 0.0035462        total: 2m 2s    remaining: 23.6s
419:    learn: 0.0035384        total: 2m 2s    remaining: 23.3s
420:    learn: 0.0035311        total: 2m 2s    remaining: 23.1s
421:    learn: 0.0035215        total: 2m 3s    remaining: 22.8s
422:    learn: 0.0035032        total: 2m 3s    remaining: 22.5s
423:    learn: 0.0034884        total: 2m 3s    remaining: 22.2s
424:    learn: 0.0034729        total: 2m 4s    remaining: 21.9s
425:    learn: 0.0034649        total: 2m 4s    remaining: 21.6s
426:    learn: 0.0034527        total: 2m 4s    remaining: 21.3s
427:    learn: 0.0034445        total: 2m 4s    remaining: 21s
428:    learn: 0.0034300        total: 2m 5s    remaining: 20.7s
429:    learn: 0.0034168        total: 2m 5s    remaining: 20.4s
430:    learn: 0.0033967        total: 2m 5s    remaining: 20.1s
431:    learn: 0.0033837        total: 2m 6s    remaining: 19.8s
432:    learn: 0.0033728        total: 2m 6s    remaining: 19.6s
433:    learn: 0.0033627        total: 2m 6s    remaining: 19.3s
434:    learn: 0.0033561        total: 2m 7s    remaining: 19s
435:    learn: 0.0033442        total: 2m 7s    remaining: 18.7s
436:    learn: 0.0033378        total: 2m 7s    remaining: 18.4s
437:    learn: 0.0033274        total: 2m 7s    remaining: 18.1s
438:    learn: 0.0033157        total: 2m 8s    remaining: 17.8s
439:    learn: 0.0033062        total: 2m 8s    remaining: 17.5s
440:    learn: 0.0032941        total: 2m 8s    remaining: 17.2s
441:    learn: 0.0032814        total: 2m 9s    remaining: 16.9s
442:    learn: 0.0032707        total: 2m 9s    remaining: 16.6s
443:    learn: 0.0032541        total: 2m 9s    remaining: 16.4s
444:    learn: 0.0032427        total: 2m 9s    remaining: 16.1s
445:    learn: 0.0032351        total: 2m 10s   remaining: 15.8s
446:    learn: 0.0032260        total: 2m 10s   remaining: 15.5s
447:    learn: 0.0032186        total: 2m 10s   remaining: 15.2s
448:    learn: 0.0032121        total: 2m 11s   remaining: 14.9s
449:    learn: 0.0031929        total: 2m 11s   remaining: 14.6s
450:    learn: 0.0031767        total: 2m 11s   remaining: 14.3s
```

```
451:    learn: 0.0031657        total: 2m 12s   remaining: 14s
452:    learn: 0.0031592        total: 2m 12s   remaining: 13.7s
453:    learn: 0.0031478        total: 2m 12s   remaining: 13.4s
454:    learn: 0.0031390        total: 2m 12s   remaining: 13.1s
455:    learn: 0.0031384        total: 2m 13s   remaining: 12.9s
456:    learn: 0.0031326        total: 2m 13s   remaining: 12.6s
457:    learn: 0.0031220        total: 2m 13s   remaining: 12.3s
458:    learn: 0.0031138        total: 2m 14s   remaining: 12s
459:    learn: 0.0031044        total: 2m 14s   remaining: 11.7s
460:    learn: 0.0030931        total: 2m 14s   remaining: 11.4s
461:    learn: 0.0030815        total: 2m 14s   remaining: 11.1s
462:    learn: 0.0030734        total: 2m 15s   remaining: 10.8s
463:    learn: 0.0030656        total: 2m 15s   remaining: 10.5s
464:    learn: 0.0030595        total: 2m 15s   remaining: 10.2s
465:    learn: 0.0030518        total: 2m 16s   remaining: 9.93s
466:    learn: 0.0030391        total: 2m 16s   remaining: 9.63s
467:    learn: 0.0030268        total: 2m 16s   remaining: 9.34s
468:    learn: 0.0030185        total: 2m 16s   remaining: 9.05s
469:    learn: 0.0030123        total: 2m 17s   remaining: 8.76s
470:    learn: 0.0030002        total: 2m 17s   remaining: 8.46s
471:    learn: 0.0029973        total: 2m 17s   remaining: 8.17s
472:    learn: 0.0029910        total: 2m 18s   remaining: 7.88s
473:    learn: 0.0029772        total: 2m 18s   remaining: 7.59s
474:    learn: 0.0029656        total: 2m 18s   remaining: 7.3s
475:    learn: 0.0029567        total: 2m 18s   remaining: 7s
476:    learn: 0.0029469        total: 2m 19s   remaining: 6.71s
477:    learn: 0.0029359        total: 2m 19s   remaining: 6.42s
478:    learn: 0.0029286        total: 2m 19s   remaining: 6.13s
479:    learn: 0.0029184        total: 2m 20s   remaining: 5.84s
480:    learn: 0.0029146        total: 2m 20s   remaining: 5.55s
481:    learn: 0.0029048        total: 2m 20s   remaining: 5.25s
482:    learn: 0.0028956        total: 2m 20s   remaining: 4.96s
483:    learn: 0.0028857        total: 2m 21s   remaining: 4.67s
484:    learn: 0.0028769        total: 2m 21s   remaining: 4.38s
485:    learn: 0.0028649        total: 2m 21s   remaining: 4.09s
486:    learn: 0.0028486        total: 2m 22s   remaining: 3.79s
487:    learn: 0.0028468        total: 2m 22s   remaining: 3.5s
488:    learn: 0.0028364        total: 2m 22s   remaining: 3.21s
489:    learn: 0.0028262        total: 2m 23s   remaining: 2.92s
490:    learn: 0.0028169        total: 2m 23s   remaining: 2.63s
491:    learn: 0.0028069        total: 2m 23s   remaining: 2.33s
492:    learn: 0.0027995        total: 2m 23s   remaining: 2.04s
493:    learn: 0.0027914        total: 2m 24s   remaining: 1.75s
494:    learn: 0.0027893        total: 2m 24s   remaining: 1.46s
495:    learn: 0.0027824        total: 2m 24s   remaining: 1.17s
496:    learn: 0.0027774        total: 2m 25s   remaining: 876ms
497:    learn: 0.0027669        total: 2m 25s   remaining: 584ms
498:    learn: 0.0027608        total: 2m 25s   remaining: 292ms
```

```
499:    learn: 0.0027515        total: 2m 26s   remaining: 0us
0:      learn: 0.9601132        total: 195ms    remaining: 1m 37s
1:      learn: 0.8451954        total: 491ms    remaining: 2m 2s
2:      learn: 0.7578077        total: 772ms    remaining: 2m 7s
3:      learn: 0.6888458        total: 1.03s    remaining: 2m 7s
4:      learn: 0.6115160        total: 1.33s    remaining: 2m 11s
5:      learn: 0.5475679        total: 1.67s    remaining: 2m 17s
6:      learn: 0.4938687        total: 1.98s    remaining: 2m 19s
7:      learn: 0.4485845        total: 2.26s    remaining: 2m 19s
8:      learn: 0.4083268        total: 2.63s    remaining: 2m 23s
9:      learn: 0.3747409        total: 2.95s    remaining: 2m 24s
10:     learn: 0.3383141        total: 3.2s     remaining: 2m 22s
11:     learn: 0.3062419        total: 3.45s    remaining: 2m 20s
12:     learn: 0.2786205        total: 3.69s    remaining: 2m 18s
13:     learn: 0.2539455        total: 3.96s    remaining: 2m 17s
14:     learn: 0.2314995        total: 4.22s    remaining: 2m 16s
15:     learn: 0.2119706        total: 4.5s     remaining: 2m 16s
16:     learn: 0.1941337        total: 4.75s    remaining: 2m 15s
17:     learn: 0.1788101        total: 5.02s    remaining: 2m 14s
18:     learn: 0.1646987        total: 5.27s    remaining: 2m 13s
19:     learn: 0.1527847        total: 5.54s    remaining: 2m 12s
20:     learn: 0.1414824        total: 5.79s    remaining: 2m 12s
21:     learn: 0.1316765        total: 6.04s    remaining: 2m 11s
22:     learn: 0.1228930        total: 6.3s     remaining: 2m 10s
23:     learn: 0.1147616        total: 6.55s    remaining: 2m 9s
24:     learn: 0.1067771        total: 6.81s    remaining: 2m 9s
25:     learn: 0.0997815        total: 7.08s    remaining: 2m 9s
26:     learn: 0.0936968        total: 7.37s    remaining: 2m 9s
27:     learn: 0.0884273        total: 7.62s    remaining: 2m 8s
28:     learn: 0.0835542        total: 7.89s    remaining: 2m 8s
29:     learn: 0.0789537        total: 8.19s    remaining: 2m 8s
30:     learn: 0.0750107        total: 8.5s     remaining: 2m 8s
31:     learn: 0.0713494        total: 8.77s    remaining: 2m 8s
32:     learn: 0.0676655        total: 9.09s    remaining: 2m 8s
33:     learn: 0.0642806        total: 9.39s    remaining: 2m 8s
34:     learn: 0.0609035        total: 9.71s    remaining: 2m 8s
35:     learn: 0.0578486        total: 9.97s    remaining: 2m 8s
36:     learn: 0.0554666        total: 10.3s    remaining: 2m 9s
37:     learn: 0.0527044        total: 10.6s    remaining: 2m 8s
38:     learn: 0.0504867        total: 10.9s    remaining: 2m 8s
39:     learn: 0.0482854        total: 11.2s    remaining: 2m 8s
40:     learn: 0.0463452        total: 11.4s    remaining: 2m 8s
41:     learn: 0.0447648        total: 11.7s    remaining: 2m 8s
42:     learn: 0.0429734        total: 12s      remaining: 2m 7s
43:     learn: 0.0413554        total: 12.3s    remaining: 2m 7s
44:     learn: 0.0396337        total: 12.5s    remaining: 2m 6s
45:     learn: 0.0381125        total: 12.9s    remaining: 2m 6s
46:     learn: 0.0367768        total: 13.2s    remaining: 2m 7s
```

```
47:     learn: 0.0356922        total: 13.5s    remaining: 2m 7s
48:     learn: 0.0344944        total: 13.8s    remaining: 2m 7s
49:     learn: 0.0336007        total: 14.1s    remaining: 2m 7s
50:     learn: 0.0326401        total: 14.5s    remaining: 2m 7s
51:     learn: 0.0316472        total: 14.7s    remaining: 2m 6s
52:     learn: 0.0309255        total: 15s      remaining: 2m 6s
53:     learn: 0.0300981        total: 15.3s    remaining: 2m 6s
54:     learn: 0.0293449        total: 15.6s    remaining: 2m 6s
55:     learn: 0.0285155        total: 15.9s    remaining: 2m 5s
56:     learn: 0.0278349        total: 16.2s    remaining: 2m 5s
57:     learn: 0.0271400        total: 16.5s    remaining: 2m 5s
58:     learn: 0.0264923        total: 16.8s    remaining: 2m 5s
59:     learn: 0.0257404        total: 17.1s    remaining: 2m 5s
60:     learn: 0.0252407        total: 17.4s    remaining: 2m 4s
61:     learn: 0.0246806        total: 17.7s    remaining: 2m 4s
62:     learn: 0.0244005        total: 17.9s    remaining: 2m 4s
63:     learn: 0.0238955        total: 18.3s    remaining: 2m 4s
64:     learn: 0.0233610        total: 18.5s    remaining: 2m 4s
65:     learn: 0.0230702        total: 18.8s    remaining: 2m 3s
66:     learn: 0.0227118        total: 19.1s    remaining: 2m 3s
67:     learn: 0.0224304        total: 19.4s    remaining: 2m 3s
68:     learn: 0.0220689        total: 19.7s    remaining: 2m 3s
69:     learn: 0.0217856        total: 20s      remaining: 2m 3s
70:     learn: 0.0216347        total: 20.3s    remaining: 2m 2s
71:     learn: 0.0212760        total: 20.6s    remaining: 2m 2s
72:     learn: 0.0211052        total: 20.9s    remaining: 2m 2s
73:     learn: 0.0209122        total: 21.2s    remaining: 2m 2s
74:     learn: 0.0207780        total: 21.5s    remaining: 2m 1s
75:     learn: 0.0204701        total: 21.8s    remaining: 2m 1s
76:     learn: 0.0202995        total: 22s      remaining: 2m 1s
77:     learn: 0.0200337        total: 22.3s    remaining: 2m
78:     learn: 0.0197890        total: 22.6s    remaining: 2m
79:     learn: 0.0196229        total: 22.9s    remaining: 2m
80:     learn: 0.0193478        total: 23.2s    remaining: 1m 59s
81:     learn: 0.0189253        total: 23.5s    remaining: 1m 59s
82:     learn: 0.0186352        total: 23.8s    remaining: 1m 59s
83:     learn: 0.0185539        total: 24.1s    remaining: 1m 59s
84:     learn: 0.0183619        total: 24.3s    remaining: 1m 58s
85:     learn: 0.0182171        total: 24.6s    remaining: 1m 58s
86:     learn: 0.0180129        total: 24.9s    remaining: 1m 58s
87:     learn: 0.0179393        total: 25.2s    remaining: 1m 58s
88:     learn: 0.0177930        total: 25.5s    remaining: 1m 57s
89:     learn: 0.0177549        total: 25.8s    remaining: 1m 57s
90:     learn: 0.0175574        total: 26.1s    remaining: 1m 57s
91:     learn: 0.0174716        total: 26.4s    remaining: 1m 56s
92:     learn: 0.0171213        total: 26.7s    remaining: 1m 56s
93:     learn: 0.0169672        total: 26.9s    remaining: 1m 56s
94:     learn: 0.0168028        total: 27.2s    remaining: 1m 56s
```

```
95:     learn: 0.0166866        total: 27.5s    remaining: 1m 55s
96:     learn: 0.0165168        total: 27.8s    remaining: 1m 55s
97:     learn: 0.0162851        total: 28.1s    remaining: 1m 55s
98:     learn: 0.0161741        total: 28.4s    remaining: 1m 55s
99:     learn: 0.0160873        total: 28.7s    remaining: 1m 54s
100:    learn: 0.0159120        total: 29s      remaining: 1m 54s
101:    learn: 0.0157693        total: 29.3s    remaining: 1m 54s
102:    learn: 0.0156422        total: 29.6s    remaining: 1m 54s
103:    learn: 0.0155410        total: 29.9s    remaining: 1m 53s
104:    learn: 0.0154817        total: 30.2s    remaining: 1m 53s
105:    learn: 0.0153097        total: 30.5s    remaining: 1m 53s
106:    learn: 0.0152230        total: 30.8s    remaining: 1m 52s
107:    learn: 0.0151075        total: 31.1s    remaining: 1m 52s
108:    learn: 0.0151002        total: 31.2s    remaining: 1m 51s
109:    learn: 0.0149730        total: 31.5s    remaining: 1m 51s
110:    learn: 0.0147706        total: 31.8s    remaining: 1m 51s
111:    learn: 0.0145673        total: 32s      remaining: 1m 50s
112:    learn: 0.0144879        total: 32.3s    remaining: 1m 50s
113:    learn: 0.0144087        total: 32.6s    remaining: 1m 50s
114:    learn: 0.0143155        total: 32.9s    remaining: 1m 50s
115:    learn: 0.0142357        total: 33.2s    remaining: 1m 49s
116:    learn: 0.0141741        total: 33.5s    remaining: 1m 49s
117:    learn: 0.0141103        total: 33.8s    remaining: 1m 49s
118:    learn: 0.0140570        total: 34.1s    remaining: 1m 49s
119:    learn: 0.0139682        total: 34.4s    remaining: 1m 48s
120:    learn: 0.0138697        total: 34.7s    remaining: 1m 48s
121:    learn: 0.0138036        total: 35s      remaining: 1m 48s
122:    learn: 0.0137424        total: 35.3s    remaining: 1m 48s
123:    learn: 0.0136708        total: 35.6s    remaining: 1m 47s
124:    learn: 0.0136213        total: 35.9s    remaining: 1m 47s
125:    learn: 0.0134885        total: 36.2s    remaining: 1m 47s
126:    learn: 0.0134258        total: 36.4s    remaining: 1m 47s
127:    learn: 0.0133385        total: 36.6s    remaining: 1m 46s
128:    learn: 0.0132484        total: 36.9s    remaining: 1m 46s
129:    learn: 0.0131078        total: 37.3s    remaining: 1m 46s
130:    learn: 0.0130246        total: 37.6s    remaining: 1m 45s
131:    learn: 0.0129756        total: 37.9s    remaining: 1m 45s
132:    learn: 0.0128909        total: 38.1s    remaining: 1m 45s
133:    learn: 0.0128160        total: 38.4s    remaining: 1m 45s
134:    learn: 0.0127646        total: 38.7s    remaining: 1m 44s
135:    learn: 0.0126925        total: 39s      remaining: 1m 44s
136:    learn: 0.0126540        total: 39.3s    remaining: 1m 44s
137:    learn: 0.0126067        total: 39.6s    remaining: 1m 43s
138:    learn: 0.0125366        total: 39.9s    remaining: 1m 43s
139:    learn: 0.0124984        total: 40.2s    remaining: 1m 43s
140:    learn: 0.0124006        total: 40.5s    remaining: 1m 43s
141:    learn: 0.0123462        total: 40.8s    remaining: 1m 42s
142:    learn: 0.0123021        total: 41.1s    remaining: 1m 42s
```

```
143:    learn: 0.0122139        total: 41.4s    remaining: 1m 42s
144:    learn: 0.0121608        total: 41.7s    remaining: 1m 41s
145:    learn: 0.0120535        total: 41.9s    remaining: 1m 41s
146:    learn: 0.0119846        total: 42.2s    remaining: 1m 41s
147:    learn: 0.0119093        total: 42.5s    remaining: 1m 41s
148:    learn: 0.0118223        total: 42.8s    remaining: 1m 40s
149:    learn: 0.0117290        total: 43.1s    remaining: 1m 40s
150:    learn: 0.0117219        total: 43.4s    remaining: 1m 40s
151:    learn: 0.0116800        total: 43.8s    remaining: 1m 40s
152:    learn: 0.0115794        total: 44.1s    remaining: 1m 40s
153:    learn: 0.0115301        total: 44.4s    remaining: 1m 39s
154:    learn: 0.0114948        total: 44.7s    remaining: 1m 39s
155:    learn: 0.0114337        total: 45s      remaining: 1m 39s
156:    learn: 0.0113425        total: 45.2s    remaining: 1m 38s
157:    learn: 0.0112895        total: 45.5s    remaining: 1m 38s
158:    learn: 0.0112158        total: 45.8s    remaining: 1m 38s
159:    learn: 0.0111546        total: 46s      remaining: 1m 37s
160:    learn: 0.0110885        total: 46.4s    remaining: 1m 37s
161:    learn: 0.0110349        total: 46.7s    remaining: 1m 37s
162:    learn: 0.0109937        total: 46.9s    remaining: 1m 37s
163:    learn: 0.0109576        total: 47.2s    remaining: 1m 36s
164:    learn: 0.0108981        total: 47.5s    remaining: 1m 36s
165:    learn: 0.0108566        total: 47.8s    remaining: 1m 36s
166:    learn: 0.0107460        total: 48.1s    remaining: 1m 35s
167:    learn: 0.0106639        total: 48.4s    remaining: 1m 35s
168:    learn: 0.0105853        total: 48.7s    remaining: 1m 35s
169:    learn: 0.0105212        total: 49.1s    remaining: 1m 35s
170:    learn: 0.0104748        total: 49.4s    remaining: 1m 34s
171:    learn: 0.0103642        total: 49.6s    remaining: 1m 34s
172:    learn: 0.0103197        total: 49.9s    remaining: 1m 34s
173:    learn: 0.0102342        total: 50.2s    remaining: 1m 34s
174:    learn: 0.0101657        total: 50.5s    remaining: 1m 33s
175:    learn: 0.0101063        total: 50.8s    remaining: 1m 33s
176:    learn: 0.0100845        total: 51.1s    remaining: 1m 33s
177:    learn: 0.0100234        total: 51.4s    remaining: 1m 32s
178:    learn: 0.0099658        total: 51.7s    remaining: 1m 32s
179:    learn: 0.0099311        total: 52s      remaining: 1m 32s
180:    learn: 0.0098840        total: 52.4s    remaining: 1m 32s
181:    learn: 0.0098347        total: 52.7s    remaining: 1m 32s
182:    learn: 0.0097813        total: 53s      remaining: 1m 31s
183:    learn: 0.0097361        total: 53.3s    remaining: 1m 31s
184:    learn: 0.0096749        total: 53.6s    remaining: 1m 31s
185:    learn: 0.0096437        total: 53.9s    remaining: 1m 30s
186:    learn: 0.0095366        total: 54.2s    remaining: 1m 30s
187:    learn: 0.0094927        total: 54.5s    remaining: 1m 30s
188:    learn: 0.0094670        total: 54.8s    remaining: 1m 30s
189:    learn: 0.0093967        total: 55s      remaining: 1m 29s
190:    learn: 0.0093382        total: 55.3s    remaining: 1m 29s
```

```
191:    learn: 0.0092985        total: 55.6s    remaining: 1m 29s
192:    learn: 0.0092631        total: 55.9s    remaining: 1m 28s
193:    learn: 0.0092238        total: 56.2s    remaining: 1m 28s
194:    learn: 0.0091668        total: 56.5s    remaining: 1m 28s
195:    learn: 0.0091232        total: 56.8s    remaining: 1m 28s
196:    learn: 0.0090795        total: 57.1s    remaining: 1m 27s
197:    learn: 0.0090344        total: 57.4s    remaining: 1m 27s
198:    learn: 0.0089937        total: 57.7s    remaining: 1m 27s
199:    learn: 0.0089667        total: 58s      remaining: 1m 26s
200:    learn: 0.0088863        total: 58.3s    remaining: 1m 26s
201:    learn: 0.0088353        total: 58.6s    remaining: 1m 26s
202:    learn: 0.0088066        total: 58.9s    remaining: 1m 26s
203:    learn: 0.0087390        total: 59.2s    remaining: 1m 25s
204:    learn: 0.0086701        total: 59.4s    remaining: 1m 25s
205:    learn: 0.0086306        total: 59.7s    remaining: 1m 25s
206:    learn: 0.0085822        total: 1m       remaining: 1m 24s
207:    learn: 0.0085098        total: 1m       remaining: 1m 24s
208:    learn: 0.0084629        total: 1m       remaining: 1m 24s
209:    learn: 0.0083854        total: 1m       remaining: 1m 24s
210:    learn: 0.0083363        total: 1m 1s    remaining: 1m 23s
211:    learn: 0.0083046        total: 1m 1s    remaining: 1m 23s
212:    learn: 0.0082730        total: 1m 1s    remaining: 1m 23s
213:    learn: 0.0082258        total: 1m 2s    remaining: 1m 22s
214:    learn: 0.0081844        total: 1m 2s    remaining: 1m 22s
215:    learn: 0.0080795        total: 1m 2s    remaining: 1m 22s
216:    learn: 0.0080419        total: 1m 2s    remaining: 1m 22s
217:    learn: 0.0080166        total: 1m 3s    remaining: 1m 21s
218:    learn: 0.0079865        total: 1m 3s    remaining: 1m 21s
219:    learn: 0.0079334        total: 1m 3s    remaining: 1m 21s
220:    learn: 0.0078941        total: 1m 4s    remaining: 1m 20s
221:    learn: 0.0078626        total: 1m 4s    remaining: 1m 20s
222:    learn: 0.0078178        total: 1m 4s    remaining: 1m 20s
223:    learn: 0.0077590        total: 1m 4s    remaining: 1m 20s
224:    learn: 0.0077222        total: 1m 5s    remaining: 1m 19s
225:    learn: 0.0076989        total: 1m 5s    remaining: 1m 19s
226:    learn: 0.0076635        total: 1m 5s    remaining: 1m 19s
227:    learn: 0.0076281        total: 1m 6s    remaining: 1m 18s
228:    learn: 0.0076049        total: 1m 6s    remaining: 1m 18s
229:    learn: 0.0075541        total: 1m 6s    remaining: 1m 18s
230:    learn: 0.0075186        total: 1m 6s    remaining: 1m 17s
231:    learn: 0.0074863        total: 1m 7s    remaining: 1m 17s
232:    learn: 0.0074542        total: 1m 7s    remaining: 1m 17s
233:    learn: 0.0074306        total: 1m 7s    remaining: 1m 17s
234:    learn: 0.0073998        total: 1m 8s    remaining: 1m 16s
235:    learn: 0.0073522        total: 1m 8s    remaining: 1m 16s
236:    learn: 0.0073096        total: 1m 8s    remaining: 1m 16s
237:    learn: 0.0072709        total: 1m 9s    remaining: 1m 15s
238:    learn: 0.0072586        total: 1m 9s    remaining: 1m 15s
```

```
239:    learn: 0.0072255        total: 1m 9s    remaining: 1m 15s
240:    learn: 0.0072072        total: 1m 9s    remaining: 1m 15s
241:    learn: 0.0071949        total: 1m 10s   remaining: 1m 14s
242:    learn: 0.0071510        total: 1m 10s   remaining: 1m 14s
243:    learn: 0.0071422        total: 1m 10s   remaining: 1m 14s
244:    learn: 0.0071063        total: 1m 11s   remaining: 1m 13s
245:    learn: 0.0070722        total: 1m 11s   remaining: 1m 13s
246:    learn: 0.0070587        total: 1m 11s   remaining: 1m 13s
247:    learn: 0.0070200        total: 1m 11s   remaining: 1m 13s
248:    learn: 0.0069975        total: 1m 12s   remaining: 1m 12s
249:    learn: 0.0069674        total: 1m 12s   remaining: 1m 12s
250:    learn: 0.0069254        total: 1m 12s   remaining: 1m 12s
251:    learn: 0.0068979        total: 1m 13s   remaining: 1m 11s
252:    learn: 0.0068238        total: 1m 13s   remaining: 1m 11s
253:    learn: 0.0068040        total: 1m 13s   remaining: 1m 11s
254:    learn: 0.0067895        total: 1m 13s   remaining: 1m 11s
255:    learn: 0.0067518        total: 1m 14s   remaining: 1m 10s
256:    learn: 0.0067224        total: 1m 14s   remaining: 1m 10s
257:    learn: 0.0066868        total: 1m 14s   remaining: 1m 10s
258:    learn: 0.0066334        total: 1m 15s   remaining: 1m 9s
259:    learn: 0.0066147        total: 1m 15s   remaining: 1m 9s
260:    learn: 0.0065881        total: 1m 15s   remaining: 1m 9s
261:    learn: 0.0065504        total: 1m 15s   remaining: 1m 8s
262:    learn: 0.0065294        total: 1m 16s   remaining: 1m 8s
263:    learn: 0.0065107        total: 1m 16s   remaining: 1m 8s
264:    learn: 0.0064909        total: 1m 16s   remaining: 1m 8s
265:    learn: 0.0064890        total: 1m 17s   remaining: 1m 7s
266:    learn: 0.0064684        total: 1m 17s   remaining: 1m 7s
267:    learn: 0.0064358        total: 1m 17s   remaining: 1m 7s
268:    learn: 0.0063928        total: 1m 18s   remaining: 1m 6s
269:    learn: 0.0063642        total: 1m 18s   remaining: 1m 6s
270:    learn: 0.0063359        total: 1m 18s   remaining: 1m 6s
271:    learn: 0.0063155        total: 1m 18s   remaining: 1m 6s
272:    learn: 0.0062797        total: 1m 19s   remaining: 1m 5s
273:    learn: 0.0062609        total: 1m 19s   remaining: 1m 5s
274:    learn: 0.0062249        total: 1m 19s   remaining: 1m 5s
275:    learn: 0.0062021        total: 1m 20s   remaining: 1m 5s
276:    learn: 0.0061915        total: 1m 20s   remaining: 1m 4s
277:    learn: 0.0061703        total: 1m 20s   remaining: 1m 4s
278:    learn: 0.0061544        total: 1m 21s   remaining: 1m 4s
279:    learn: 0.0061389        total: 1m 21s   remaining: 1m 3s
280:    learn: 0.0061185        total: 1m 21s   remaining: 1m 3s
281:    learn: 0.0061009        total: 1m 21s   remaining: 1m 3s
282:    learn: 0.0060724        total: 1m 22s   remaining: 1m 3s
283:    learn: 0.0060477        total: 1m 22s   remaining: 1m 2s
284:    learn: 0.0060250        total: 1m 22s   remaining: 1m 2s
285:    learn: 0.0059974        total: 1m 23s   remaining: 1m 2s
286:    learn: 0.0059739        total: 1m 23s   remaining: 1m 1s
```

```
287:    learn: 0.0059470        total: 1m 23s   remaining: 1m 1s
288:    learn: 0.0059299        total: 1m 23s   remaining: 1m 1s
289:    learn: 0.0059047        total: 1m 24s   remaining: 1m 1s
290:    learn: 0.0058884        total: 1m 24s   remaining: 1m
291:    learn: 0.0058604        total: 1m 24s   remaining: 1m
292:    learn: 0.0058374        total: 1m 25s   remaining: 1m
293:    learn: 0.0058041        total: 1m 25s   remaining: 59.9s
294:    learn: 0.0057753        total: 1m 25s   remaining: 59.6s
295:    learn: 0.0057441        total: 1m 26s   remaining: 59.3s
296:    learn: 0.0057205        total: 1m 26s   remaining: 59s
297:    learn: 0.0056896        total: 1m 26s   remaining: 58.8s
298:    learn: 0.0056728        total: 1m 26s   remaining: 58.5s
299:    learn: 0.0056625        total: 1m 27s   remaining: 58.1s
300:    learn: 0.0056346        total: 1m 27s   remaining: 57.9s
301:    learn: 0.0056321        total: 1m 27s   remaining: 57.5s
302:    learn: 0.0056138        total: 1m 28s   remaining: 57.3s
303:    learn: 0.0055793        total: 1m 28s   remaining: 57s
304:    learn: 0.0055558        total: 1m 28s   remaining: 56.7s
305:    learn: 0.0055408        total: 1m 28s   remaining: 56.4s
306:    learn: 0.0055202        total: 1m 29s   remaining: 56.1s
307:    learn: 0.0054929        total: 1m 29s   remaining: 55.9s
308:    learn: 0.0054789        total: 1m 29s   remaining: 55.6s
309:    learn: 0.0054613        total: 1m 30s   remaining: 55.3s
310:    learn: 0.0054351        total: 1m 30s   remaining: 55s
311:    learn: 0.0054078        total: 1m 30s   remaining: 54.8s
312:    learn: 0.0053825        total: 1m 31s   remaining: 54.5s
313:    learn: 0.0053605        total: 1m 31s   remaining: 54.2s
314:    learn: 0.0053559        total: 1m 31s   remaining: 54s
315:    learn: 0.0053268        total: 1m 32s   remaining: 53.8s
316:    learn: 0.0052947        total: 1m 32s   remaining: 53.5s
317:    learn: 0.0052757        total: 1m 33s   remaining: 53.2s
318:    learn: 0.0052649        total: 1m 33s   remaining: 53s
319:    learn: 0.0052480        total: 1m 33s   remaining: 52.7s
320:    learn: 0.0052348        total: 1m 33s   remaining: 52.4s
321:    learn: 0.0052154        total: 1m 34s   remaining: 52.1s
322:    learn: 0.0051989        total: 1m 34s   remaining: 51.8s
323:    learn: 0.0051816        total: 1m 34s   remaining: 51.6s
324:    learn: 0.0051668        total: 1m 35s   remaining: 51.3s
325:    learn: 0.0051466        total: 1m 35s   remaining: 51s
326:    learn: 0.0051304        total: 1m 35s   remaining: 50.6s
327:    learn: 0.0051194        total: 1m 36s   remaining: 50.3s
328:    learn: 0.0050987        total: 1m 36s   remaining: 50s
329:    learn: 0.0050711        total: 1m 36s   remaining: 49.7s
330:    learn: 0.0050524        total: 1m 36s   remaining: 49.4s
331:    learn: 0.0050204        total: 1m 37s   remaining: 49.1s
332:    learn: 0.0050123        total: 1m 37s   remaining: 48.8s
333:    learn: 0.0049925        total: 1m 37s   remaining: 48.5s
334:    learn: 0.0049816        total: 1m 37s   remaining: 48.2s
```

```
335:    learn: 0.0049626        total: 1m 38s   remaining: 47.9s
336:    learn: 0.0049544        total: 1m 38s   remaining: 47.6s
337:    learn: 0.0049335        total: 1m 38s   remaining: 47.3s
338:    learn: 0.0049165        total: 1m 38s   remaining: 47s
339:    learn: 0.0049051        total: 1m 39s   remaining: 46.7s
340:    learn: 0.0048804        total: 1m 39s   remaining: 46.3s
341:    learn: 0.0048665        total: 1m 39s   remaining: 46s
342:    learn: 0.0048417        total: 1m 39s   remaining: 45.7s
343:    learn: 0.0048344        total: 1m 40s   remaining: 45.4s
344:    learn: 0.0048117        total: 1m 40s   remaining: 45.1s
345:    learn: 0.0047909        total: 1m 40s   remaining: 44.8s
346:    learn: 0.0047702        total: 1m 41s   remaining: 44.5s
347:    learn: 0.0047469        total: 1m 41s   remaining: 44.2s
348:    learn: 0.0047343        total: 1m 41s   remaining: 43.9s
349:    learn: 0.0047156        total: 1m 41s   remaining: 43.6s
350:    learn: 0.0046917        total: 1m 42s   remaining: 43.4s
351:    learn: 0.0046817        total: 1m 42s   remaining: 43.1s
352:    learn: 0.0046648        total: 1m 42s   remaining: 42.8s
353:    learn: 0.0046535        total: 1m 42s   remaining: 42.5s
354:    learn: 0.0046353        total: 1m 43s   remaining: 42.2s
355:    learn: 0.0046126        total: 1m 43s   remaining: 41.9s
356:    learn: 0.0045948        total: 1m 43s   remaining: 41.6s
357:    learn: 0.0045827        total: 1m 44s   remaining: 41.3s
358:    learn: 0.0045648        total: 1m 44s   remaining: 41s
359:    learn: 0.0045497        total: 1m 44s   remaining: 40.7s
360:    learn: 0.0045348        total: 1m 44s   remaining: 40.4s
361:    learn: 0.0045296        total: 1m 45s   remaining: 40.1s
362:    learn: 0.0045190        total: 1m 45s   remaining: 39.8s
363:    learn: 0.0045063        total: 1m 45s   remaining: 39.5s
364:    learn: 0.0044884        total: 1m 45s   remaining: 39.2s
365:    learn: 0.0044712        total: 1m 46s   remaining: 38.9s
366:    learn: 0.0044611        total: 1m 46s   remaining: 38.6s
367:    learn: 0.0044452        total: 1m 46s   remaining: 38.3s
368:    learn: 0.0044329        total: 1m 47s   remaining: 38s
369:    learn: 0.0044019        total: 1m 47s   remaining: 37.7s
370:    learn: 0.0043892        total: 1m 47s   remaining: 37.4s
371:    learn: 0.0043618        total: 1m 47s   remaining: 37.1s
372:    learn: 0.0043514        total: 1m 48s   remaining: 36.8s
373:    learn: 0.0043317        total: 1m 48s   remaining: 36.5s
374:    learn: 0.0043169        total: 1m 48s   remaining: 36.2s
375:    learn: 0.0043109        total: 1m 48s   remaining: 35.9s
376:    learn: 0.0042944        total: 1m 49s   remaining: 35.6s
377:    learn: 0.0042804        total: 1m 49s   remaining: 35.3s
378:    learn: 0.0042688        total: 1m 49s   remaining: 35s
379:    learn: 0.0042527        total: 1m 50s   remaining: 34.8s
380:    learn: 0.0042359        total: 1m 50s   remaining: 34.5s
381:    learn: 0.0042152        total: 1m 50s   remaining: 34.2s
382:    learn: 0.0042032        total: 1m 50s   remaining: 33.9s
```

```
383:    learn: 0.0041921        total: 1m 51s   remaining: 33.6s
384:    learn: 0.0041717        total: 1m 51s   remaining: 33.3s
385:    learn: 0.0041612        total: 1m 51s   remaining: 33s
386:    learn: 0.0041409        total: 1m 52s   remaining: 32.7s
387:    learn: 0.0041188        total: 1m 52s   remaining: 32.4s
388:    learn: 0.0041005        total: 1m 52s   remaining: 32.1s
389:    learn: 0.0040830        total: 1m 52s   remaining: 31.8s
390:    learn: 0.0040720        total: 1m 53s   remaining: 31.5s
391:    learn: 0.0040632        total: 1m 53s   remaining: 31.3s
392:    learn: 0.0040446        total: 1m 53s   remaining: 31s
393:    learn: 0.0040292        total: 1m 53s   remaining: 30.7s
394:    learn: 0.0040177        total: 1m 54s   remaining: 30.4s
395:    learn: 0.0040100        total: 1m 54s   remaining: 30.1s
396:    learn: 0.0039911        total: 1m 54s   remaining: 29.8s
397:    learn: 0.0039872        total: 1m 54s   remaining: 29.4s
398:    learn: 0.0039778        total: 1m 55s   remaining: 29.2s
399:    learn: 0.0039575        total: 1m 55s   remaining: 28.9s
400:    learn: 0.0039493        total: 1m 55s   remaining: 28.6s
401:    learn: 0.0039393        total: 1m 55s   remaining: 28.3s
402:    learn: 0.0039256        total: 1m 56s   remaining: 28s
403:    learn: 0.0039156        total: 1m 56s   remaining: 27.7s
404:    learn: 0.0038998        total: 1m 56s   remaining: 27.4s
405:    learn: 0.0038869        total: 1m 57s   remaining: 27.1s
406:    learn: 0.0038765        total: 1m 57s   remaining: 26.8s
407:    learn: 0.0038659        total: 1m 57s   remaining: 26.5s
408:    learn: 0.0038513        total: 1m 57s   remaining: 26.2s
409:    learn: 0.0038414        total: 1m 58s   remaining: 25.9s
410:    learn: 0.0038236        total: 1m 58s   remaining: 25.6s
411:    learn: 0.0038115        total: 1m 58s   remaining: 25.3s
412:    learn: 0.0038025        total: 1m 58s   remaining: 25s
413:    learn: 0.0037940        total: 1m 59s   remaining: 24.7s
414:    learn: 0.0037874        total: 1m 59s   remaining: 24.4s
415:    learn: 0.0037699        total: 1m 59s   remaining: 24.2s
416:    learn: 0.0037633        total: 1m 59s   remaining: 23.9s
417:    learn: 0.0037553        total: 2m       remaining: 23.6s
418:    learn: 0.0037525        total: 2m       remaining: 23.3s
419:    learn: 0.0037425        total: 2m       remaining: 23s
420:    learn: 0.0037293        total: 2m       remaining: 22.7s
421:    learn: 0.0037140        total: 2m 1s    remaining: 22.4s
422:    learn: 0.0037085        total: 2m 1s    remaining: 22.1s
423:    learn: 0.0036959        total: 2m 1s    remaining: 21.8s
424:    learn: 0.0036886        total: 2m 2s    remaining: 21.5s
425:    learn: 0.0036776        total: 2m 2s    remaining: 21.3s
426:    learn: 0.0036650        total: 2m 2s    remaining: 21s
427:    learn: 0.0036562        total: 2m 2s    remaining: 20.7s
428:    learn: 0.0036467        total: 2m 3s    remaining: 20.4s
429:    learn: 0.0036353        total: 2m 3s    remaining: 20.1s
430:    learn: 0.0036260        total: 2m 3s    remaining: 19.8s
```

```
431:    learn: 0.0036099        total: 2m 4s   remaining: 19.5s
432:    learn: 0.0036024        total: 2m 4s   remaining: 19.2s
433:    learn: 0.0035930        total: 2m 4s   remaining: 18.9s
434:    learn: 0.0035760        total: 2m 4s   remaining: 18.7s
435:    learn: 0.0035657        total: 2m 5s   remaining: 18.4s
436:    learn: 0.0035508        total: 2m 5s   remaining: 18.1s
437:    learn: 0.0035413        total: 2m 5s   remaining: 17.8s
438:    learn: 0.0035271        total: 2m 5s   remaining: 17.5s
439:    learn: 0.0035175        total: 2m 6s   remaining: 17.2s
440:    learn: 0.0035102        total: 2m 6s   remaining: 16.9s
441:    learn: 0.0034948        total: 2m 6s   remaining: 16.6s
442:    learn: 0.0034880        total: 2m 7s   remaining: 16.3s
443:    learn: 0.0034705        total: 2m 7s   remaining: 16.1s
444:    learn: 0.0034655        total: 2m 7s   remaining: 15.8s
445:    learn: 0.0034533        total: 2m 7s   remaining: 15.5s
446:    learn: 0.0034390        total: 2m 8s   remaining: 15.2s
447:    learn: 0.0034268        total: 2m 8s   remaining: 14.9s
448:    learn: 0.0034113        total: 2m 8s   remaining: 14.6s
449:    learn: 0.0033957        total: 2m 8s   remaining: 14.3s
450:    learn: 0.0033874        total: 2m 9s   remaining: 14s
451:    learn: 0.0033824        total: 2m 9s   remaining: 13.8s
452:    learn: 0.0033692        total: 2m 9s   remaining: 13.5s
453:    learn: 0.0033617        total: 2m 10s  remaining: 13.2s
454:    learn: 0.0033531        total: 2m 10s  remaining: 12.9s
455:    learn: 0.0033448        total: 2m 10s  remaining: 12.6s
456:    learn: 0.0033386        total: 2m 10s  remaining: 12.3s
457:    learn: 0.0033271        total: 2m 11s  remaining: 12s
458:    learn: 0.0033169        total: 2m 11s  remaining: 11.7s
459:    learn: 0.0033110        total: 2m 11s  remaining: 11.5s
460:    learn: 0.0032981        total: 2m 11s  remaining: 11.2s
461:    learn: 0.0032826        total: 2m 12s  remaining: 10.9s
462:    learn: 0.0032645        total: 2m 12s  remaining: 10.6s
463:    learn: 0.0032591        total: 2m 12s  remaining: 10.3s
464:    learn: 0.0032502        total: 2m 13s  remaining: 10s
465:    learn: 0.0032445        total: 2m 13s  remaining: 9.73s
466:    learn: 0.0032349        total: 2m 13s  remaining: 9.44s
467:    learn: 0.0032251        total: 2m 13s  remaining: 9.15s
468:    learn: 0.0032143        total: 2m 14s  remaining: 8.86s
469:    learn: 0.0031982        total: 2m 14s  remaining: 8.58s
470:    learn: 0.0031866        total: 2m 14s  remaining: 8.29s
471:    learn: 0.0031710        total: 2m 14s  remaining: 8s
472:    learn: 0.0031610        total: 2m 15s  remaining: 7.72s
473:    learn: 0.0031512        total: 2m 15s  remaining: 7.43s
474:    learn: 0.0031404        total: 2m 15s  remaining: 7.14s
475:    learn: 0.0031299        total: 2m 16s  remaining: 6.86s
476:    learn: 0.0031147        total: 2m 16s  remaining: 6.57s
477:    learn: 0.0031080        total: 2m 16s  remaining: 6.29s
478:    learn: 0.0030978        total: 2m 16s  remaining: 6s
```

```
479:    learn: 0.0030931        total: 2m 17s   remaining: 5.71s
480:    learn: 0.0030828        total: 2m 17s   remaining: 5.43s
481:    learn: 0.0030704        total: 2m 17s   remaining: 5.14s
482:    learn: 0.0030651        total: 2m 17s   remaining: 4.86s
483:    learn: 0.0030590        total: 2m 18s   remaining: 4.57s
484:    learn: 0.0030480        total: 2m 18s   remaining: 4.28s
485:    learn: 0.0030434        total: 2m 18s   remaining: 4s
486:    learn: 0.0030359        total: 2m 19s   remaining: 3.71s
487:    learn: 0.0030254        total: 2m 19s   remaining: 3.43s
488:    learn: 0.0030186        total: 2m 19s   remaining: 3.14s
489:    learn: 0.0030116        total: 2m 20s   remaining: 2.86s
490:    learn: 0.0030025        total: 2m 20s   remaining: 2.57s
491:    learn: 0.0029874        total: 2m 20s   remaining: 2.29s
492:    learn: 0.0029751        total: 2m 20s   remaining: 2s
493:    learn: 0.0029665        total: 2m 21s   remaining: 1.72s
494:    learn: 0.0029606        total: 2m 21s   remaining: 1.43s
495:    learn: 0.0029512        total: 2m 21s   remaining: 1.14s
496:    learn: 0.0029426        total: 2m 22s   remaining: 857ms
497:    learn: 0.0029290        total: 2m 22s   remaining: 572ms
498:    learn: 0.0029183        total: 2m 22s   remaining: 286ms
499:    learn: 0.0029053        total: 2m 22s   remaining: 0us
0:      learn: 0.9673649        total: 157ms    remaining: 1m 18s
1:      learn: 0.8517171        total: 440ms    remaining: 1m 49s
2:      learn: 0.7634208        total: 711ms    remaining: 1m 57s
3:      learn: 0.6771553        total: 978ms    remaining: 2m 1s
4:      learn: 0.6048609        total: 1.26s    remaining: 2m 4s
5:      learn: 0.5457834        total: 1.57s    remaining: 2m 9s
6:      learn: 0.4907402        total: 1.83s    remaining: 2m 8s
7:      learn: 0.4444585        total: 2.08s    remaining: 2m 7s
8:      learn: 0.4023031        total: 2.34s    remaining: 2m 7s
9:      learn: 0.3676351        total: 2.61s    remaining: 2m 8s
10:     learn: 0.3372446        total: 2.88s    remaining: 2m 8s
11:     learn: 0.3094714        total: 3.18s    remaining: 2m 9s
12:     learn: 0.2856615        total: 3.5s     remaining: 2m 11s
13:     learn: 0.2626836        total: 3.83s    remaining: 2m 13s
14:     learn: 0.2429614        total: 4.1s     remaining: 2m 12s
15:     learn: 0.2219064        total: 4.34s    remaining: 2m 11s
16:     learn: 0.2028202        total: 4.57s    remaining: 2m 9s
17:     learn: 0.1860408        total: 4.81s    remaining: 2m 8s
18:     learn: 0.1704871        total: 5.08s    remaining: 2m 8s
19:     learn: 0.1570327        total: 5.32s    remaining: 2m 7s
20:     learn: 0.1457580        total: 5.57s    remaining: 2m 7s
21:     learn: 0.1343605        total: 5.82s    remaining: 2m 6s
22:     learn: 0.1243349        total: 6.05s    remaining: 2m 5s
23:     learn: 0.1153291        total: 6.34s    remaining: 2m 5s
24:     learn: 0.1073142        total: 6.6s     remaining: 2m 5s
25:     learn: 0.1003349        total: 6.85s    remaining: 2m 4s
26:     learn: 0.0940772        total: 7.09s    remaining: 2m 4s
```

```
27:     learn: 0.0879334        total: 7.33s    remaining: 2m 3s
28:     learn: 0.0826950        total: 7.56s    remaining: 2m 2s
29:     learn: 0.0778750        total: 7.82s    remaining: 2m 2s
30:     learn: 0.0743027        total: 8.1s     remaining: 2m 2s
31:     learn: 0.0704177        total: 8.34s    remaining: 2m 1s
32:     learn: 0.0664079        total: 8.6s     remaining: 2m 1s
33:     learn: 0.0627546        total: 8.88s    remaining: 2m 1s
34:     learn: 0.0594175        total: 9.14s    remaining: 2m 1s
35:     learn: 0.0565431        total: 9.39s    remaining: 2m 1s
36:     learn: 0.0538559        total: 9.66s    remaining: 2m
37:     learn: 0.0514690        total: 9.92s    remaining: 2m
38:     learn: 0.0493039        total: 10.2s    remaining: 2m
39:     learn: 0.0473053        total: 10.5s    remaining: 2m
40:     learn: 0.0453878        total: 10.7s    remaining: 2m
41:     learn: 0.0436736        total: 11s      remaining: 1m 59s
42:     learn: 0.0422608        total: 11.3s    remaining: 1m 59s
43:     learn: 0.0407324        total: 11.6s    remaining: 1m 59s
44:     learn: 0.0392514        total: 11.8s    remaining: 1m 59s
45:     learn: 0.0380677        total: 12.1s    remaining: 1m 59s
46:     learn: 0.0366423        total: 12.4s    remaining: 1m 59s
47:     learn: 0.0351671        total: 12.7s    remaining: 1m 59s
48:     learn: 0.0340296        total: 12.9s    remaining: 1m 58s
49:     learn: 0.0326836        total: 13.2s    remaining: 1m 58s
50:     learn: 0.0319432        total: 13.4s    remaining: 1m 58s
51:     learn: 0.0311549        total: 13.8s    remaining: 1m 58s
52:     learn: 0.0302398        total: 14s      remaining: 1m 58s
53:     learn: 0.0294429        total: 14.3s    remaining: 1m 57s
54:     learn: 0.0286969        total: 14.5s    remaining: 1m 57s
55:     learn: 0.0280179        total: 14.8s    remaining: 1m 57s
56:     learn: 0.0275152        total: 15.1s    remaining: 1m 57s
57:     learn: 0.0270533        total: 15.3s    remaining: 1m 56s
58:     learn: 0.0265602        total: 15.6s    remaining: 1m 56s
59:     learn: 0.0261326        total: 15.9s    remaining: 1m 56s
60:     learn: 0.0256426        total: 16.1s    remaining: 1m 56s
61:     learn: 0.0249487        total: 16.4s    remaining: 1m 56s
62:     learn: 0.0244678        total: 16.7s    remaining: 1m 56s
63:     learn: 0.0240046        total: 17.1s    remaining: 1m 56s
64:     learn: 0.0237036        total: 17.4s    remaining: 1m 56s
65:     learn: 0.0232805        total: 17.6s    remaining: 1m 55s
66:     learn: 0.0229298        total: 17.9s    remaining: 1m 55s
67:     learn: 0.0223823        total: 18.1s    remaining: 1m 55s
68:     learn: 0.0221115        total: 18.4s    remaining: 1m 55s
69:     learn: 0.0218764        total: 18.7s    remaining: 1m 54s
70:     learn: 0.0215838        total: 19s      remaining: 1m 54s
71:     learn: 0.0215374        total: 19.2s    remaining: 1m 54s
72:     learn: 0.0209544        total: 19.5s    remaining: 1m 53s
73:     learn: 0.0206676        total: 19.8s    remaining: 1m 53s
74:     learn: 0.0205426        total: 20s      remaining: 1m 53s
```

```
75:     learn: 0.0204093        total: 20.3s    remaining: 1m 53s
76:     learn: 0.0201434        total: 20.6s    remaining: 1m 53s
77:     learn: 0.0196604        total: 20.9s    remaining: 1m 52s
78:     learn: 0.0193634        total: 21.1s    remaining: 1m 52s
79:     learn: 0.0192263        total: 21.4s    remaining: 1m 52s
80:     learn: 0.0188908        total: 21.7s    remaining: 1m 52s
81:     learn: 0.0185705        total: 22s      remaining: 1m 52s
82:     learn: 0.0181618        total: 22.4s    remaining: 1m 52s
83:     learn: 0.0179312        total: 22.6s    remaining: 1m 52s
84:     learn: 0.0178030        total: 22.9s    remaining: 1m 51s
85:     learn: 0.0177282        total: 23.2s    remaining: 1m 51s
86:     learn: 0.0175579        total: 23.5s    remaining: 1m 51s
87:     learn: 0.0174734        total: 23.7s    remaining: 1m 51s
88:     learn: 0.0172881        total: 24s      remaining: 1m 50s
89:     learn: 0.0171998        total: 24.3s    remaining: 1m 50s
90:     learn: 0.0170598        total: 24.6s    remaining: 1m 50s
91:     learn: 0.0169420        total: 24.9s    remaining: 1m 50s
92:     learn: 0.0168369        total: 25.2s    remaining: 1m 50s
93:     learn: 0.0165384        total: 25.5s    remaining: 1m 50s
94:     learn: 0.0163961        total: 25.7s    remaining: 1m 49s
95:     learn: 0.0163494        total: 26s      remaining: 1m 49s
96:     learn: 0.0160190        total: 26.3s    remaining: 1m 49s
97:     learn: 0.0159114        total: 26.5s    remaining: 1m 48s
98:     learn: 0.0157155        total: 26.8s    remaining: 1m 48s
99:     learn: 0.0155373        total: 27.1s    remaining: 1m 48s
100:    learn: 0.0154174        total: 27.3s    remaining: 1m 47s
101:    learn: 0.0152441        total: 27.6s    remaining: 1m 47s
102:    learn: 0.0151834        total: 27.9s    remaining: 1m 47s
103:    learn: 0.0150709        total: 28.2s    remaining: 1m 47s
104:    learn: 0.0149764        total: 28.5s    remaining: 1m 47s
105:    learn: 0.0148590        total: 28.7s    remaining: 1m 46s
106:    learn: 0.0147678        total: 29s      remaining: 1m 46s
107:    learn: 0.0146355        total: 29.3s    remaining: 1m 46s
108:    learn: 0.0145051        total: 29.5s    remaining: 1m 45s
109:    learn: 0.0143671        total: 29.8s    remaining: 1m 45s
110:    learn: 0.0142626        total: 30s      remaining: 1m 45s
111:    learn: 0.0141518        total: 30.3s    remaining: 1m 45s
112:    learn: 0.0140978        total: 30.6s    remaining: 1m 44s
113:    learn: 0.0139938        total: 30.9s    remaining: 1m 44s
114:    learn: 0.0139252        total: 31.1s    remaining: 1m 44s
115:    learn: 0.0138155        total: 31.4s    remaining: 1m 43s
116:    learn: 0.0137436        total: 31.6s    remaining: 1m 43s
117:    learn: 0.0136616        total: 31.9s    remaining: 1m 43s
118:    learn: 0.0135673        total: 32.1s    remaining: 1m 42s
119:    learn: 0.0134838        total: 32.4s    remaining: 1m 42s
120:    learn: 0.0134402        total: 32.7s    remaining: 1m 42s
121:    learn: 0.0133940        total: 33s      remaining: 1m 42s
122:    learn: 0.0132142        total: 33.2s    remaining: 1m 41s
```

```
123:    learn: 0.0130565        total: 33.5s    remaining: 1m 41s
124:    learn: 0.0130107        total: 33.8s    remaining: 1m 41s
125:    learn: 0.0129042        total: 34.1s    remaining: 1m 41s
126:    learn: 0.0128609        total: 34.3s    remaining: 1m 40s
127:    learn: 0.0127655        total: 34.6s    remaining: 1m 40s
128:    learn: 0.0127115        total: 34.9s    remaining: 1m 40s
129:    learn: 0.0126482        total: 35.1s    remaining: 1m 40s
130:    learn: 0.0125241        total: 35.4s    remaining: 1m 39s
131:    learn: 0.0123958        total: 35.7s    remaining: 1m 39s
132:    learn: 0.0123110        total: 36s      remaining: 1m 39s
133:    learn: 0.0122768        total: 36.3s    remaining: 1m 39s
134:    learn: 0.0122204        total: 36.5s    remaining: 1m 38s
135:    learn: 0.0121331        total: 36.8s    remaining: 1m 38s
136:    learn: 0.0120112        total: 37.1s    remaining: 1m 38s
137:    learn: 0.0119411        total: 37.4s    remaining: 1m 38s
138:    learn: 0.0119113        total: 37.7s    remaining: 1m 37s
139:    learn: 0.0118792        total: 37.9s    remaining: 1m 37s
140:    learn: 0.0118089        total: 38.2s    remaining: 1m 37s
141:    learn: 0.0117659        total: 38.5s    remaining: 1m 37s
142:    learn: 0.0116799        total: 38.8s    remaining: 1m 36s
143:    learn: 0.0115960        total: 39s      remaining: 1m 36s
144:    learn: 0.0115476        total: 39.3s    remaining: 1m 36s
145:    learn: 0.0115191        total: 39.6s    remaining: 1m 36s
146:    learn: 0.0114551        total: 39.9s    remaining: 1m 35s
147:    learn: 0.0114102        total: 40.2s    remaining: 1m 35s
148:    learn: 0.0113392        total: 40.4s    remaining: 1m 35s
149:    learn: 0.0112437        total: 40.7s    remaining: 1m 34s
150:    learn: 0.0111770        total: 41s      remaining: 1m 34s
151:    learn: 0.0111025        total: 41.3s    remaining: 1m 34s
152:    learn: 0.0110886        total: 41.5s    remaining: 1m 34s
153:    learn: 0.0110176        total: 41.8s    remaining: 1m 33s
154:    learn: 0.0109714        total: 42s      remaining: 1m 33s
155:    learn: 0.0109211        total: 42.3s    remaining: 1m 33s
156:    learn: 0.0107982        total: 42.6s    remaining: 1m 33s
157:    learn: 0.0107576        total: 42.8s    remaining: 1m 32s
158:    learn: 0.0106674        total: 43.1s    remaining: 1m 32s
159:    learn: 0.0106223        total: 43.4s    remaining: 1m 32s
160:    learn: 0.0105432        total: 43.7s    remaining: 1m 31s
161:    learn: 0.0104971        total: 43.9s    remaining: 1m 31s
162:    learn: 0.0104703        total: 44.2s    remaining: 1m 31s
163:    learn: 0.0104186        total: 44.5s    remaining: 1m 31s
164:    learn: 0.0103352        total: 44.8s    remaining: 1m 30s
165:    learn: 0.0102567        total: 45s      remaining: 1m 30s
166:    learn: 0.0102151        total: 45.3s    remaining: 1m 30s
167:    learn: 0.0102037        total: 45.5s    remaining: 1m 29s
168:    learn: 0.0101371        total: 45.8s    remaining: 1m 29s
169:    learn: 0.0100986        total: 46.1s    remaining: 1m 29s
170:    learn: 0.0100535        total: 46.3s    remaining: 1m 29s
```

```
171:    learn: 0.0100064        total: 46.6s    remaining: 1m 28s
172:    learn: 0.0099290        total: 46.9s    remaining: 1m 28s
173:    learn: 0.0098793        total: 47.2s    remaining: 1m 28s
174:    learn: 0.0098475        total: 47.5s    remaining: 1m 28s
175:    learn: 0.0097762        total: 47.8s    remaining: 1m 27s
176:    learn: 0.0097505        total: 48s      remaining: 1m 27s
177:    learn: 0.0097258        total: 48.3s    remaining: 1m 27s
178:    learn: 0.0096815        total: 48.5s    remaining: 1m 27s
179:    learn: 0.0096036        total: 48.8s    remaining: 1m 26s
180:    learn: 0.0095267        total: 49.1s    remaining: 1m 26s
181:    learn: 0.0094750        total: 49.4s    remaining: 1m 26s
182:    learn: 0.0094232        total: 49.6s    remaining: 1m 25s
183:    learn: 0.0093665        total: 49.9s    remaining: 1m 25s
184:    learn: 0.0092827        total: 50.2s    remaining: 1m 25s
185:    learn: 0.0092610        total: 50.5s    remaining: 1m 25s
186:    learn: 0.0092047        total: 50.8s    remaining: 1m 24s
187:    learn: 0.0091730        total: 51s      remaining: 1m 24s
188:    learn: 0.0091282        total: 51.3s    remaining: 1m 24s
189:    learn: 0.0091005        total: 51.6s    remaining: 1m 24s
190:    learn: 0.0090706        total: 51.8s    remaining: 1m 23s
191:    learn: 0.0090316        total: 52.1s    remaining: 1m 23s
192:    learn: 0.0089842        total: 52.4s    remaining: 1m 23s
193:    learn: 0.0089179        total: 52.7s    remaining: 1m 23s
194:    learn: 0.0088881        total: 53s      remaining: 1m 22s
195:    learn: 0.0088614        total: 53.2s    remaining: 1m 22s
196:    learn: 0.0088107        total: 53.5s    remaining: 1m 22s
197:    learn: 0.0087350        total: 53.8s    remaining: 1m 22s
198:    learn: 0.0087112        total: 54.1s    remaining: 1m 21s
199:    learn: 0.0086580        total: 54.4s    remaining: 1m 21s
200:    learn: 0.0085909        total: 54.6s    remaining: 1m 21s
201:    learn: 0.0085562        total: 54.9s    remaining: 1m 20s
202:    learn: 0.0085251        total: 55.2s    remaining: 1m 20s
203:    learn: 0.0084858        total: 55.4s    remaining: 1m 20s
204:    learn: 0.0084393        total: 55.7s    remaining: 1m 20s
205:    learn: 0.0083736        total: 56s      remaining: 1m 19s
206:    learn: 0.0083208        total: 56.3s    remaining: 1m 19s
207:    learn: 0.0082611        total: 56.6s    remaining: 1m 19s
208:    learn: 0.0082050        total: 56.8s    remaining: 1m 19s
209:    learn: 0.0081777        total: 57.1s    remaining: 1m 18s
210:    learn: 0.0081401        total: 57.4s    remaining: 1m 18s
211:    learn: 0.0081265        total: 57.7s    remaining: 1m 18s
212:    learn: 0.0080854        total: 58s      remaining: 1m 18s
213:    learn: 0.0080477        total: 58.2s    remaining: 1m 17s
214:    learn: 0.0080233        total: 58.5s    remaining: 1m 17s
215:    learn: 0.0079937        total: 58.8s    remaining: 1m 17s
216:    learn: 0.0079422        total: 59s      remaining: 1m 16s
217:    learn: 0.0079076        total: 59.3s    remaining: 1m 16s
218:    learn: 0.0078612        total: 59.5s    remaining: 1m 16s
```

```
219:    learn: 0.0078280        total: 59.8s    remaining: 1m 16s
220:    learn: 0.0077952        total: 1m       remaining: 1m 15s
221:    learn: 0.0077618        total: 1m       remaining: 1m 15s
222:    learn: 0.0077186        total: 1m       remaining: 1m 15s
223:    learn: 0.0076897        total: 1m       remaining: 1m 15s
224:    learn: 0.0076719        total: 1m 1s    remaining: 1m 14s
225:    learn: 0.0076603        total: 1m 1s    remaining: 1m 14s
226:    learn: 0.0076242        total: 1m 1s    remaining: 1m 14s
227:    learn: 0.0076041        total: 1m 1s    remaining: 1m 13s
228:    learn: 0.0075572        total: 1m 2s    remaining: 1m 13s
229:    learn: 0.0075241        total: 1m 2s    remaining: 1m 13s
230:    learn: 0.0074906        total: 1m 2s    remaining: 1m 13s
231:    learn: 0.0074225        total: 1m 3s    remaining: 1m 12s
232:    learn: 0.0073979        total: 1m 3s    remaining: 1m 12s
233:    learn: 0.0073600        total: 1m 3s    remaining: 1m 12s
234:    learn: 0.0073293        total: 1m 3s    remaining: 1m 12s
235:    learn: 0.0073063        total: 1m 4s    remaining: 1m 11s
236:    learn: 0.0072786        total: 1m 4s    remaining: 1m 11s
237:    learn: 0.0072557        total: 1m 4s    remaining: 1m 11s
238:    learn: 0.0071888        total: 1m 5s    remaining: 1m 11s
239:    learn: 0.0071696        total: 1m 5s    remaining: 1m 10s
240:    learn: 0.0071348        total: 1m 5s    remaining: 1m 10s
241:    learn: 0.0071020        total: 1m 5s    remaining: 1m 10s
242:    learn: 0.0070766        total: 1m 6s    remaining: 1m 10s
243:    learn: 0.0069977        total: 1m 6s    remaining: 1m 9s
244:    learn: 0.0069756        total: 1m 6s    remaining: 1m 9s
245:    learn: 0.0069504        total: 1m 7s    remaining: 1m 9s
246:    learn: 0.0069150        total: 1m 7s    remaining: 1m 8s
247:    learn: 0.0068566        total: 1m 7s    remaining: 1m 8s
248:    learn: 0.0068181        total: 1m 7s    remaining: 1m 8s
249:    learn: 0.0067807        total: 1m 8s    remaining: 1m 8s
250:    learn: 0.0067462        total: 1m 8s    remaining: 1m 7s
251:    learn: 0.0067162        total: 1m 8s    remaining: 1m 7s
252:    learn: 0.0066593        total: 1m 8s    remaining: 1m 7s
253:    learn: 0.0066384        total: 1m 9s    remaining: 1m 7s
254:    learn: 0.0065950        total: 1m 9s    remaining: 1m 6s
255:    learn: 0.0065863        total: 1m 9s    remaining: 1m 6s
256:    learn: 0.0065611        total: 1m 9s    remaining: 1m 6s
257:    learn: 0.0065272        total: 1m 10s   remaining: 1m 5s
258:    learn: 0.0065000        total: 1m 10s   remaining: 1m 5s
259:    learn: 0.0064688        total: 1m 10s   remaining: 1m 5s
260:    learn: 0.0064539        total: 1m 11s   remaining: 1m 5s
261:    learn: 0.0064272        total: 1m 11s   remaining: 1m 4s
262:    learn: 0.0064140        total: 1m 11s   remaining: 1m 4s
263:    learn: 0.0063959        total: 1m 11s   remaining: 1m 4s
264:    learn: 0.0063900        total: 1m 12s   remaining: 1m 3s
265:    learn: 0.0063716        total: 1m 12s   remaining: 1m 3s
266:    learn: 0.0063482        total: 1m 12s   remaining: 1m 3s
```

```
267:    learn: 0.0063309        total: 1m 13s   remaining: 1m 3s
268:    learn: 0.0063138        total: 1m 13s   remaining: 1m 3s
269:    learn: 0.0062817        total: 1m 13s   remaining: 1m 2s
270:    learn: 0.0062579        total: 1m 13s   remaining: 1m 2s
271:    learn: 0.0062324        total: 1m 14s   remaining: 1m 2s
272:    learn: 0.0061912        total: 1m 14s   remaining: 1m 1s
273:    learn: 0.0061624        total: 1m 14s   remaining: 1m 1s
274:    learn: 0.0061299        total: 1m 14s   remaining: 1m 1s
275:    learn: 0.0060983        total: 1m 15s   remaining: 1m 1s
276:    learn: 0.0060794        total: 1m 15s   remaining: 1m
277:    learn: 0.0060552        total: 1m 15s   remaining: 1m
278:    learn: 0.0060281        total: 1m 16s   remaining: 1m
279:    learn: 0.0060047        total: 1m 16s   remaining: 60s
280:    learn: 0.0059920        total: 1m 16s   remaining: 59.7s
281:    learn: 0.0059848        total: 1m 16s   remaining: 59.4s
282:    learn: 0.0059670        total: 1m 17s   remaining: 59.1s
283:    learn: 0.0059398        total: 1m 17s   remaining: 58.9s
284:    learn: 0.0059169        total: 1m 17s   remaining: 58.6s
285:    learn: 0.0059027        total: 1m 17s   remaining: 58.3s
286:    learn: 0.0058779        total: 1m 18s   remaining: 58.1s
287:    learn: 0.0058552        total: 1m 18s   remaining: 57.8s
288:    learn: 0.0058420        total: 1m 18s   remaining: 57.5s
289:    learn: 0.0058042        total: 1m 19s   remaining: 57.3s
290:    learn: 0.0057787        total: 1m 19s   remaining: 57s
291:    learn: 0.0057529        total: 1m 19s   remaining: 56.7s
292:    learn: 0.0057388        total: 1m 19s   remaining: 56.5s
293:    learn: 0.0057144        total: 1m 20s   remaining: 56.2s
294:    learn: 0.0056935        total: 1m 20s   remaining: 55.9s
295:    learn: 0.0056760        total: 1m 20s   remaining: 55.7s
296:    learn: 0.0056626        total: 1m 20s   remaining: 55.3s
297:    learn: 0.0056456        total: 1m 21s   remaining: 55.1s
298:    learn: 0.0056274        total: 1m 21s   remaining: 54.8s
299:    learn: 0.0056116        total: 1m 21s   remaining: 54.5s
300:    learn: 0.0055909        total: 1m 22s   remaining: 54.3s
301:    learn: 0.0055826        total: 1m 22s   remaining: 54s
302:    learn: 0.0055663        total: 1m 22s   remaining: 53.7s
303:    learn: 0.0055473        total: 1m 22s   remaining: 53.4s
304:    learn: 0.0055349        total: 1m 23s   remaining: 53.2s
305:    learn: 0.0055136        total: 1m 23s   remaining: 52.9s
306:    learn: 0.0054877        total: 1m 23s   remaining: 52.6s
307:    learn: 0.0054738        total: 1m 24s   remaining: 52.4s
308:    learn: 0.0054442        total: 1m 24s   remaining: 52.1s
309:    learn: 0.0054107        total: 1m 24s   remaining: 51.8s
310:    learn: 0.0053945        total: 1m 24s   remaining: 51.6s
311:    learn: 0.0053715        total: 1m 25s   remaining: 51.3s
312:    learn: 0.0053416        total: 1m 25s   remaining: 51s
313:    learn: 0.0053097        total: 1m 25s   remaining: 50.7s
314:    learn: 0.0052943        total: 1m 25s   remaining: 50.4s
```

```
315:    learn: 0.0052802        total: 1m 26s   remaining: 50.2s
316:    learn: 0.0052447        total: 1m 26s   remaining: 49.9s
317:    learn: 0.0052390        total: 1m 26s   remaining: 49.6s
318:    learn: 0.0052266        total: 1m 26s   remaining: 49.3s
319:    learn: 0.0052057        total: 1m 27s   remaining: 49.1s
320:    learn: 0.0051848        total: 1m 27s   remaining: 48.8s
321:    learn: 0.0051650        total: 1m 27s   remaining: 48.5s
322:    learn: 0.0051508        total: 1m 28s   remaining: 48.3s
323:    learn: 0.0051220        total: 1m 28s   remaining: 48s
324:    learn: 0.0051051        total: 1m 28s   remaining: 47.7s
325:    learn: 0.0050905        total: 1m 28s   remaining: 47.4s
326:    learn: 0.0050722        total: 1m 29s   remaining: 47.1s
327:    learn: 0.0050524        total: 1m 29s   remaining: 46.9s
328:    learn: 0.0050376        total: 1m 29s   remaining: 46.6s
329:    learn: 0.0050120        total: 1m 29s   remaining: 46.3s
330:    learn: 0.0049852        total: 1m 30s   remaining: 46s
331:    learn: 0.0049625        total: 1m 30s   remaining: 45.8s
332:    learn: 0.0049451        total: 1m 30s   remaining: 45.5s
333:    learn: 0.0049225        total: 1m 30s   remaining: 45.2s
334:    learn: 0.0048964        total: 1m 31s   remaining: 44.9s
335:    learn: 0.0048848        total: 1m 31s   remaining: 44.7s
336:    learn: 0.0048667        total: 1m 31s   remaining: 44.4s
337:    learn: 0.0048317        total: 1m 32s   remaining: 44.1s
338:    learn: 0.0048199        total: 1m 32s   remaining: 43.8s
339:    learn: 0.0048126        total: 1m 32s   remaining: 43.6s
340:    learn: 0.0047948        total: 1m 32s   remaining: 43.3s
341:    learn: 0.0047635        total: 1m 33s   remaining: 43s
342:    learn: 0.0047540        total: 1m 33s   remaining: 42.7s
343:    learn: 0.0047267        total: 1m 33s   remaining: 42.4s
344:    learn: 0.0047222        total: 1m 33s   remaining: 42.2s
345:    learn: 0.0046994        total: 1m 34s   remaining: 41.9s
346:    learn: 0.0046941        total: 1m 34s   remaining: 41.6s
347:    learn: 0.0046847        total: 1m 34s   remaining: 41.3s
348:    learn: 0.0046732        total: 1m 34s   remaining: 41.1s
349:    learn: 0.0046483        total: 1m 35s   remaining: 40.8s
350:    learn: 0.0046431        total: 1m 35s   remaining: 40.5s
351:    learn: 0.0046258        total: 1m 35s   remaining: 40.3s
352:    learn: 0.0046077        total: 1m 36s   remaining: 40s
353:    learn: 0.0045866        total: 1m 36s   remaining: 39.7s
354:    learn: 0.0045766        total: 1m 36s   remaining: 39.4s
355:    learn: 0.0045708        total: 1m 36s   remaining: 39.2s
356:    learn: 0.0045672        total: 1m 37s   remaining: 38.9s
357:    learn: 0.0045532        total: 1m 37s   remaining: 38.6s
358:    learn: 0.0045360        total: 1m 37s   remaining: 38.3s
359:    learn: 0.0045148        total: 1m 37s   remaining: 38.1s
360:    learn: 0.0044989        total: 1m 38s   remaining: 37.8s
361:    learn: 0.0044823        total: 1m 38s   remaining: 37.5s
362:    learn: 0.0044624        total: 1m 38s   remaining: 37.3s
```

```
363:    learn: 0.0044345        total: 1m 39s   remaining: 37s
364:    learn: 0.0044292        total: 1m 39s   remaining: 36.7s
365:    learn: 0.0044269        total: 1m 39s   remaining: 36.4s
366:    learn: 0.0044166        total: 1m 39s   remaining: 36.2s
367:    learn: 0.0043969        total: 1m 40s   remaining: 35.9s
368:    learn: 0.0043778        total: 1m 40s   remaining: 35.6s
369:    learn: 0.0043612        total: 1m 40s   remaining: 35.3s
370:    learn: 0.0043441        total: 1m 40s   remaining: 35.1s
371:    learn: 0.0043292        total: 1m 41s   remaining: 34.8s
372:    learn: 0.0043220        total: 1m 41s   remaining: 34.5s
373:    learn: 0.0043097        total: 1m 41s   remaining: 34.3s
374:    learn: 0.0042963        total: 1m 41s   remaining: 34s
375:    learn: 0.0042849        total: 1m 42s   remaining: 33.7s
376:    learn: 0.0042781        total: 1m 42s   remaining: 33.4s
377:    learn: 0.0042595        total: 1m 42s   remaining: 33.2s
378:    learn: 0.0042409        total: 1m 43s   remaining: 32.9s
379:    learn: 0.0042289        total: 1m 43s   remaining: 32.6s
380:    learn: 0.0042103        total: 1m 43s   remaining: 32.3s
381:    learn: 0.0041973        total: 1m 43s   remaining: 32s
382:    learn: 0.0041812        total: 1m 44s   remaining: 31.8s
383:    learn: 0.0041802        total: 1m 44s   remaining: 31.5s
384:    learn: 0.0041647        total: 1m 44s   remaining: 31.2s
385:    learn: 0.0041577        total: 1m 44s   remaining: 31s
386:    learn: 0.0041449        total: 1m 45s   remaining: 30.7s
387:    learn: 0.0041312        total: 1m 45s   remaining: 30.4s
388:    learn: 0.0041224        total: 1m 45s   remaining: 30.1s
389:    learn: 0.0041155        total: 1m 45s   remaining: 29.8s
390:    learn: 0.0041006        total: 1m 45s   remaining: 29.5s
391:    learn: 0.0040782        total: 1m 46s   remaining: 29.3s
392:    learn: 0.0040510        total: 1m 46s   remaining: 29s
393:    learn: 0.0040387        total: 1m 46s   remaining: 28.7s
394:    learn: 0.0040261        total: 1m 47s   remaining: 28.5s
395:    learn: 0.0040130        total: 1m 47s   remaining: 28.2s
396:    learn: 0.0039960        total: 1m 47s   remaining: 27.9s
397:    learn: 0.0039855        total: 1m 47s   remaining: 27.6s
398:    learn: 0.0039717        total: 1m 48s   remaining: 27.4s
399:    learn: 0.0039564        total: 1m 48s   remaining: 27.1s
400:    learn: 0.0039324        total: 1m 48s   remaining: 26.8s
401:    learn: 0.0039157        total: 1m 48s   remaining: 26.6s
402:    learn: 0.0039043        total: 1m 49s   remaining: 26.3s
403:    learn: 0.0038906        total: 1m 49s   remaining: 26s
404:    learn: 0.0038830        total: 1m 49s   remaining: 25.7s
405:    learn: 0.0038657        total: 1m 49s   remaining: 25.5s
406:    learn: 0.0038569        total: 1m 50s   remaining: 25.2s
407:    learn: 0.0038450        total: 1m 50s   remaining: 24.9s
408:    learn: 0.0038221        total: 1m 50s   remaining: 24.6s
409:    learn: 0.0038113        total: 1m 51s   remaining: 24.4s
410:    learn: 0.0038074        total: 1m 51s   remaining: 24.1s
```

```
411:    learn: 0.0038019        total: 1m 51s   remaining: 23.8s
412:    learn: 0.0037877        total: 1m 51s   remaining: 23.6s
413:    learn: 0.0037793        total: 1m 52s   remaining: 23.3s
414:    learn: 0.0037693        total: 1m 52s   remaining: 23s
415:    learn: 0.0037527        total: 1m 52s   remaining: 22.7s
416:    learn: 0.0037388        total: 1m 52s   remaining: 22.5s
417:    learn: 0.0037306        total: 1m 53s   remaining: 22.2s
418:    learn: 0.0037230        total: 1m 53s   remaining: 21.9s
419:    learn: 0.0037108        total: 1m 53s   remaining: 21.6s
420:    learn: 0.0037032        total: 1m 53s   remaining: 21.4s
421:    learn: 0.0036984        total: 1m 54s   remaining: 21.1s
422:    learn: 0.0036861        total: 1m 54s   remaining: 20.9s
423:    learn: 0.0036677        total: 1m 54s   remaining: 20.6s
424:    learn: 0.0036597        total: 1m 55s   remaining: 20.3s
425:    learn: 0.0036464        total: 1m 55s   remaining: 20s
426:    learn: 0.0036391        total: 1m 55s   remaining: 19.8s
427:    learn: 0.0036334        total: 1m 55s   remaining: 19.5s
428:    learn: 0.0036189        total: 1m 56s   remaining: 19.2s
429:    learn: 0.0035998        total: 1m 56s   remaining: 19s
430:    learn: 0.0035894        total: 1m 56s   remaining: 18.7s
431:    learn: 0.0035812        total: 1m 56s   remaining: 18.4s
432:    learn: 0.0035714        total: 1m 57s   remaining: 18.1s
433:    learn: 0.0035557        total: 1m 57s   remaining: 17.9s
434:    learn: 0.0035537        total: 1m 57s   remaining: 17.6s
435:    learn: 0.0035446        total: 1m 58s   remaining: 17.3s
436:    learn: 0.0035304        total: 1m 58s   remaining: 17.1s
437:    learn: 0.0035176        total: 1m 58s   remaining: 16.8s
438:    learn: 0.0035168        total: 1m 58s   remaining: 16.5s
439:    learn: 0.0035132        total: 1m 59s   remaining: 16.2s
440:    learn: 0.0035037        total: 1m 59s   remaining: 16s
441:    learn: 0.0034890        total: 1m 59s   remaining: 15.7s
442:    learn: 0.0034831        total: 1m 59s   remaining: 15.4s
443:    learn: 0.0034671        total: 2m       remaining: 15.2s
444:    learn: 0.0034485        total: 2m       remaining: 14.9s
445:    learn: 0.0034433        total: 2m       remaining: 14.6s
446:    learn: 0.0034352        total: 2m       remaining: 14.3s
447:    learn: 0.0034219        total: 2m 1s    remaining: 14.1s
448:    learn: 0.0034135        total: 2m 1s    remaining: 13.8s
449:    learn: 0.0034003        total: 2m 1s    remaining: 13.5s
450:    learn: 0.0033899        total: 2m 2s    remaining: 13.3s
451:    learn: 0.0033788        total: 2m 2s    remaining: 13s
452:    learn: 0.0033632        total: 2m 2s    remaining: 12.7s
453:    learn: 0.0033544        total: 2m 2s    remaining: 12.4s
454:    learn: 0.0033475        total: 2m 3s    remaining: 12.2s
455:    learn: 0.0033378        total: 2m 3s    remaining: 11.9s
456:    learn: 0.0033295        total: 2m 3s    remaining: 11.6s
457:    learn: 0.0033192        total: 2m 3s    remaining: 11.4s
458:    learn: 0.0033074        total: 2m 4s    remaining: 11.1s
```

```
459:    learn: 0.0033050        total: 2m 4s    remaining: 10.8s
460:    learn: 0.0032953        total: 2m 4s    remaining: 10.6s
461:    learn: 0.0032904        total: 2m 5s    remaining: 10.3s
462:    learn: 0.0032815        total: 2m 5s    remaining: 10s
463:    learn: 0.0032748        total: 2m 5s    remaining: 9.74s
464:    learn: 0.0032644        total: 2m 5s    remaining: 9.47s
465:    learn: 0.0032570        total: 2m 6s    remaining: 9.21s
466:    learn: 0.0032507        total: 2m 6s    remaining: 8.93s
467:    learn: 0.0032439        total: 2m 6s    remaining: 8.66s
468:    learn: 0.0032359        total: 2m 6s    remaining: 8.39s
469:    learn: 0.0032236        total: 2m 7s    remaining: 8.12s
470:    learn: 0.0032131        total: 2m 7s    remaining: 7.85s
471:    learn: 0.0032056        total: 2m 7s    remaining: 7.58s
472:    learn: 0.0031999        total: 2m 8s    remaining: 7.31s
473:    learn: 0.0031918        total: 2m 8s    remaining: 7.04s
474:    learn: 0.0031866        total: 2m 8s    remaining: 6.77s
475:    learn: 0.0031792        total: 2m 8s    remaining: 6.5s
476:    learn: 0.0031739        total: 2m 9s    remaining: 6.23s
477:    learn: 0.0031633        total: 2m 9s    remaining: 5.96s
478:    learn: 0.0031545        total: 2m 9s    remaining: 5.68s
479:    learn: 0.0031454        total: 2m 9s    remaining: 5.41s
480:    learn: 0.0031383        total: 2m 10s   remaining: 5.14s
481:    learn: 0.0031168        total: 2m 10s   remaining: 4.87s
482:    learn: 0.0031051        total: 2m 10s   remaining: 4.6s
483:    learn: 0.0030938        total: 2m 11s   remaining: 4.33s
484:    learn: 0.0030871        total: 2m 11s   remaining: 4.06s
485:    learn: 0.0030774        total: 2m 11s   remaining: 3.79s
486:    learn: 0.0030671        total: 2m 11s   remaining: 3.52s
487:    learn: 0.0030605        total: 2m 12s   remaining: 3.25s
488:    learn: 0.0030566        total: 2m 12s   remaining: 2.98s
489:    learn: 0.0030462        total: 2m 12s   remaining: 2.71s
490:    learn: 0.0030391        total: 2m 12s   remaining: 2.44s
491:    learn: 0.0030306        total: 2m 13s   remaining: 2.17s
492:    learn: 0.0030233        total: 2m 13s   remaining: 1.9s
493:    learn: 0.0030144        total: 2m 13s   remaining: 1.62s
494:    learn: 0.0030064        total: 2m 14s   remaining: 1.35s
495:    learn: 0.0029968        total: 2m 14s   remaining: 1.08s
496:    learn: 0.0029840        total: 2m 14s   remaining: 812ms
497:    learn: 0.0029727        total: 2m 14s   remaining: 541ms
498:    learn: 0.0029620        total: 2m 15s   remaining: 271ms
499:    learn: 0.0029567        total: 2m 15s   remaining: 0us
Cross-validation scores: [1.      0.99995 1.      0.9997  0.99995]
Mean cross-validation score: 0.99992
```

Interpretation

After training, the model's performance was assessed using a cross-validation procedure. This validation is crucial for ensuring that the risk categories defined by the K-Modes are not only statistically sound but also predictive.

The cross-validation results yield high scores across five folds: `[1.0, 1.0, 1.0, 0.99975, 0.99995]`. The mean cross-validation score, approximately `0.99994`, highlights the model's consistency and accuracy across different subsets of the original binned categories dataset `df_categorical`.

The significance of these results extends to the initial risk categorization performed using the K-Modes clustering method. The alignment of the CatBoost model's predictions with the K-Modes derived clusters supports the accuracy of the initial risk groups identified.

3.1.6 Cluster Evaluation using Silhouette Score

The `silhouette score` is a metric used to evaluate the quality of clusters in a clustering model. It helps determine whether objects within a cluster are well-grouped (intra-cluster cohesion) and adequately separated from other clusters (inter-cluster separation).

Before calculating the silhouette score, categorical data must be transformed into a numerical format so that clustering algorithms can process it properly, typically using "One-Hot" encoding. This transformation allows for the calculation of distances between categorical points and thus evaluates their cluster membership.

The overall silhouette score is the average of the individual scores for all points in each clusters, calculated using the following code:

This code takes around 5 minutes to run, it can be skipped - the conclusion is written below

[672]:

```python
# Enconding categorical data to evaluate the clusters
df_categorical_encoded = OneHotEncoder().fit_transform(df_categorical)
silhouette = silhouette_score(df_categorical_encoded, claims_data['Predicted Risk Cluster'])
davies_bouldin = davies_bouldin_score(df_categorical_encoded.toarray(), claims_data['Predicted Risk Cluster'])

print(f"Silhouette Score: {silhouette}")
print(f"Davies Bouldin Score: {davies_bouldin}")
```

```
Silhouette Score: 0.036988210717677747
Davies Bouldin Score: 4.323812368887318
```

Our K-Modes clustering approach resulted in a silhouette score of 0.04, indicating minimal distinction among the identified risk clusters. This low score suggests that the clusters may not be well-defined or distinct. Despite this, we chose K-Modes for its simplicity and ease of implementation and understanding. This decision reflects a balance between ease of use and interpretability.

See appendix 2 for the distribution of risk variables for each cluster.

3.1.7 Exploration of Alternative Clustering Techniques

In our quest to improve clustering performance, we experimented with a series of advanced analytical techniques: 1. **`DBSCAN` Algorithm:** Initially used to identify dense clusters of data. 2. **Dimensionality Reduction via `PCA`:** Employed to simplify the data structure after DBSCAN produced over 10,000 clusters. 3. **`KMeans` Algorithm:** Applied to the transformed dataset to define clearer, fewer clusters, improving comprehension of the risk clusters.

This approach led to a higher silhouette score of approximately `0.3`, indicating better separation between clusters compared to the K-Modes method. However, the increased complexity of this multi-step process introduced significant challenges: - **Complexity:** Computationally intensive and complex, requiring careful tuning of parameters. - **Interpretability:** Resulting clusters were statistically distinct but less meaningful in practical terms, exhibiting overlap in key risk-defining characteristics such as Vehicle Age, Vehicle Brand, and Vehicle Power.

These overlaps reduced the effectiveness of the clusters in distinctly categorizing risks, crucial for practical applications in the underwriting process.

3.1.8 Rationale for Selecting K-Modes Clustering

Given the complexities and the ambiguous nature of the risk clusters obtained from the alternative method, we opted to continue with the `K-Modes` clustering approach due to: - **Simplicity:** K-Modes is computationally less demanding and easier to implement. - **Actionability:** Clusters generated are more interpretable and directly applicable to real-world risk assessment tasks. - **Practical Relevance:** Despite its lower silhouette score, K-Modes provides a more straightforward interpretation of the data, facilitating easier decision-making in underwriting activities.

3.1.9 Reasonable Principles for our Risk Categorization

Our risk classification system is built upon a foundation of statistical analysis, enhanced by informed judgment. This structured approach allows us to justify the system under key actuarial criteria, ensuring each classification accurately reflects risk levels. By addressing criteria such as accuracy, homogeneity, and credibility, we aim to create a robust, fair, and reliable system for categorizing risks.

Accuracy

Each of our risk cluster demonstrate a clear correlation with expected costs and losses: - **Cluster 0 (Standard)**: This group consists of drivers aged 48 to 51 with recent vehicles (0 to 1 year old) and moderate power (4 to 5), located in Zone C, and driving a vehicle of brand B12. Their claim costs are average for the firm's policyholders, with average losses of $220 (refer to the "Impact of Risk Clusters on Average Losses" table below).

- **Cluster 1 (Sub-Standard)**: This group includes young drivers aged 18 to 25 with older vehicles (12 to 15 years old) and higher power (5 to 6), located in Zone D, and driving a vehicle of brand B1. Historical data indicates that this group has high claim costs, with average losses of $450.
- **Cluster 2 (Preferred)**: This cluster comprises drivers aged 57 to 61 with moderately aged vehicles (4 to 6 years old) and power of 6 to 7, located in Zone E, and driving vehicles of brand B2. This group shows lower claim costs, with average losses below $200.

Homogeneity

The members of each cluster share similar expected claim costs, reducing claim variability: - **Clusters 0, 1, and 2**: Drivers in each cluster present homogeneous risk profiles due to shared characteristics (driver age, vehicle age and power, geographic zone, vehicle brand), enabling the grouping of individuals with similar expected claim costs and minimizing cost dispersion.

Credibility

Our clusters are statistically significant, with each containing a sufficient number of members for the risk categorization algorithm to be robust and reliable. - **Cluster 0**: Contains 56,680 members. - **Cluster 1**: Contains 26,313 members. - **Cluster 2**: Includes 17,007 members.

Also, in our risk classification, certain criteria can serve as **incentives** for policyholders for reducing hazard and ultimately minimizing their insurance costs.

Binned Vehicle Age: Newer vehicles typically have more advanced safety features, potentially lowering the likelihood and severity of accidents. Incentivizing upgrades or regular maintenance of older vehicles could encourage insureds to reduce risk, as well-maintained or safer cars reduce expected losses.

Vehicle Brand: Certain brands may be associated with different levels of reliability or safety. Encouraging insureds to select brands with higher safety standards or proven reliability may also reduce expected losses, reflecting positively on their risk classification.

3.2 2.2. Results of Risk Categorization and Analysis of Clusters

Analysis of these clusters revealed:

- That our risk clusters are well diversified and defined, meaning they do not exhibited overlap in key risk-defining characteristics, such as in `Driver Age`, `Vehicle Age`, `Vehicle Brand`, `Vehicle Power` and `Area`.
- The `Bonus/Malus` variable does not show a significant impact on the clustering process. Also, it is highly correlated with Driver Age (refer to the correlation matrix at the beginning of the code). In that case, this is not a variable we will retain in our final risk classification algorithm.
- In our risk classification process, `Cluster 1` emerges as the riskiest group, exhibiting the highest claims frequency and thus exerting the greatest impact on overall risk from an insurer's perspective. This is followed by `Cluster 0` and `Cluster 2`, based on the comprehensive data analysis of claims frequency and severity.
- The analysis using the CatBoost `model.get_feature_importance()` to extract feature importance confirms our previous intuition. All variables, except for the `Bonus/Malus` variable, show significant importance in predicting the cluster labels. This further supports our decision to exclude the `Bonus/Malus` variable from the final risk classification process, as it does not contribute meaningfully (see appendix 3).

3.2.1 Characteristics of each risk cluster

The table below summarizes the characteristics of each risk cluster identified by the K-Modes clustering algorithm. Each row represents a distinct cluster, with the most frequent values (modes) for the features, as well as the count of data points in each cluster.

[628]:

```
# Count the number of data points in each cluster predicted by the CatBoost␣
  ↪model
predicted_cluster_counts = claims_data['Risk Cluster K-Mode'].value_counts()
predicted_cluster_counts
```

```
# Function to get the most frequent values (mode) for each feature in each␣
↪cluster
def get_cluster_modes(df, cluster_col, feature_cols):
    # Group by the cluster column and calculate the mode for each feature
    cluster_modes = df.groupby(cluster_col)[feature_cols].agg(lambda x: x.
↪mode().iloc[0])
    # Add a column for the count of each cluster
    cluster_modes['Count'] = df[cluster_col].value_counts()
    return cluster_modes

# Columns to analyze
feature_cols = ['Binned DrivAge', 'Binned VehAge', 'Binned VehPower', 'Area',␣
↪'VehBrand']

# Get the most frequent values for each cluster
cluster_modes = get_cluster_modes(claims_data,'Risk Cluster K-Mode',␣
↪feature_cols)

# Display common characteristics for each cluster
cluster_modes
```

[628]:

```
                    Binned DrivAge Binned VehAge Binned VehPower Area  \
Risk Cluster K-Mode
0                        48.0-51.0       0.0-1.0         4.0-5.0    C
1                        18.0-25.0     12.0-15.0         5.0-6.0    D
2                        57.0-61.0       4.0-6.0         6.0-7.0    E

                    VehBrand  Count
Risk Cluster K-Mode
0                        B12  56680
1                         B1  26313
2                         B2  17007
```

Cluster 0: Characterized by middle-aged drivers (48-51 years) in relatively new vehicles (0-1 year old) with moderate vehicle power (4-5). Predominantly located in Area C with vehicles from Brand B12. This cluster has the highest count, suggesting it is the most common risk profile among our data.

Cluster 1: Includes younger drivers (18-25 years) in older vehicles (12-15 years old) with slightly higher vehicle power (5-6). This group is concentrated in Area D with vehicles from Brand B1.

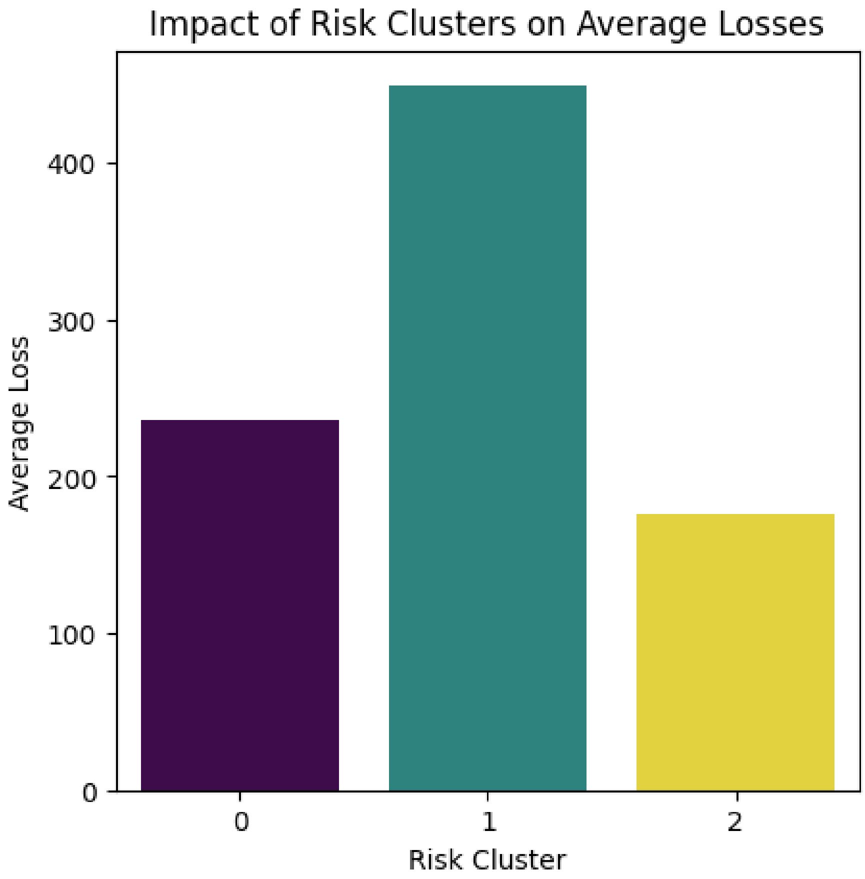
Cluster 2: Comprises older drivers (57-61 years) in vehicles of moderate age (4-6 years old) and higher power (6-7). Located in Area E with vehicles from Brand B2.

These clusters help in tailoring risk management strategies and insurance premiums more accurately by identifying patterns and commonalities within the policyholders of the insurance company.

[629]:

```
# Create a new column for total losses by multiplying frequency and severity of␣
↪claims
```

```
claims_data['Total Loss'] = claims_data['Frequency'] * claims_data['Severity']

# Calculate the mean total losses for each predicted risk cluster
cluster_total_losses = claims_data.groupby('Risk Cluster K-Mode')['Total Loss'].
 ↪mean().reset_index()

# Create a bar plot to visualize the impact of each cluster on total losses to␣
 ↪see the riskiest cluster and the 'safest' cluster
plt.figure(figsize=(5, 5))
bar_plot = sns.barplot(data=cluster_total_losses, x='Risk Cluster K-Mode',␣
 ↪y='Total Loss', hue='Risk Cluster K-Mode', palette='viridis', dodge=False,␣
 ↪legend=False)

plt.title('Impact of Risk Clusters on Average Losses')
plt.xlabel('Risk Cluster')
plt.ylabel('Average Loss')
plt.show()
```



Interpretation

The bar chart clearly indicates that Cluster 1 is the riskiest, confirming our classroom discussions that drivers aged 18-25 tend to exhibit higher risk-taking behaviors. Cluster 0 follows as the second riskiest, with Cluster 2 being the least risky.

[630]:
```
# Create DataFrames based on clusters for further analysis (premium␣
 ↪calculation, etc.)
df_cluster_0 = claims_data[claims_data['Risk Cluster K-Mode'] == 0].copy()
df_cluster_1 = claims_data[claims_data['Risk Cluster K-Mode'] == 1].copy()
df_cluster_2 = claims_data[claims_data['Risk Cluster K-Mode'] == 2].copy()
```

4 3. Predicting total losses

In this section, we seeked to estimate the distribution of total losses for the next year. We first created formulas for the AIC and BIC criteria, then tested the Poisson, Binomial and Negative Binomial distributions for the frequency of losses (i.e. how often we have claims), and the Log-normal, Pareto and Gamma distributions for the severity of losses (i.e. how large the claims are). The lower AIC and BIC values informs us on the distribution that fits best.

We first conducted this analysis on the total distribution. Then, in order to ensure that each cluster has the same distribution as our entire data set, we conducted the same analysis per cluster. This methodology, while perhaps not necessary, allowed us to be the most precise possible, erasing any doubts on the distribution fit of each cluster.

4.1 3.1. Testing distributions

AIC and BIC criterias

[631]:
```
# Functions to calculate AIC and BIC as measures of the goodness of fit for␣
 ↪different distribution models
def calculate_aic(n, ll, k):
    """ Calculate Akaike Information Criterion. """
    return 2 * k - 2 * ll

def calculate_bic(n, ll, k):
    """ Calculate Bayesian Information Criterion. """
    return -2 * ll + k * np.log(n)
```

Function to test distributions

[632]:
```
def distrib_test(x, dist, params, title, dist_type):
    """ Test distribution and calculate AIC and BIC. """
    # Safe log-likelihood calculation
    eps = 1e-10  # A small constant to prevent log(0)
    if dist_type == 'discrete':
        log_likelihood = np.sum(np.log(dist.pmf(x, *params) + eps))
    elif dist_type == 'continuous':
        log_likelihood = np.sum(np.log(dist.pdf(x, *params) + eps))
    else:
        raise ValueError("dist_type must be 'discrete' or 'continuous'")

    n = len(x)
    k = len(params)
    aic = calculate_aic(n, log_likelihood, k)
    bic = calculate_bic(n, log_likelihood, k)
    print(f"AIC for {title}: {aic}")
    print(f"BIC for {title}: {bic}")
```

Total claims

[633]:

```
# Filter out zero claim amounts for severity analysis
claim_counts = claims_data['Frequency']
non_zero_claims = claims_data[claims_data['Severity'] > 0]['Severity']
```

Frequency

[634]:

```
# Poisson Distribution Fitting for Claim Frequency
lambda_poisson = np.mean(claim_counts)
params_poisson = [lambda_poisson]
distrib_test(claim_counts, poisson, params_poisson, 'Fit with Poisson␣
 ↪Distribution', dist_type='discrete')

# Binomial Distribution Fitting for Claim Frequency
n_trials = 1  # Number of trials (1 for each policyholder)
p_success = np.mean(claim_counts) / n_trials
params_binom = [n_trials, p_success]
distrib_test(claim_counts, binom, params_binom, 'Fit with Binomial␣
 ↪Distribution', dist_type='discrete')

# Negative Binomial Distribution Fitting for Claim Frequency
mean_claims = np.mean(claim_counts)
var_claims = np.var(claim_counts, ddof=1) # Use ddof=1 for sample variance
if var_claims > mean_claims:
    r_negbin = (mean_claims ** 2) / (var_claims - mean_claims)
    p_negbin = r_negbin / (r_negbin + mean_claims)
    params_negbin = [r_negbin, p_negbin]

    # Fit check with our distribution testing function
    distrib_test(claim_counts, nbinom, params_negbin, 'Fit with Negative␣
 ↪Binomial Distribution', dist_type='discrete')
```

```
AIC for Fit with Poisson Distribution: 133832.56732566294
BIC for Fit with Poisson Distribution: 133842.08025112792
AIC for Fit with Binomial Distribution: 141115.09246842703
BIC for Fit with Binomial Distribution: 141134.11831935696
AIC for Fit with Negative Binomial Distribution: 120200.78647043131
BIC for Fit with Negative Binomial Distribution: 120219.81232136126
```

Results

The Negative Binomial Distribution gives the lowest values for the AIC and BIC criteria. This is the best distribution fit for the frequency of losses.

Severity

[635]:

```
# Log-normal fitting for severity
shape, loc, scale = lognorm.fit(non_zero_claims)
params_lognorm = [shape, loc, scale]
distrib_test(non_zero_claims, lognorm, params_lognorm, 'Fit with Log-normal␣
 ↪Distribution', dist_type='continuous')

# Pareto fitting
b, loc_pareto, scale_pareto = pareto.fit(non_zero_claims)
params_pareto = [b, loc_pareto, scale_pareto]
distrib_test(non_zero_claims, pareto, params_pareto, 'Fit with Pareto␣
 ↪Distribution', dist_type='continuous')

# Gamma fitting
alpha, loc_gamma, beta = gamma.fit(non_zero_claims)
params_gamma = [alpha, loc_gamma, beta]
distrib_test(non_zero_claims, gamma, params_gamma, 'Fit with Gamma␣
 ↪Distribution', dist_type='continuous')
```

```
AIC for Fit with Log-normal Distribution: 61011.03355955744
BIC for Fit with Log-normal Distribution: 61029.653310400405
AIC for Fit with Pareto Distribution: 61251.71150028282
BIC for Fit with Pareto Distribution: 61270.33125112578
AIC for Fit with Gamma Distribution: 167970.14982245426
BIC for Fit with Gamma Distribution: 167988.76957329724
```

Results

The Log-normal Distribution gives the lowest values for the AIC and BIC criteria. This is the best distribution fit for the severity of losses.

Cluster 0

[636]:

```
# Filter out zero claim amounts for severity
claim_counts_0 = df_cluster_0['Frequency']
non_zero_claims_0 = df_cluster_0[df_cluster_0['Severity'] > 0]['Severity']
```

Frequency

[637]:

```
# Poisson Distribution Fitting for Claim Frequency (Cluster 0)
lambda_poisson_0 = np.mean(claim_counts_0)
params_poisson_0 = [lambda_poisson_0]
distrib_test(claim_counts_0, poisson, params_poisson_0, 'Fit with Poisson␣
 ↪Distribution', dist_type='discrete')
```

```
# Binomial Distribution Fitting for Claim Frequency (Cluster 0)
n_trials_0 = 1  # Number of trials (1 for each policyholder)
p_success_0 = np.mean(claim_counts_0) / n_trials_0
params_binom_0 = [n_trials_0, p_success_0]
distrib_test(claim_counts_0, binom, params_binom_0, 'Fit with Binomial␣
 ↪Distribution', dist_type='discrete')

# Negative Binomial Distribution Fitting for Claim Frequency (Cluster 0)
mean_claims_0 = np.mean(claim_counts_0)
var_claims_0 = np.var(claim_counts_0, ddof=1)  # Use ddof=1 for sample variance
if var_claims_0 > mean_claims_0:
    r_negbin_0 = (mean_claims_0 ** 2) / (var_claims_0 - mean_claims_0)
    p_negbin_0 = r_negbin_0 / (r_negbin_0 + mean_claims_0)
    params_negbin_0 = [r_negbin_0, p_negbin_0]

    # Fit check with our distribution testing function
    distrib_test(claim_counts_0, nbinom, params_negbin_0, 'Fit with Negative␣
 ↪Binomial Distribution', dist_type='discrete')
```

```
AIC for Fit with Poisson Distribution: 69317.96509096696
BIC for Fit with Poisson Distribution: 69326.91026766076
AIC for Fit with Binomial Distribution: 72651.72826332867
BIC for Fit with Binomial Distribution: 72669.61861671628
AIC for Fit with Negative Binomial Distribution: 61699.27596738552
BIC for Fit with Negative Binomial Distribution: 61717.166320773125
```

Severity

[638]:

```
# Log-normal fitting
shape_0, loc_0, scale_0 = lognorm.fit(non_zero_claims_0)
params_lognorm_0 = [shape_0, loc_0, scale_0]
distrib_test(non_zero_claims_0, lognorm, params_lognorm_0, 'Fit with Log-normal␣
 ↪Distribution', dist_type='continuous')

# Pareto fitting
b_pareto_0, loc_pareto_0, scale_pareto_0 = pareto.fit(non_zero_claims_0)
params_pareto_0 = [b_pareto_0, loc_pareto_0, scale_pareto_0]
distrib_test(non_zero_claims_0, pareto, params_pareto_0, 'Fit with Pareto␣
 ↪Distribution', dist_type='continuous')

# Gamma fitting
alpha_0, loc_gamme_0, beta_0 = gamma.fit(non_zero_claims_0)
params_gamma_0 = [alpha_0, loc_gamme_0, beta_0]
distrib_test(non_zero_claims_0, gamma, params_gamma_0, 'Fit with Gamma␣
 ↪Distribution', dist_type='continuous')
```

```
AIC for Fit with Log-normal Distribution: 30071.413377883982
BIC for Fit with Log-normal Distribution: 30087.908325496282
```

```
AIC for Fit with Pareto Distribution: 30207.66167850583
BIC for Fit with Pareto Distribution: 30224.15662611813
AIC for Fit with Gamma Distribution: 82775.57817446705
BIC for Fit with Gamma Distribution: 82792.07312207935
```

Cluster 1

[639]:

```
# Filter out zero claim amounts for severity
claim_counts_1 = df_cluster_1['Frequency']
non_zero_claims_1 = df_cluster_1[df_cluster_1['Severity'] > 0]['Severity']
```

Frequency

[640]:

```
# Poisson Distribution Fitting for Claim Frequency (Cluster 1)
lambda_poisson_1 = np.mean(claim_counts_1)
params_poisson_1 = [lambda_poisson_1]
distrib_test(claim_counts_1, poisson, params_poisson_1, 'Fit with Poisson␣
  ↪Distribution', dist_type='discrete')

# Binomial Distribution Fitting for Claim Frequency (Cluster 1)
n_trials_1 = 1  # Number of trials (1 for each policyholder)
p_success_1 = np.mean(claim_counts_1) / n_trials_1
params_binom_1 = [n_trials_1, p_success_1]
distrib_test(claim_counts_1, binom, params_binom_1, 'Fit with Binomial␣
  ↪Distribution', dist_type='discrete')

# Negative Binomial Distribution Fitting for Claim Frequency (Cluster 1)
mean_claims_1 = np.mean(claim_counts_1)
var_claims_1 = np.var(claim_counts_1, ddof=1)  # Use ddof=1 for sample variance
if var_claims_1 > mean_claims_1:
    r_negbin_1 = (mean_claims_1 ** 2) / (var_claims_1 - mean_claims_1)
    p_negbin_1 = r_negbin_1 / (r_negbin_1 + mean_claims_1)
    params_negbin_1 = [r_negbin_1, p_negbin_1]

    # Fit check with our distribution testing function
    distrib_test(claim_counts_1, nbinom, params_negbin_1, 'Fit with Negative␣
  ↪Binomial Distribution', dist_type='discrete')
```

```
AIC for Fit with Poisson Distribution: 41980.71341737464
BIC for Fit with Poisson Distribution: 41988.89123576726
AIC for Fit with Binomial Distribution: 44396.52139047038
BIC for Fit with Binomial Distribution: 44412.877027255614
AIC for Fit with Negative Binomial Distribution: 37367.02169802738
BIC for Fit with Negative Binomial Distribution: 37383.37733481262
```

Severity

[641]:

```
# Log-normal fitting
shape_1, loc_1, scale_1 = lognorm.fit(non_zero_claims_1)
```

```
params_lognorm_1 = [shape_1, loc_1, scale_1]
distrib_test(non_zero_claims_1, lognorm, params_lognorm_1, 'Fit with Log-normal␣
 ↪Distribution', dist_type='continuous')

# Pareto fitting
b_pareto_1, loc_pareto_1, scale_pareto_1 = pareto.fit(non_zero_claims_1)
params_pareto_1 = [b_pareto_1, loc_pareto_1, scale_pareto_1]
distrib_test(non_zero_claims_1, pareto, params_pareto_1, 'Fit with Pareto␣
 ↪Distribution', dist_type='continuous')

# Gamma fitting
alpha_1, loc_gamma_1, beta_1 = gamma.fit(non_zero_claims_1)
params_gamma_1 = [alpha_1, loc_gamma_1, beta_1]
distrib_test(non_zero_claims_1, gamma, params_gamma_1, 'Fit with Gamma␣
 ↪Distribution', dist_type='continuous')
```

```
AIC for Fit with Log-normal Distribution: 19324.554551397487
BIC for Fit with Log-normal Distribution: 19339.712714545185
AIC for Fit with Pareto Distribution: 19403.095097762525
BIC for Fit with Pareto Distribution: 19418.253260910224
AIC for Fit with Gamma Distribution: 53015.516666628304
BIC for Fit with Gamma Distribution: 53030.674829776
```

Cluster 2

[642]:
```
# Filter out zero claim amounts for severity
claim_counts_2 = df_cluster_2['Frequency']
non_zero_claims_2 = df_cluster_2[df_cluster_2['Severity'] > 0]['Severity']
```

Frequency

[643]:
```
# Poisson Distribution Fitting for Claim Frequency (Cluster 2)
lambda_poisson_2 = np.mean(claim_counts_2)
params_poisson_2 = [lambda_poisson_2]
distrib_test(claim_counts_2, poisson, params_poisson_2, 'Fit with Poisson␣
 ↪Distribution', dist_type='discrete')

# Binomial Distribution Fitting for Claim Frequency (Cluster 2)
n_trials_2 = 1  # Number of trials (1 for each policyholder)
p_success_2 = np.mean(claim_counts_2) / n_trials_2
params_binom_2 = [n_trials_2, p_success_2]
distrib_test(claim_counts_2, binom, params_binom_2, 'Fit with Binomial␣
 ↪Distribution', dist_type='discrete')

# Negative Binomial Distribution Fitting for Claim Frequency (Cluster 2)
mean_claims_2 = np.mean(claim_counts_2)
var_claims_2 = np.var(claim_counts_2, ddof=1)  # Use ddof=1 for sample variance
if var_claims_2 > mean_claims_2:
    r_negbin_2 = (mean_claims_2 ** 2) / (var_claims_2 - mean_claims_2)
    p_negbin_2 = r_negbin_2 / (r_negbin_2 + mean_claims_2)
    params_negbin_2 = [r_negbin_2, p_negbin_2]

    # Fit check with our distribution testing function
    distrib_test(claim_counts_2, nbinom, params_negbin_2, 'Fit with Negative␣
 ↪Binomial Distribution', dist_type='discrete')
```

```
AIC for Fit with Poisson Distribution: 22518.747894799446
BIC for Fit with Poisson Distribution: 22526.489275102438
AIC for Fit with Binomial Distribution: 24085.23869630995
BIC for Fit with Binomial Distribution: 24100.721456915933
AIC for Fit with Negative Binomial Distribution: 20920.801717520935
BIC for Fit with Negative Binomial Distribution: 20936.28447812692
```

Severity

[644]:

```
# Log-normal fitting
shape_2, loc_2, scale_2 = lognorm.fit(non_zero_claims_2)
params_lognorm_2 = [shape_2, loc_2, scale_2]
distrib_test(non_zero_claims_2, lognorm, params_lognorm_2, 'Fit with Log-normal␣
 ↪Distribution', dist_type='continuous')

# Pareto fitting
b_pareto_2, loc_pareto_2, scale_pareto_2 = pareto.fit(non_zero_claims_2)
params_pareto_2 = [b_pareto_2, loc_pareto_2, scale_pareto_2]
distrib_test(non_zero_claims_2, pareto, params_pareto_2, 'Fit with Pareto␣
 ↪Distribution', dist_type='continuous')

# Gamma fitting
alpha_2, loc_gamma_2, beta_2 = gamma.fit(non_zero_claims_2)
params_gamma_2 = [alpha_2, loc_gamma_2, beta_2]
distrib_test(non_zero_claims_2, gamma, params_gamma_2, 'Fit with Gamma␣
 ↪Distribution', dist_type='continuous')
```

```
AIC for Fit with Log-normal Distribution: 11618.868614209274
BIC for Fit with Log-normal Distribution: 11632.538949277749
AIC for Fit with Pareto Distribution: 11633.201286699441
BIC for Fit with Pareto Distribution: 11646.871621767916
AIC for Fit with Gamma Distribution: 32192.25048578799
BIC for Fit with Gamma Distribution: 32205.920820856463
```

Results

For each cluster, the Negative Binomial distribution gives the lowest AIC and BIC criteria for the frequency of losses, and the Log-normal distribution gives the lowest values for the severity of losses, thus giving us the same results as for total losses. However, it's interesting to note that for the severity of losses, the Pareto and Log-normal fits give AIC and BIC criteria that are rather close. To push our analysis further, we could conduct more sophisticated distribution tests to investigate more.

5 4. Premium determination

Now that the distributions are known, we can proceed to compute the premiums for the next year. To do so, we first need to know the total losses for the next year, which we have done using a monte carlo simulation. We then computed the premiums for total claims and for each cluster, alwhile ensuring that the probability of observing claims that are larger than our premiums does not exceed a threshold of 0.5%.

5.1 4.1 Monte carlo simulation for total losses

[645]:

```
# Monte carlo simulation for ALL CLAIMS (Entire Data Set)

np.random.seed(23) # Set seed for reproducibility

# Parameters
n_simulations = 100
n_policies = len(claims_data)

# Create a list to store each simulation's DataFrame for faster concatenation
simulation_dfs = []

# Monte Carlo simulation loop
for i in range(n_simulations):
    # Simulate number of claims for all policies at once (Frequency per year␣
 ↪per policy)
    frequency = nbinom.rvs(r_negbin, p_negbin, size=n_policies)

    # Create a severity matrix for policies with claims
    max_claim = frequency.max()  # Max number of claims across policies
    severity_matrix = np.zeros((n_policies, max_claim))

    # Generate severities per policy based on the number of claims (frequency)
    for policy_idx in range(n_policies):
        claim_count = frequency[policy_idx]
        if claim_count > 0:
            severity_matrix[policy_idx, :claim_count] = lognorm.rvs(
                shape, loc=0, scale=scale, size=claim_count
            )

    # We want to sum up all of the claims to get the premium, because premium =␣
 ↪severity x frequency
    Total_claim_severity = np.sum(severity_matrix, axis=1)

    # Create a temporary DataFrame for this simulation
    temp_df = pd.DataFrame({
        'Policy': np.arange(n_policies),
        'Frequency': frequency,
        'Premium_per_policy': Total_claim_severity,
        'Simulation': i
    })

    # Append the DataFrame to the list
    simulation_dfs.append(temp_df)

# Concatenate all temporary DataFrames at once for better performance
simulation_results = pd.concat(simulation_dfs, ignore_index=True)


# Compute the mean of all claims (total loss) for each simulation
premium_per_simulation = simulation_results.
 ↪groupby('Simulation')['Premium_per_policy'].mean()

# Calculate the average premium across all simulations
average_premium = premium_per_simulation.mean()
print(f"The premium for the total loss distribution is: {average_premium:.2f}$")
```

```
The premium for the total loss distribution is: 200.95$
```

5.1.1 Calculation of adjusted premium

Here, we compute the premium that respects the 0.5% condition. We also compute a simple hit measure, that validates that the premium respects the condition.

[646]:

```
# Calculate the 99.5th percentile of all total claim severities across all␣
 ↪policies and simulations
adjusted_premium = scoreatpercentile(simulation_results['Premium_per_policy'],␣
 ↪99.5)

# Output the adjusted premium
print(f"Adjusted premium for total losses is: {adjusted_premium:.2f}$")

# Compute hit measure : 1 if Premium_per_policy > adjusted_premium_0, otherwise␣
 ↪0
simulation_results['Hit_Test'] = simulation_results['Premium_per_policy'].
 ↪apply(lambda x: 1 if x > adjusted_premium else 0)

# Calculate the percentage of hits
total_hits = simulation_results['Hit_Test'].sum()
total_records = len(simulation_results)
hit_percentage = (total_hits / total_records) * 100

print("Percentage of Hits (Total claims > adjusted premium): {:.2f}%".
 ↪format(hit_percentage))
```

```
Adjusted premium for total losses is: 8561.26$
Percentage of Hits (Total claims > adjusted premium): 0.50%
```

5.2 4.2 Monte carlo simulation per cluster

Cluster 0

[647]:

```
# Monte carlo simulation for cluster 0

np.random.seed(24) # Set seed for reproducibility

# Parameters
n_simulations_0 = 100
n_policies_0 = len(df_cluster_0)

# Create a list to store each simulation's DataFrame for faster concatenation
simulation_dfs_0 = []

# Monte Carlo simulation loop
for i in range(n_simulations_0):
    # Simulate number of claims for all policies at once (Frequency_0 per year␣
 ↪per policy)
    frequency_0 = nbinom.rvs(r_negbin_0, p_negbin_0, size=n_policies_0)

    # Create a severity matrix for policies with claims
    max_claim_0 = frequency_0.max()  # Max number of claims across policies for␣
 ↪Cluster 0
    severity_matrix_0 = np.zeros((n_policies_0, max_claim_0))

    # Generate severities per policy based on the number of claims (frequency_0)
    for policy_idx in range(n_policies_0):
        claim_count_0 = frequency_0[policy_idx]
        if claim_count_0 > 0:
            severity_matrix_0[policy_idx, :claim_count_0] = lognorm.rvs(
                shape_0, loc=0, scale=scale_0, size=claim_count_0
            )

    # We want to sum up all of the claims to get the premium, because premium =␣
 ↪severity x frequency
    Total_claim_severity_0 = np.sum(severity_matrix_0, axis=1)

    # Create a temporary DataFrame for this simulation
    temp_df_0 = pd.DataFrame({
        'Policy': np.arange(n_policies_0),
        'Frequency': frequency_0,
        'Premium_per_policy': Total_claim_severity_0,
        'Simulation': i
    })

    # Append the DataFrame to the list
    simulation_dfs_0.append(temp_df_0)
```

```
# Concatenate all temporary DataFrames at once for better performance
simulation_results_0 = pd.concat(simulation_dfs_0, ignore_index=True)

# Compute the mean of all claims (total loss) for each simulation
premium_per_simulation_0 = simulation_results_0.
 ↪groupby('Simulation')['Premium_per_policy'].mean().reset_index()

# Calculate the average premium across all simulations
average_premium_0 = (premium_per_simulation_0.mean()).iloc[1]

print(f"The premium for the Cluster 0 is: {average_premium_0:.2f}$")
```

```
The premium for the Cluster 0 is: 188.93$
```

Calculation of the adjusted premium

[648]:
```
# Calculate the total losses for the entire data set and for cluster 0
all_total_losses = simulation_results['Premium_per_policy'].sum()
total_losses_0 = simulation_results_0 ['Premium_per_policy'].sum()

# Calculate the proportion of total losses for cluster 0
proportion_0 = total_losses_0 / all_total_losses

# Calculate the adjusted premium for cluster 0
adjusted_premium_0 = adjusted_premium * proportion_0

print(f"The adjusted premium for cluster 0 is: {adjusted_premium_0:.2f}$")
```

```
The adjusted premium for cluster 0 is: 4562.42$
```

Cluster 1

[649]:
```
# Monte carlo simulation for cluster 1

np.random.seed(25) # Set seed for reproducibility

# Parameters
n_simulations_1 = 100
n_policies_1 = len(df_cluster_1)

# Create a list to store each simulation's DataFrame for faster concatenation
simulation_dfs_1 = []

# Monte Carlo simulation loop
for i in range(n_simulations_1):
    # Simulate number of claims for all policies at once (Frequency_1 per year␣
 ↪per policy)
    frequency_1 = nbinom.rvs(r_negbin_1, p_negbin_1, size=n_policies_1)
    # Create a severity matrix for policies with claims
    max_claim_1 = frequency_1.max()  # Max number of claims across policies for␣
 ↪cluster 1
    severity_matrix_1 = np.zeros((n_policies_1, max_claim_1))

    # Generate severities per policy based on the number of claims (frequency_1)
    for policy_idx in range(n_policies_1):
        claim_count_1 = frequency_1[policy_idx]
        if claim_count_1 > 0:
            severity_matrix_1[policy_idx, :claim_count_1] = lognorm.rvs(
                shape_1, loc=0, scale=scale_1, size=claim_count_1
            )

    # We want to sum up all of the claims to get the premium, because premium =␣
 ↪severity x frequency
    Total_claim_severity_1 = np.sum(severity_matrix_1, axis=1)

    # Create a temporary DataFrame for this simulation
    temp_df_1 = pd.DataFrame({
        'Policy': np.arange(n_policies_1),
        'Frequency': frequency_1,
        'Premium_per_policy': Total_claim_severity_1,
        'Simulation': i
    })

    # Append the DataFrame to the list
    simulation_dfs_1.append(temp_df_1)

# Concatenate all temporary DataFrames at once for better performance
simulation_results_1 = pd.concat(simulation_dfs_1, ignore_index=True)


# Compute the mean of all claims (total loss) for each simulation
premium_per_simulation_1 = simulation_results_1.
 ↪groupby('Simulation')['Premium_per_policy'].mean().reset_index()

# Calculate the average premium across all simulations
average_premium_1 = premium_per_simulation_1.mean().iloc[1]
print(f"The premium for the Cluster 1 is: {average_premium_1:.2f}$")
```

```
The premium for the Cluster 1 is: 259.45$
```

Calulculation of the adjusted premium

[650]:
```
# Calculate the total losses for cluster 1
total_losses_1 = simulation_results_1['Premium_per_policy'].sum()
```

```
# Calculate the proportion of total losses for cluster 1
proportion_1 = total_losses_1 / all_total_losses

# Calculate the adjusted premium for cluster 1
adjusted_premium_1 = adjusted_premium * proportion_1

print(f"The adjusted premium for cluster 1 is: {adjusted_premium_1:.2f}$")
```

```
The adjusted premium for cluster 1 is: 2908.52$
```

Cluster 2

[651]:
```
# Monte carlo simulation for cluster 2

np.random.seed(26) # Set seed for reproducibility

# Parameters
n_simulations_2 = 100
n_policies_2 = len(df_cluster_2)

# Create a list to store each simulation's DataFrame for faster concatenation
simulation_dfs_2 = []

# Monte Carlo simulation loop
for i in range(n_simulations_2):
    # Simulate number of claims for all policies at once (Frequency_2 per year␣
 ↪per policy)
    frequency_2 = nbinom.rvs(r_negbin_2, p_negbin_2, size=n_policies_2)

    # Create a severity matrix for policies with claims
    max_claim_2 = frequency_2.max()  # Max number of claims across policies for␣
 ↪cluster 2
    severity_matrix_2 = np.zeros((n_policies_2, max_claim_2))

    # Generate severities per policy based on the number of claims (frequency_2)
    for policy_idx in range(n_policies_2):
        claim_count_2 = frequency_2[policy_idx]
        if claim_count_2 > 0:
            severity_matrix_2[policy_idx, :claim_count_2] = lognorm.rvs(
                shape_2, loc=0, scale=scale_2, size=claim_count_2
            )

    # We want to sum up all of the claims to get the premium, because premium =␣
 ↪severity x frequency
    Total_claim_severity_2 = np.sum(severity_matrix_2, axis=1)

    # Create a temporary DataFrame for this simulation
    temp_df_2 = pd.DataFrame({
        'Policy': np.arange(n_policies_2),
        'Frequency': frequency_2,
        'Premium_per_policy': Total_claim_severity_2,
        'Simulation': i
    })

    # Append the DataFrame to the list
    simulation_dfs_2.append(temp_df_2)

# Concatenate all temporary DataFrames at once for better performance
simulation_results_2 = pd.concat(simulation_dfs_2, ignore_index=True)


# Compute the mean of all claims (total loss) for each simulation
premium_per_simulation_2 = simulation_results_2.
 ↪groupby('Simulation')['Premium_per_policy'].mean().reset_index()

# Calculate the average premium across all simulations
average_premium_2 = premium_per_simulation_2.mean().iloc[1]
print(f"The premium for the Cluster 2 is: {average_premium_2:.2f}$")
```

```
The premium for the Cluster 2 is: 162.78$
```

Calulculation of the adjusted premium

[652]:

```
# Calculate the total losses for cluster 2
total_losses_2 = simulation_results_2['Premium_per_policy'].sum()

# Calculate the proportion of total losses for cluster 2
proportion_2 = total_losses_2 / all_total_losses

# Calculate the adjusted premium for cluster 2
adjusted_premium_2 = adjusted_premium * proportion_2

print(f"The adjusted premium for cluster 2 is: {adjusted_premium_2:.2f}$")
```

```
The adjusted premium for cluster 2 is: 1179.48$
```

Interpretation

We obtained a premium of $188.93 for Cluster 0, $259.45 for Cluster 1, and $162.78 for Cluster 2. These results indicate an order of risk from most to least risky: Cluster 1, Cluster 0, and Cluster 2, which aligns with our cluster analysis findings. While these premiums are lower than those observed in reality, they reflect only the calculated risk-based premium, excluding additional costs, such as operational expenses, typically included in real-world premiums.

When we adjusted premiums to ensure claims did not exceed the set amount more than 0.5% of the time, the premiums increased significantly. Interestingly, Cluster 0's adjusted premium surpassed that of Cluster 1, which is counter-intuitive. This discrepancy could stem from our methodology; we calculate the total losses for each cluster rather than the average. Consequently, although Cluster

1 has a higher average loss (indicating higher risk), Cluster 0's higher frequency of claims results in a greater proportion of total losses, leading to a higher adjusted premium for Cluster 0.

In practice, some insurance companies use redlining during underwriting, excluding extremely high losses from premium calculations. This practice, while potentially discriminatory, could reduce adjusted premiums and meet the 0.5% condition at a lower premium by eliminating the highest outlier losses.

6 5. Evaluation

[653]:

```
# Data frame containing the real claims data
actual_claims = pd.read_csv('claim_data_group4_2025.csv')
```

6.1 5.1 Implementation of the k-modes clustering method on actual data

[654]:

```
# List of features to bin and number of bins for each feature
features_actual = ['DrivAge', 'VehAge','VehPower']
n_bins_actual = 10

# Define the numbers of bins for specific features (DrivAge), to get more␣
 ↪granularity into the bins classification
specific_bins_actual = {'DrivAge': 20}  # Par exemple, 30 bins pour DrivAge␣
 ↪pour plus de granularité

# Call the function
bin_labels_dict = quantile_binning_and_apply(actual_claims, features_actual,␣
 ↪n_bins_actual, specific_bins_actual)

# k-modes function
additional_categorical_columns_actual = ['Area', 'VehBrand']
actual_claims_clusters = apply_k_modes_clustering(
    claims_data=actual_claims,
    features=features_actual,
    additional_categorical_columns=additional_categorical_columns_actual,
    n_clusters=3,  # Number of clusters you want
    random_state=43  # Seed for reproducibility
)
```

[655]:

```
# Columns to analyze
feature_cols = ['Binned DrivAge', 'Binned VehAge', 'Binned VehPower', 'Area',␣
 ↪'VehBrand']

# Get the most frequent values for each cluster
cluster_modes = get_cluster_modes(actual_claims,'Risk Cluster K-Mode',␣
 ↪feature_cols)
cluster_modes
```

[655]:

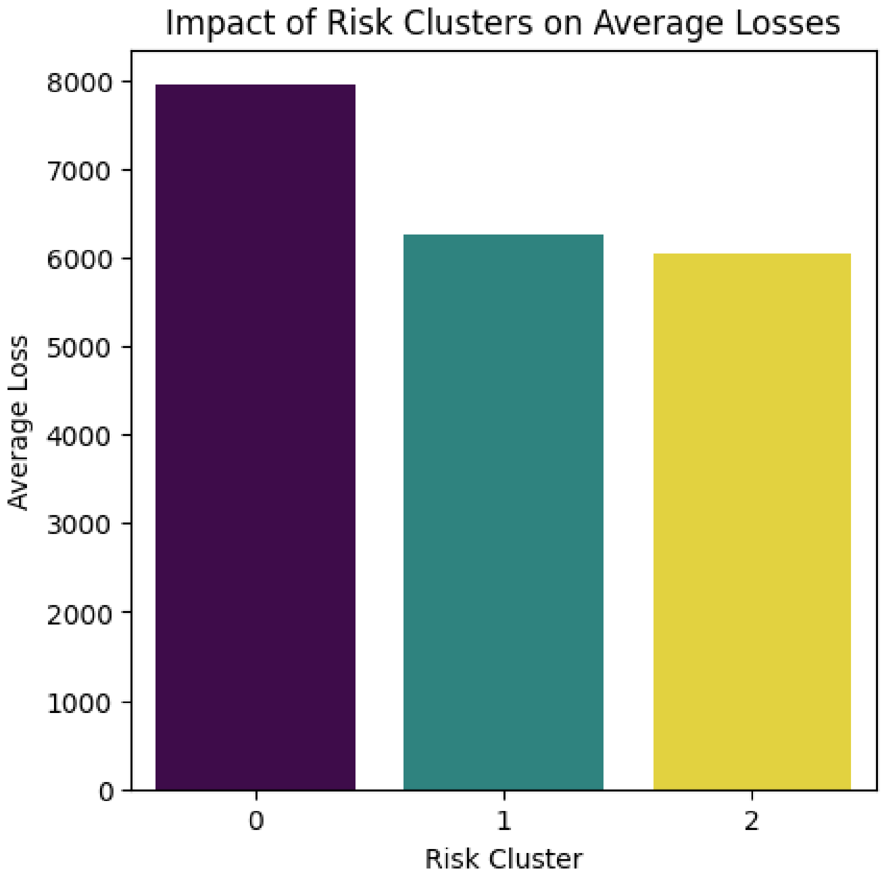
```
                      Binned DrivAge Binned VehAge Binned VehPower Area  \
Risk Cluster K-Mode
0                          46.0-49.0       0.0-1.0         4.0-5.0    C
1                          18.0-25.0     12.0-15.0         5.0-6.0    D
2                          57.0-61.0       4.0-6.0         6.0-7.0    E

                      VehBrand  Count
Risk Cluster K-Mode
0                          B12  56276
1                           B1  26547
2                           B2  17177
```

Interpretation

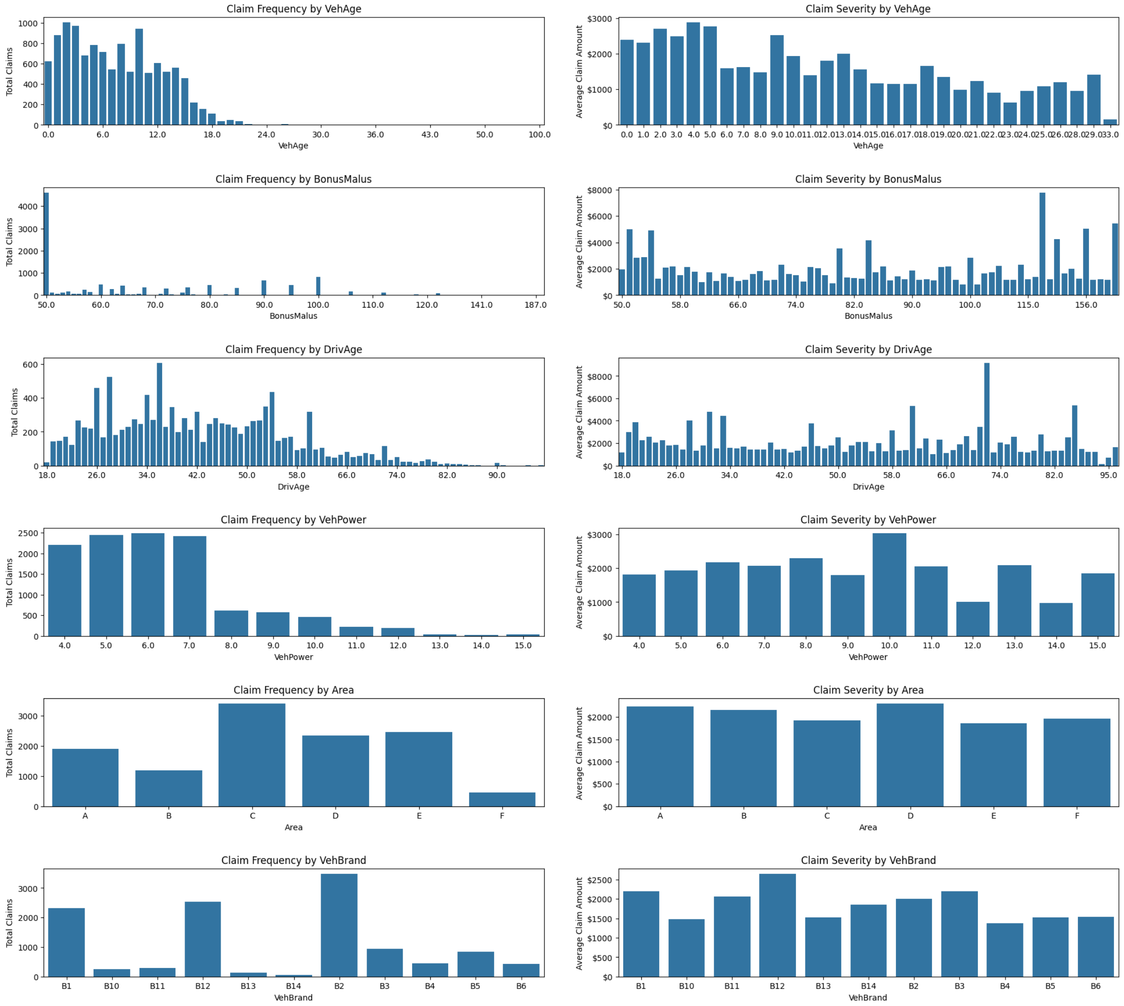
The clusters for the actual data reveal to be the exact same as the clusters of the simulated data, indicating that our clustering method is accurate.

[656]:

```
actual_claims['Frequency'] = actual_claims['ClaimNb'] /␣
  ↪actual_claims['Exposure']
actual_claims['Severity'] = actual_claims['ClaimAmount'] /␣
  ↪actual_claims['ClaimNb']

# Create a new column for total losses by multiplying frequency and severity of␣
  ↪claims
actual_claims['Total Loss'] = actual_claims['Frequency'] *␣
  ↪actual_claims['Severity']

# Calculate the mean total losses for each predicted risk cluster
cluster_total_losses = actual_claims.groupby('Risk Cluster K-Mode')['Total␣
  ↪Loss'].mean().reset_index()

# Create a bar plot to visualize the impact of each cluster on total losses to␣
  ↪see the riskiest cluster and the 'safest' cluster
plt.figure(figsize=(5, 5))
bar_plot = sns.barplot(data=cluster_total_losses, x='Risk Cluster K-Mode',␣
  ↪y='Total Loss', hue='Risk Cluster K-Mode', palette='viridis', dodge=False,␣
  ↪legend=False)

plt.title('Impact of Risk Clusters on Average Losses')
plt.xlabel('Risk Cluster')
plt.ylabel('Average Loss')
plt.show()
```



```
[657]: # Variables to plot frequency and severity of claims by policyholder␣
         ↪characteristics
       variables = ['VehAge', 'BonusMalus', 'DrivAge', 'VehPower', 'Area', 'VehBrand']

       fig, axes = plt.subplots(nrows=len(variables), ncols=2, figsize=(20, 3 *␣
         ↪len(variables)))

       # Loop through each variable and apply the plotting function on subplots
       for idx, var in enumerate(variables):
           plot_variable(actual_claims, var, axes[idx, 0], axes[idx, 1])

       plt.tight_layout(pad=3.0)
       plt.show()
```



Interpretation

This chart indicates that Cluster 0, which has the highest average loss, appears to be the riskiest cluster in the current data set. Interestingly, Cluster 1 held this position in last year's data. However, this shift aligns with typical fluctuations observed in yearly analyses.

For example, in the current data, the claim frequency for the `DrivAge` variable is concentrated around age 34, with claim severity remaining stable, except for a peak near age 70. In contrast, last year's data showed the highest claim frequency around ages 18 and 45, with significant spikes in claim severity at ages 18 and 82.

Aside from 'DriveAge', the other variables relevant to our clusters exhibit similar patterns in frequency and severity when compared to last year's dataset, reinforcing the consistency across most variables despite some shifts in cluster risk levels.

6.2 5.2 Premium determination

[658]:

```
# Create DataFrames based on clusters for further analysis (premium␣
 ↪calculation, etc.)
real_cluster_0 = actual_claims[actual_claims['Risk Cluster K-Mode'] == 0].copy()
real_cluster_1 = actual_claims[actual_claims['Risk Cluster K-Mode'] == 1].copy()
real_cluster_2 = actual_claims[actual_claims['Risk Cluster K-Mode'] == 2].copy()

# Create Frequency column (nb of claims per year)
real_cluster_0['Frequency'] = real_cluster_0['ClaimNb'] /␣
 ↪real_cluster_0['Exposure'] #Number of claims per year
real_cluster_1['Frequency'] = real_cluster_1['ClaimNb'] /␣
 ↪real_cluster_1['Exposure']
real_cluster_2['Frequency'] = real_cluster_2['ClaimNb'] /␣
 ↪real_cluster_2['Exposure']

# Create Severity column (amount per claim)
real_cluster_0['Severity'] = real_cluster_0['ClaimAmount'] /␣
 ↪real_cluster_0['ClaimNb']
real_cluster_1['Severity'] = real_cluster_1['ClaimAmount'] /␣
 ↪real_cluster_1['ClaimNb']
real_cluster_2['Severity'] = real_cluster_2['ClaimAmount'] /␣
 ↪real_cluster_2['ClaimNb']
```

Cluster 0

[659]:

```
# Cluster 0
real_cluster_0['Premium_per_policy'] = real_cluster_0['Frequency'] *␣
 ↪real_cluster_0['Severity']
real_cluster_0['Premium_per_policy'] = real_cluster_0['Premium_per_policy'].
 ↪fillna(0)
real_premium_0 = real_cluster_0['Premium_per_policy'].mean()

print(f"The real premium for cluster 0 is: {real_premium_0:.2f}$")
```

```
The real premium for cluster 0 is: 258.82$
```

Cluster 1

[660]:

```
# Cluster 1
real_cluster_1['Premium_per_policy'] = real_cluster_1['Frequency'] *␣
 ↪real_cluster_1['Severity']
real_cluster_1['Premium_per_policy'] = real_cluster_1['Premium_per_policy'].
 ↪fillna(0)
real_premium_1 = real_cluster_1['Premium_per_policy'].mean()

print(f"The real premium for cluster 1 is: {real_premium_1:.2f}$")
```

```
The real premium for cluster 1 is: 266.51$
```

Cluster 2

[661]:
```
# Cluster 2
real_cluster_2['Premium_per_policy'] = real_cluster_2['Frequency'] *␣
 ↪real_cluster_2['Severity']
real_cluster_2['Premium_per_policy'] = real_cluster_2['Premium_per_policy'].
 ↪fillna(0)
real_premium_2 = real_cluster_2['Premium_per_policy'].mean()

print(f"The real premium for cluster 2 is: {real_premium_2:.2f}$")
```

```
The real premium for cluster 2 is: 266.93$
```

Interpretation

It's noteworthy that the premiums are now relatively similar across all clusters, with a premium of $258.82 for Cluster 0 (the 'riskiest' cluster by our analysis), $266.51 for Cluster 1, and $266.93 for Cluster 2. These close values suggest that the clusters exhibit similar levels of risk based on premium pricing.

This convergence could be attributed to changes in the frequency and severity patterns for the `DrivAge` variable. Last year's data showed pronounced spikes in claim severity at ages 18 and 82. However, this trend has stabilized in the new data, with claim severity showing a more consistent pattern across different ages.

Interestingly, this observation relates to discussions we've had in class about the role of age in auto insurance pricing. As age-based variations in severity diminish, age may become less impactful in differentiating premiums among clusters. This stability in premiums across clusters might suggest that age, while traditionally a significant factor, could be losing its relevance as a primary risk determinant in our clustering model. Instead, age could serve more effectively as one of several factors contributing to overall risk profiles, rather than being a primary driver of premium differences.

Without sufficient differentiation in premiums, there's a risk of adverse selection: higher-risk individuals may receive premiums similar to lower-risk individuals, allowing them to benefit disproportionately, while lower-risk individuals bear part of the cost for riskier clients. To push this analysis further, it would be interesting to test if the use of credit-based insurance scores improves our pricing.

6.3 5.3 Loss ratio

6.3.1 Calculation of the loss ratio

[662]:
```
# Computing loss ratios for each cluster

total_severity_0 = real_cluster_0['Severity'].sum()
total_premiums_0 = real_cluster_0['Premium_per_policy'].sum()
loss_ratio_0 = total_severity_0 / total_premiums_0

total_severity_1 = real_cluster_1['Severity'].sum()
total_premiums_1 = real_cluster_1['Premium_per_policy'].sum()
loss_ratio_1 = total_severity_1 / total_premiums_1
```

```python
total_severity_2 = real_cluster_2['Severity'].sum()
total_premiums_2 = real_cluster_2['Premium_per_policy'].sum()
loss_ratio_2 = total_severity_2 / total_premiums_2


print(f"The loss ratio for cluster 0 is: {loss_ratio_0:.2f}")
print(f"The loss ratio for cluster 1 is: {loss_ratio_1:.2f}")
print(f"The loss ratio for cluster 2 is: {loss_ratio_2:.2f}")
```

```
The loss ratio for cluster 0 is: 0.23
The loss ratio for cluster 1 is: 0.32
The loss ratio for cluster 2 is: 0.32
```

Interpretation

With loss ratios being relatively low and stable across all three clusters—0.23 for Cluster 0, and 0.32 for Clusters 1 and 2—we can confidently say that our analysis is sound. These loss ratios indicate that our premium calculations are sufficient to cover claims costs across all clusters while allowing for a buffer that contributes to profitability. The similar loss ratios for Clusters 1 and 2 further validate the consistency of our risk categorization, while the slightly lower loss ratio for Cluster 0 suggests either lower-than-expected risk or conservative premium pricing for this group. Overall, this stability in loss ratios confirms the reliability of our model and supports our premium pricing strategy.

7 Problems we encountered

Throughout our analysis, we encountered several challenges that required careful consideration and additional research. First, fitting an appropriate distribution to our data proved difficult. We tested numerous distributions, including some outside the scope of our course, such as the Barr distribution, to find the best fit for our claims data. Another significant challenge was understanding how to structure our Monte Carlo simulation—whether to simulate separately for each cluster or to apply the simulation to the entire dataset. This decision had implications for both accuracy and computational efficiency, making it crucial to determine the best approach. Calculating the premium also presented obstacles; initially, our calculations yielded unreasonably high values, prompting us to conduct extensive research and refine our methodology. Finally, selecting a clustering method involved a trade-off between complexity and accuracy. We had to balance the desire for precise risk segmentation with the practicality of a simpler, more interpretable model.

8 Adjustments after in-class discussion

Following our in-class discussion, we adjusted our approach to calculating the adjusted premium that meets the 0.5% condition. Initially, we calculated the percentile value (VaR) individually for each cluster, assuming that this would yield an appropriate adjusted premium. However, during the discussion, we learned that Value at Risk (VaR) is not additive, meaning that calculating VaR separately for each cluster does not correctly represent the combined risk. To address this, we modified our method by first determining the adjusted premium for total losses across all clusters, then allocating it proportionally based on each cluster's share of total losses.

9 How we divided the work

Throughout this homework, we collaborated closely, with each person taking the lead on specific sections. Thomas led sections 1 and 2, Mariève led sections 4 and 5, and we jointly led section 3, fully sharing responsibility for its development. Although each of us held primary responsibility for certain sections, we consistently worked together on each part. This structure allowed us to leverage individual strengths while continuously exchanging ideas to find effective solutions.

10 Appendix

10.0.1 Appendix 1: Extra Descriptive Analysis

```
[663]: total_claims_frequency = pd.DataFrame(claims_data['Frequency'].value_counts())
       total_claims_frequency
```

```
[663]:             count
       Frequency
       0.000000    96335
       1.000000     1247
       2.000000      146
       2.040816       59
       4.166667       53
       …             …
       22.222222       1
       2.597403        1
       121.666668      1
       28.571429       1
       122.000005      1

       [149 rows x 1 columns]
```

```
[664]: total_claims_severity = pd.DataFrame(claims_data['Severity'].value_counts())
       total_claims_severity
```

```
[664]:           count
       Severity
       0.00      96335
       1204.00     662
       1128.12     409
       1172.00     290
       1128.00      96
       …           …
       1307.64       1
       1858.81       1
       741.77        1
       4285.99       1
       1117.64       1
```

```
[1937 rows x 1 columns]
```

[665]:
```
# Total claims by driver's age
total_claims_by_age_frequency = claims_data.groupby('DrivAge')['Frequency'].
 ↪sum()
total_claims_by_age_severity = claims_data.groupby('DrivAge')['Severity'].sum()
total_claims_by_age_frequency = pd.DataFrame(total_claims_by_age_frequency)
total_claims_by_age_severity = pd.DataFrame(total_claims_by_age_severity)
pd.concat([total_claims_by_age_frequency, total_claims_by_age_severity],␣
 ↪axis=1).head(3)
```

[665]:
```
          Frequency    Severity
DrivAge
18        16.322587  210524.965
19        91.403724   78742.210
20       213.609450   90009.665
```

[666]:
```
# Total claims by vehicle's age
total_claims_by_vech_age_frequency = claims_data.groupby('VehAge')['Frequency'].
 ↪sum()
total_claims_by_vech_age_severity = claims_data.groupby('VehAge')['Severity'].
 ↪sum()
total_claims_by_vech_age_frequency = pd.
 ↪DataFrame(total_claims_by_vech_age_frequency)
total_claims_by_vech_age_severity = pd.
 ↪DataFrame(total_claims_by_vech_age_severity)
pd.concat([total_claims_by_vech_age_frequency,␣
 ↪total_claims_by_vech_age_severity], axis=1).head(3)
```

[666]:
```
          Frequency       Severity
VehAge
0        570.358022  362637.860000
1       1244.758156  512712.610000
2        896.493005  692091.993333
```

[667]:
```
# Total claims by Bonus/Malus
total_claims_by_bonus_malus_frequency = claims_data.
 ↪groupby('BonusMalus')['Frequency'].sum()
total_claims_by_bonus_malus_severity = claims_data.
 ↪groupby('BonusMalus')['Severity'].sum()
total_claims_by_vech_age_frequency = pd.
 ↪DataFrame(total_claims_by_bonus_malus_frequency)
total_claims_by_bonus_malus_severity = pd.
 ↪DataFrame(total_claims_by_bonus_malus_severity)
pd.concat([total_claims_by_bonus_malus_frequency,␣
 ↪total_claims_by_bonus_malus_severity], axis=1).head(3)
```

[667]:
```
              Frequency      Severity
BonusMalus
50          4527.854394  3.367297e+06
51           144.747215  5.606332e+04
52            66.867186  4.866783e+04
```

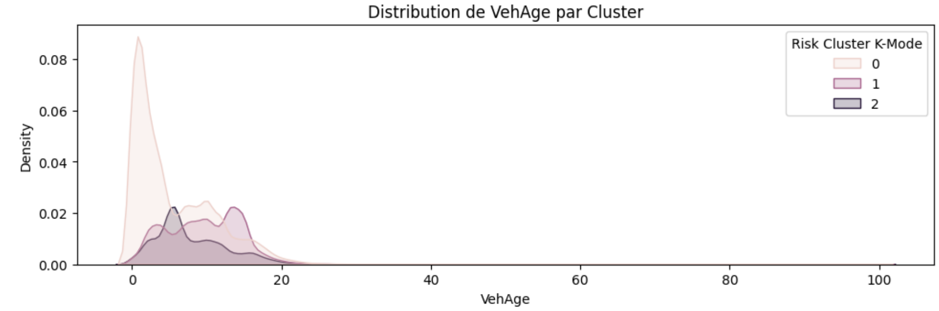
[668]:
```
# Total claims by vechicle's power
total_claims_by_vech_power_frequency = claims_data.
 ↪groupby('VehPower')['Frequency'].sum()
total_claims_by_vech_power_severity = claims_data.
 ↪groupby('VehPower')['Severity'].sum()
total_claims_by_vech_power_frequency = pd.
 ↪DataFrame(total_claims_by_vech_power_frequency)
total_claims_by_vech_power_severity = pd.
 ↪DataFrame(total_claims_by_vech_power_severity)
pd.concat([total_claims_by_vech_power_frequency,␣
 ↪total_claims_by_vech_power_severity], axis=1).head(3)
```

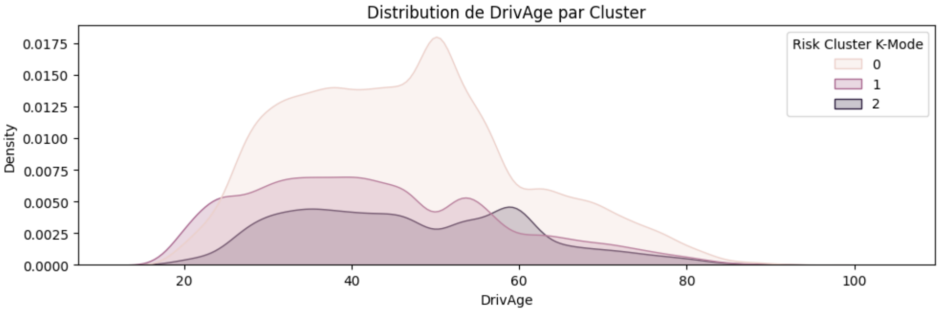
[668]:
```
            Frequency      Severity
VehPower
4         2241.808393  8.955589e+05
5         2248.298384  1.481127e+06
6         2487.080157  1.690078e+06
```

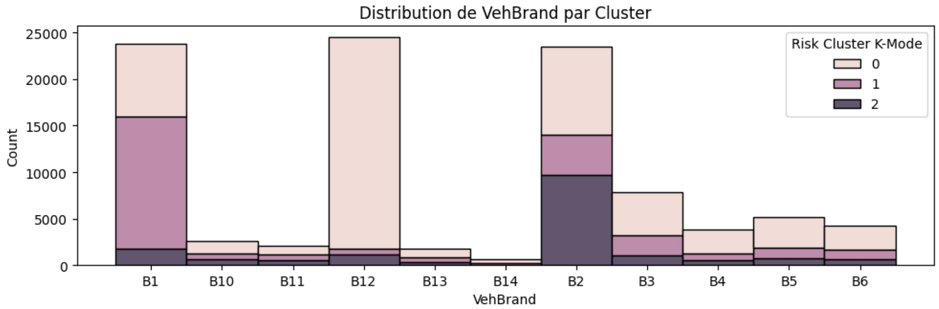
10.0.2 Appendix 2: Distribution of Fisk Variables Per Cluster (histograms for categorical variables and CDF for numerical variables)

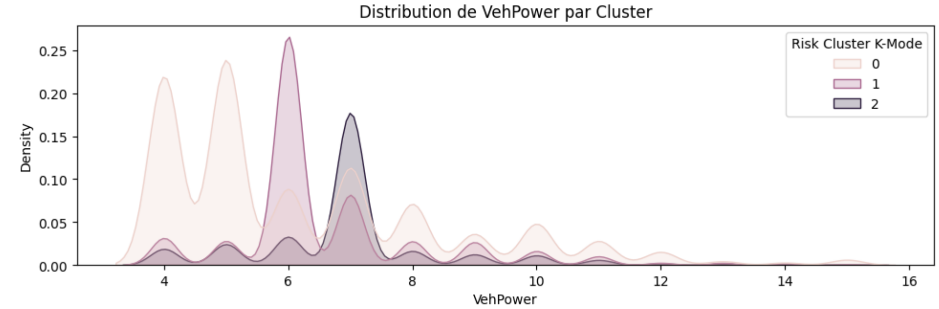
[669]:
```
# Define the variables to plot
variables = ['VehAge', 'DrivAge', 'VehBrand', 'VehPower', 'Area', 'Region']
fig, axes = plt.subplots(len(variables), 1, figsize=(10, 20))

# Plot the distribution of variables by predicted risk cluster
# This allows us to visualize how different variables are distributed across␣
 ↪the predicted risk clusters.
for i, var in enumerate(variables):
    assert var in claims_data.columns, f"La colonne {var} est manquante dans␣
 ↪claims_data"
    if claims_data[var].dtype.name == 'category' or claims_data[var].dtype.name␣
 ↪== 'object':
        sns.histplot(data=claims_data, x=var, hue='Risk Cluster K-Mode',␣
 ↪multiple="stack", ax=axes[i])
    else:
        sns.kdeplot(data=claims_data, x=var, hue='Risk Cluster K-Mode',␣
 ↪fill=True, ax=axes[i])
    axes[i].set_title(f'Distribution de {var} par Cluster')

# Adjust the layout of the subplots to avoid overlap
```

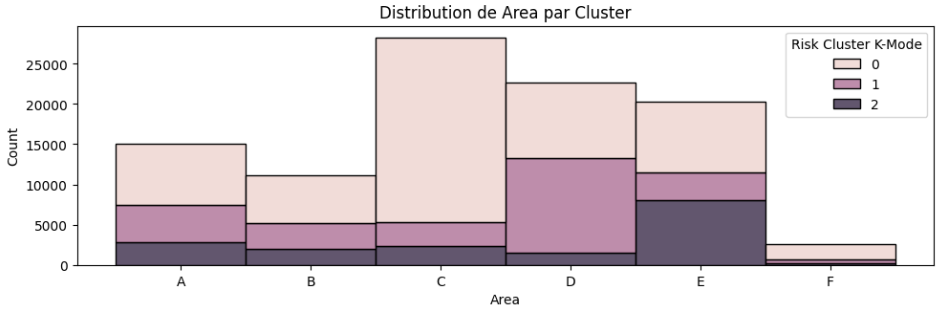
```
plt.tight_layout()
plt.show()
```

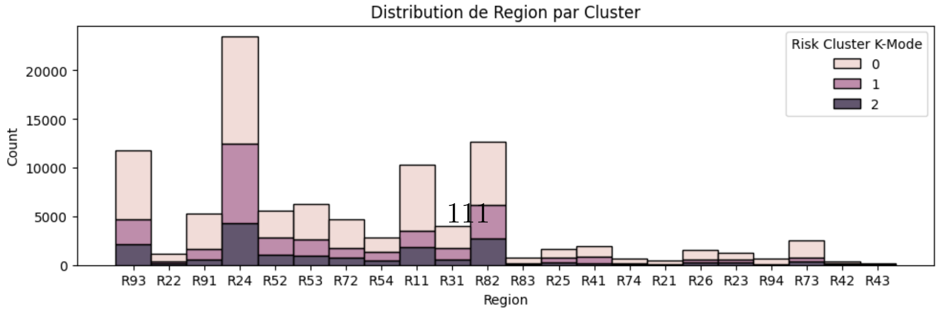

Distribution de VehAge par Cluster
Risk Cluster K-Mode
0
1
2
Density
0.00
0.02
0.04
0.06
0.08
0
20
40
60
80
100
VehAge


Distribution de DrivAge par Cluster
Risk Cluster K-Mode
0
1
2
Density
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
20
40
60
80
100
DrivAge


Distribution de VehBrand par Cluster
Risk Cluster K-Mode
0
1
2
Count
0
5000
10000
15000
20000
25000
B1
B10
B11
B12
B13
B14
B2
B3
B4
B5
B6
VehBrand

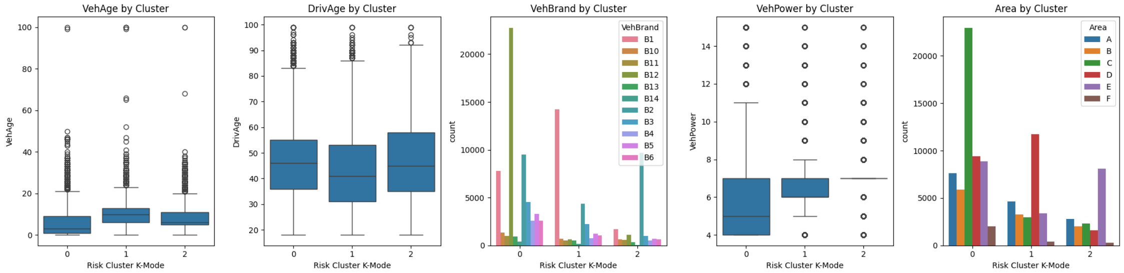

Distribution de VehPower par Cluster
Risk Cluster K-Mode
0
1
2
Density
0.00
0.05
0.10
0.15
0.20
0.25
4
6
8
10
12
14
16
VehPower


Distribution de Area par Cluster
Risk Cluster K-Mode
0
1
2
Count
0
5000
10000
15000
20000
25000
A
B
C
D
E
F
Area


Distribution de Region par Cluster
Risk Cluster K-Mode
0
1
2
Count
0
5000
10000
15000
20000
R93
R22
R91
R24
R52
R53
R72
R54
R11
R31
R82
R83
R25
R41
R74
R21
R26
R23
R94
R73
R42
R43
Region

[670]:

```
# Define the variables to plot
variables = ['VehAge', 'DrivAge', 'VehBrand', 'VehPower', 'Area']
fig, axes = plt.subplots(1, len(variables), figsize=(20, 5))

# Plot the distribution of variables by predicted risk cluster
for i, var in enumerate(variables):
    assert var in claims_data.columns, f"The column {var} is missing in␣
 ↪claims_data"
    if claims_data[var].dtype.name == 'category' or claims_data[var].dtype.name␣
 ↪== 'object':
        sns.countplot(data=claims_data, x='Risk Cluster K-Mode', hue=var,␣
 ↪ax=axes[i])
    else:
        sns.boxplot(data=claims_data, x='Risk Cluster K-Mode', y=var,␣
 ↪ax=axes[i])
    axes[i].set_title(f'{var} by Cluster')

# Adjust the layout of the subplots to avoid overlap
plt.tight_layout()
plt.show()
```

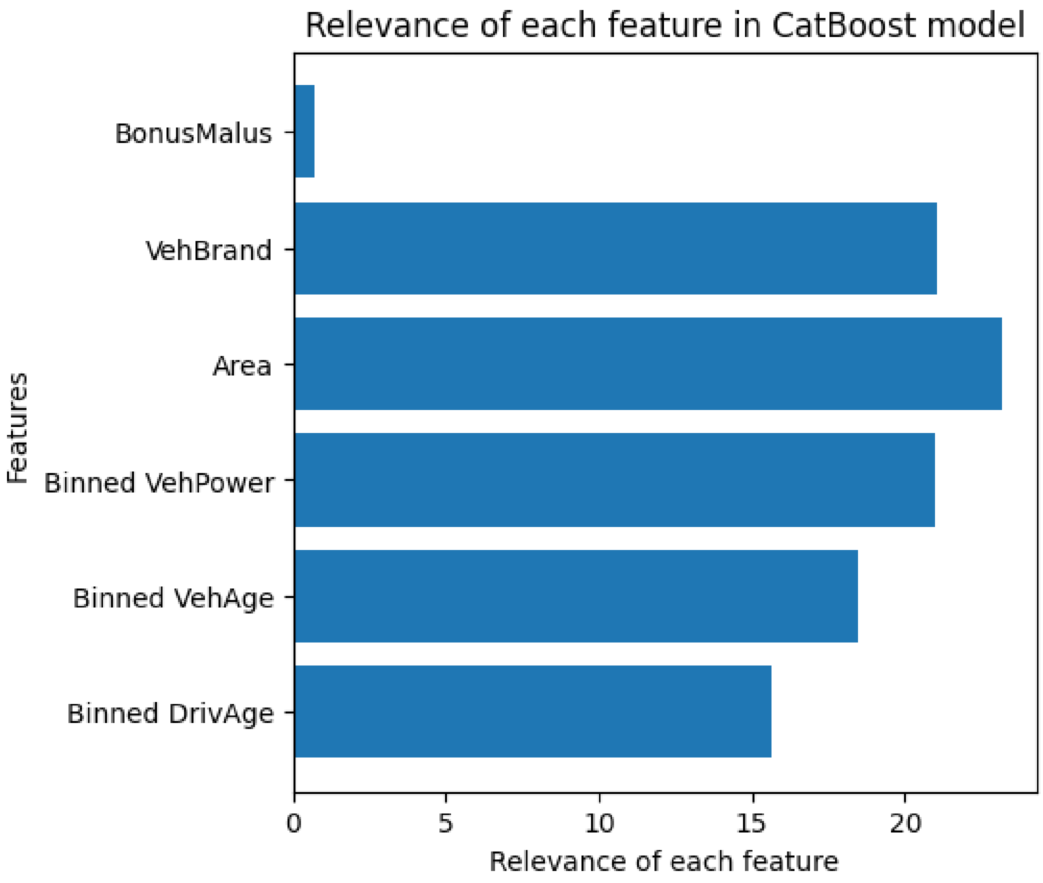


10.0.3 Appendix 3: Relevance of Bonus / Malus in CatBoost

[671]:

```
# CatBoost provides built-in methods to extract feature importance, which can␣
 ↪offer insights into what features are driving the distinctions between␣
 ↪clusters.
# Obtain which characteristics are the most important for predicting the␣
 ↪cluster labels.
feature_importances = model.get_feature_importance()
feature_names = df_categorical.columns

# Create a bar plot to visualize the importance of each characteristics in the␣
 ↪CatBoost model
```

```python
plt.figure(figsize=(5, 5))
plt.barh(feature_names, feature_importances)
plt.xlabel('Relevance of each feature')
plt.ylabel('Features')
plt.title('Relevance of each feature in CatBoost model')
plt.show()
```


Relevance of each feature in CatBoost model
Features
BonusMalus
VehBrand
Area
Binned VehPower
Binned VehAge
Binned DrivAge
Relevance of each feature