13 Supervised learning with decision trees

Nandi Christmas

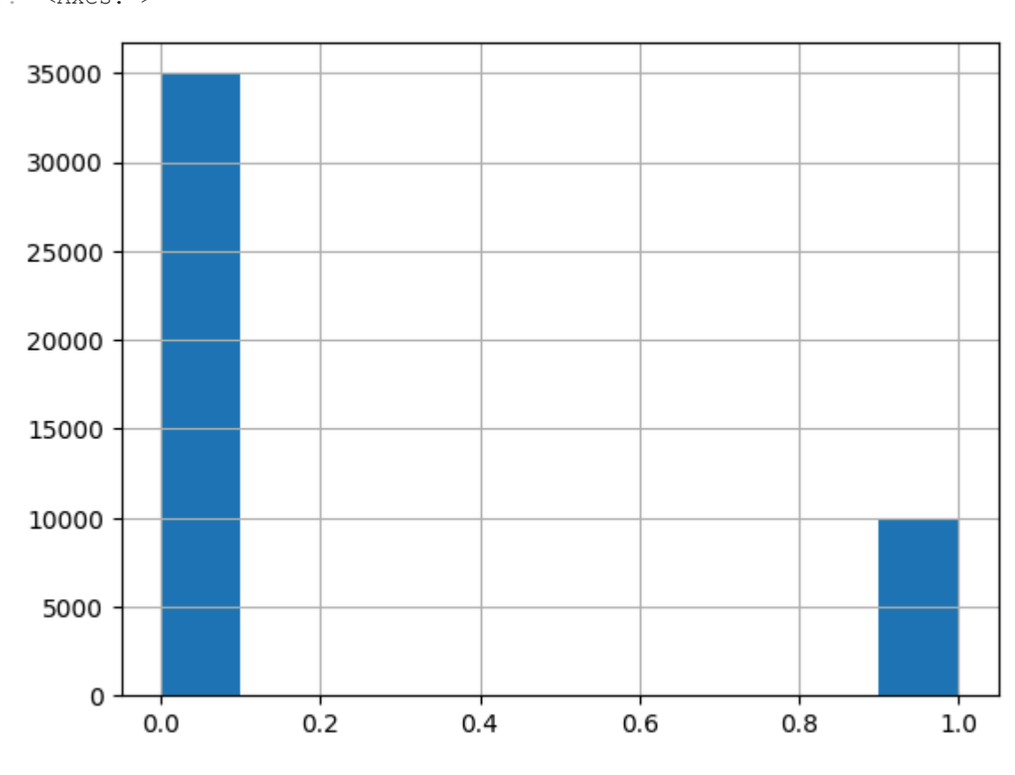
In [9]:
```python
import numpy as np
import pandas as pd
df_loan = pd.read_csv(
    filepath_or_buffer = 'D:\\loan_data.csv',
    engine = 'pyarrow'
)
df_loan.head()
```

Out[9]:

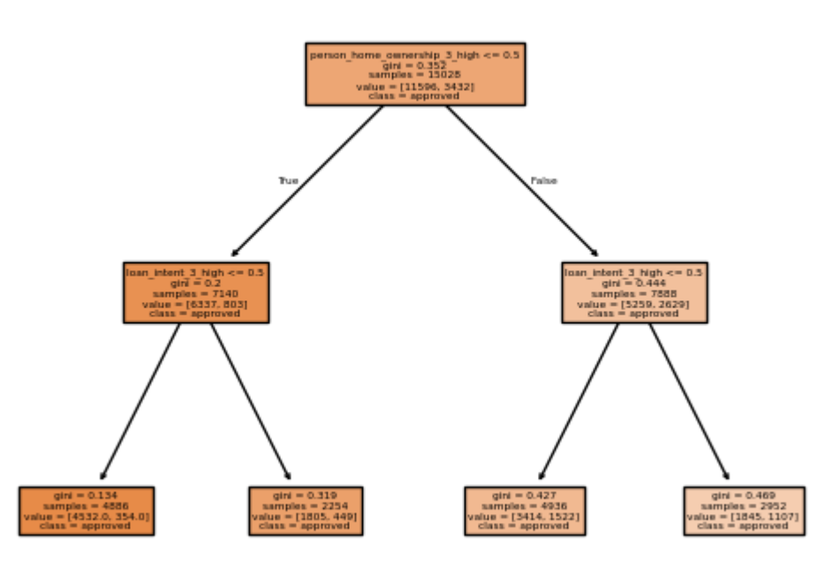
	loan_status	partition	person_education_2_moderate	person_education_3_high	person_home_ownership_3_high	loan_intent_2_moderate	loan_intent_3_high
0	1	test	0	0	1	1	0
1	0	test	1	0	0	0	0
2	1	train	1	0	0	0	1
3	1	train	0	1	1	0	1
4	1	test	0	0	1	0	1

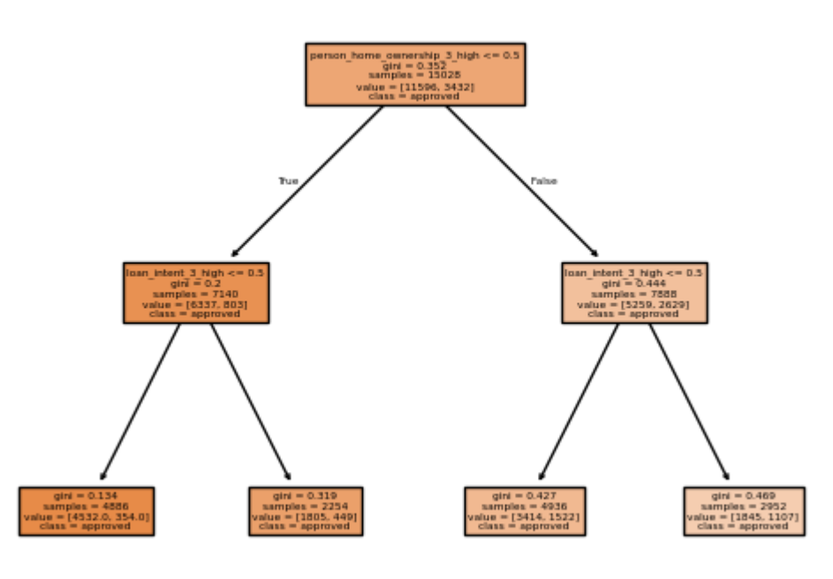
In [10]:
```python
list_predictors = [
      'person_education_2_moderate',
        'person_education_3_high',
        'person_home_ownership_3_high',
        'loan_intent_2_moderate',
        'loan_intent_3_high',
]
X = df_loan[list_predictors]
y = df_loan['loan_status']
X_Train = df_loan.loc[df_loan['partition'] == 'train'][list_predictors]
y_Train = df_loan.loc[df_loan['partition'] == 'train','loan_status']
df_loan['loan_status'].hist()
```

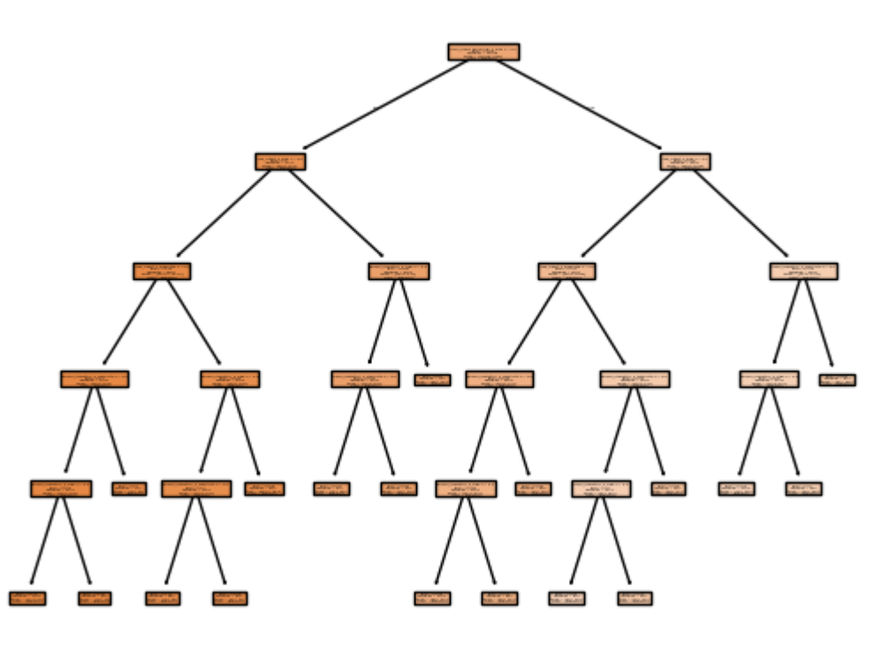
Out[10]: <Axes: >

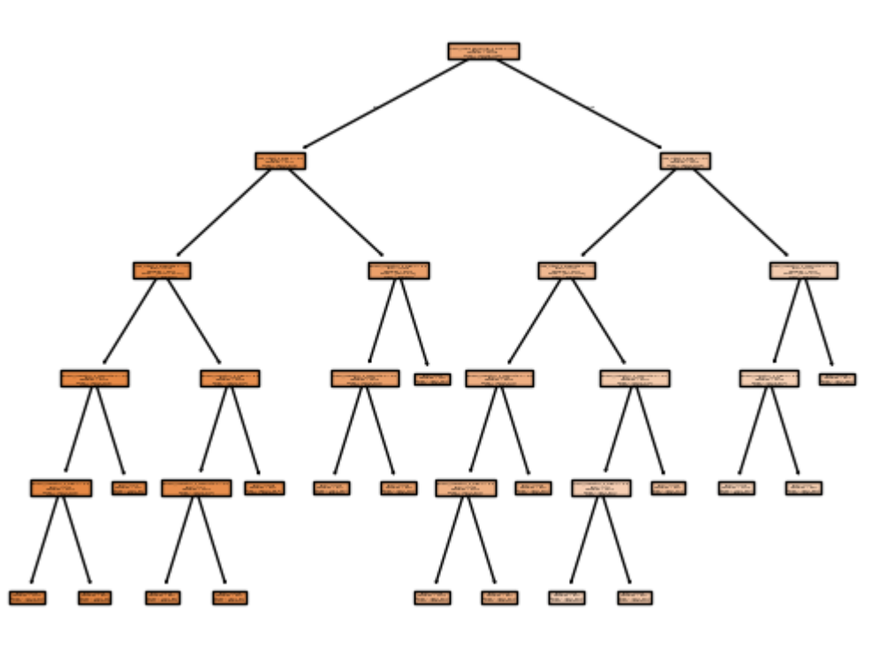


In [11]:
```python
list_max_depth = [2,None]
list_min_samples_leaf = [300,1]
from sklearn.tree import DecisionTreeClassifier
list_DecisionTreeClassifier = [DecisionTreeClassifier(max_depth = j,min_samples_leaf = k) for j in list_max_depth for k in list_min_samples_leaf]
list_DecisionTreeClassifier_fit = [model.fit(
    X = X_Train,
    y = y_Train
) for model in list_DecisionTreeClassifier]
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
for fit in list_DecisionTreeClassifier_fit:
    plot_tree(
        decision_tree = fit,
        feature_names = list_predictors,
        class_names = ['approved','denied'],
        filled = True
    )
    plt.show()
```

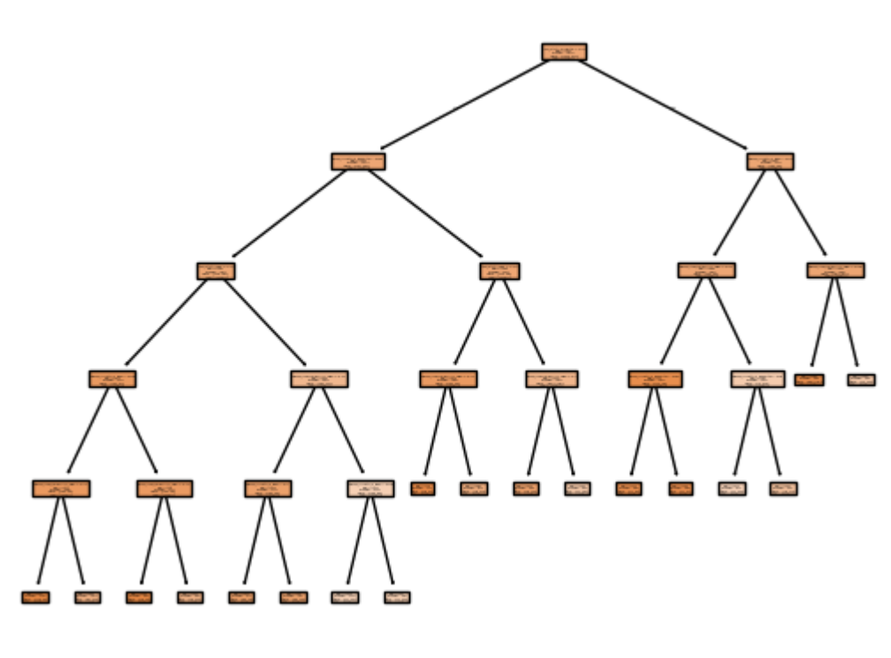


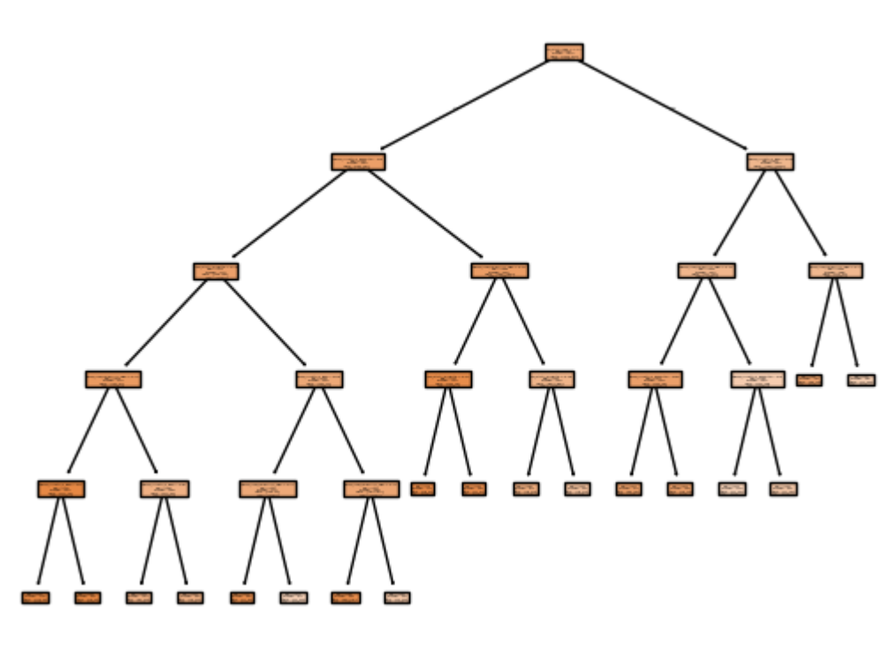


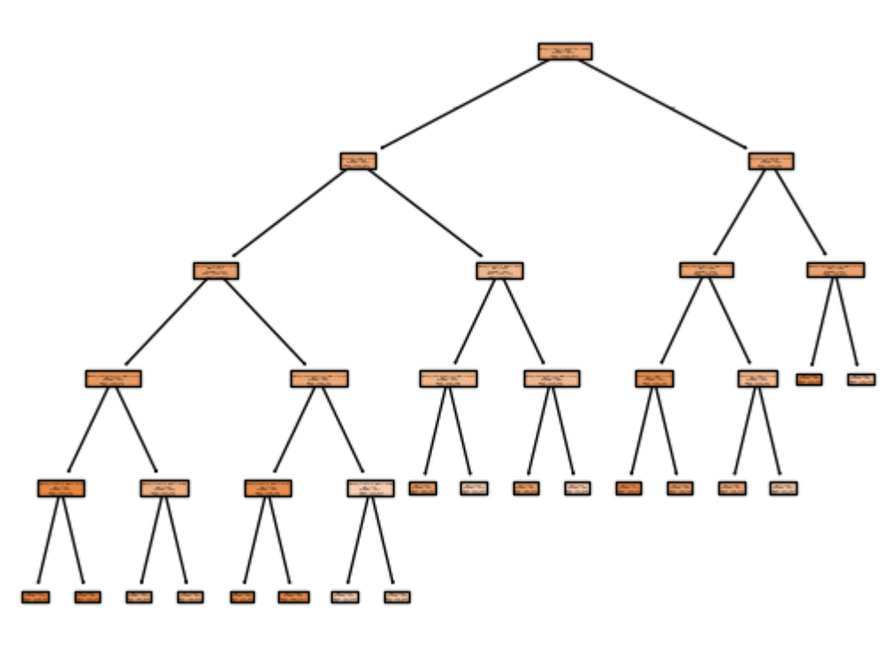


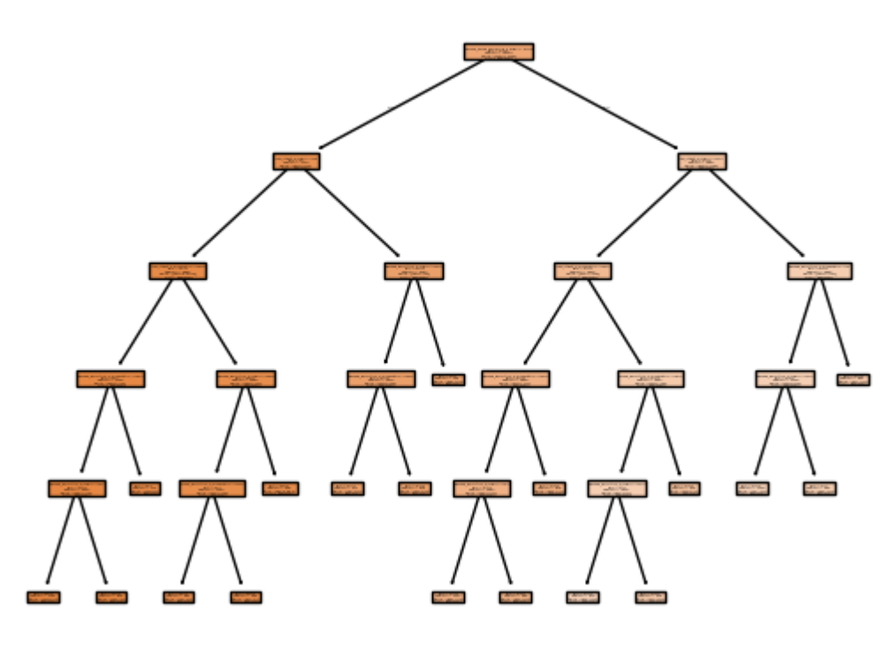


In [12]:
```python
list_max_features = [1,'sqrt','log2',None]
from sklearn.tree import ExtraTreeClassifier
list_ExtraTreeClassifier = [ExtraTreeClassifier(max_features = j) for j in list_max_features]
list_ExtraTreeClassifier_fit = [model.fit(
    X = X_Train,
    y = y_Train
) for model in list_ExtraTreeClassifier]
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
for fit in list_ExtraTreeClassifier_fit:
    plot_tree(
        decision_tree = fit,
        feature_names = list_predictors,
        class_names = ['approved','denied'],
        filled = True
    )
    plt.show()
```









In [13]:
```python
list_fit = list_DecisionTreeClassifier_fit + list_ExtraTreeClassifier_fit
```

In [15]:
```python
list_predict = [fit.predict(X = X[list_predictors]) for fit in list_fit]
list_predict_proba = [fit.predict_proba(X = X[list_predictors])[:,1] for fit in list_fit]
df_predict_proba = pd.DataFrame(list_predict_proba).T
df_predict_proba.head()
```

Out[15]:

	0	1	2	3	4	5	6	7
0	0.308347	0.308347	0.389655	0.389655	0.389655	0.389655	0.389655	0.389655
1	0.072452	0.072452	0.069948	0.069948	0.069948	0.069948	0.069948	0.069948
2	0.199201	0.199201	0.198905	0.198905	0.198905	0.198905	0.198905	0.198905
3	0.375000	0.375000	0.377895	0.377895	0.377895	0.377895	0.377895	0.377895
4	0.375000	0.375000	0.384808	0.384808	0.384808	0.384808	0.384808	0.384808

In [16]:
```python
from sklearn.metrics import roc_auc_score,roc_curve
df_loan['partition']
['train','validation','test']
list_roc_auc_score = [[j,k,roc_auc_score(
    y_true = df_loan.loc[df_loan['partition'] == k,'loan_status'],
    y_score = df_predict_proba.loc[df_loan['partition'] == k,j]
)] for j in range(df_predict_proba.shape[1]) for k in ['train','validation','test']]
df_roc_auc_score = pd.DataFrame(list_roc_auc_score)
df_roc_auc_score.columns = ['model','partition','roc_auc_score']
df_roc_auc_score = df_roc_auc_score.pivot(
    index='model',
    columns='partition',
    values='roc_auc_score'
)
df_roc_auc_score['TreeClassifier'] = list_DecisionTreeClassifier + list_ExtraTreeClassifier
df_roc_auc_score.sort_values(['test','validation','train'])
```

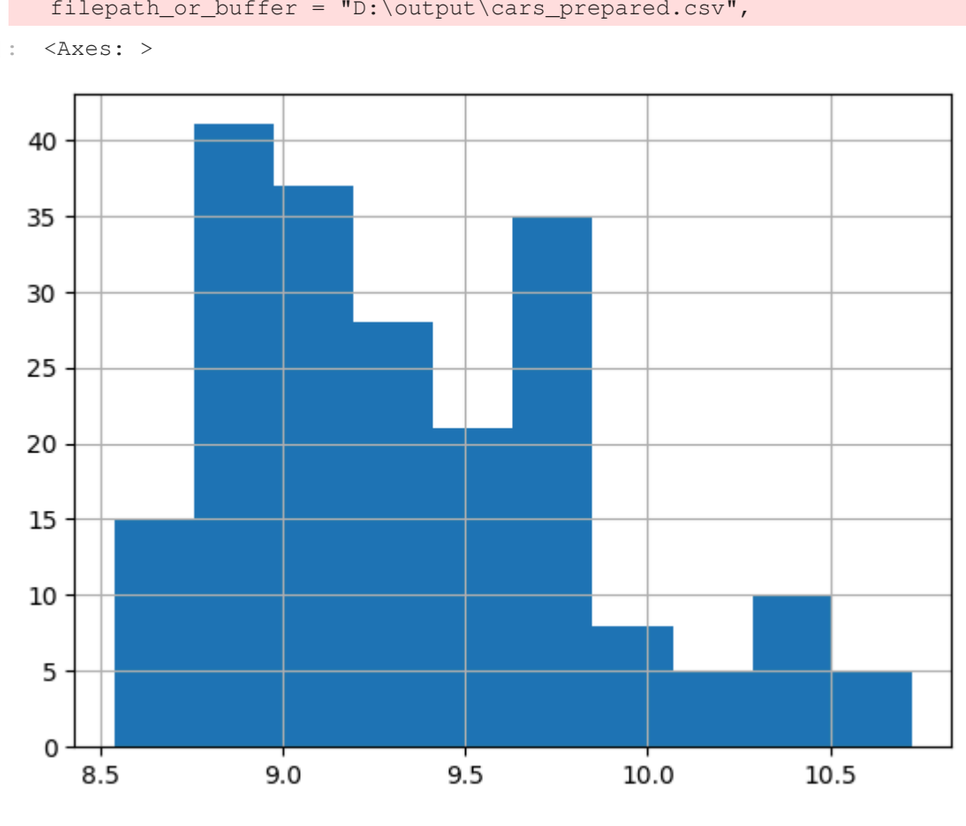
Out[16]:

partition	test	train	validation	TreeClassifier
model				
0	0.681489	0.685993	0.679326	DecisionTreeClassifier(max_depth=2, min_sample...
1	0.681489	0.685993	0.679326	DecisionTreeClassifier(max_depth=2)
2	0.688507	0.699059	0.692372	DecisionTreeClassifier(min_samples_leaf=300)
3	0.688507	0.699059	0.692372	DecisionTreeClassifier()
4	0.688507	0.699059	0.692372	ExtraTreeClassifier(max_features=1)
5	0.688507	0.699059	0.692372	ExtraTreeClassifier()
6	0.688507	0.699059	0.692372	ExtraTreeClassifier(max_features='log2')
7	0.688507	0.699059	0.692372	ExtraTreeClassifier(max_features=None)

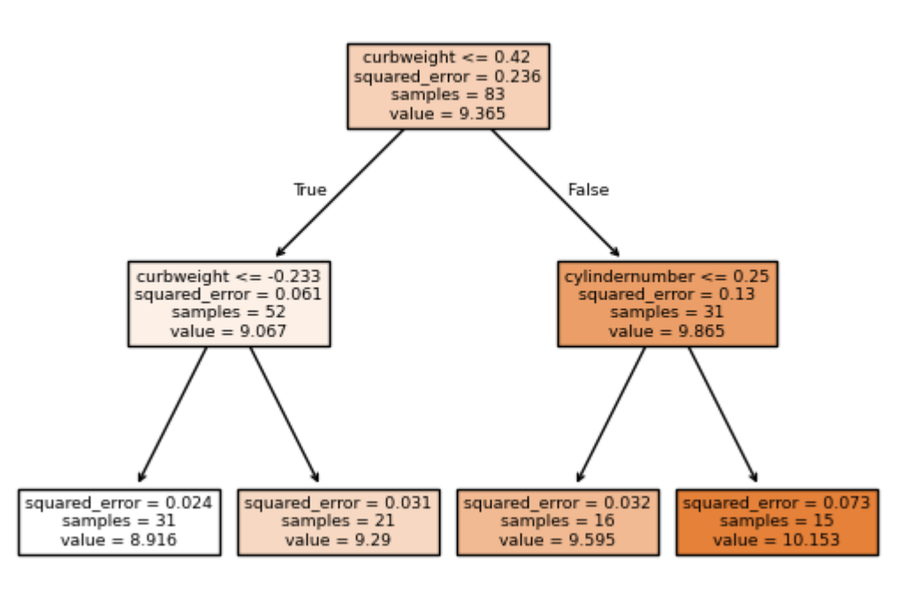
In [17]:
```python
import pandas as pd
import numpy as np
df = pd.read_csv(
    filepath_or_buffer = "D:\output\cars_prepared.csv",
    engine = 'pyarrow',
)
list_reduced = [
    'symboling',
    'doornumber',
    'wheelbase',
    'carlength',
    'carwidth',
    'carheight',
    'curbweight',
    'cylindernumber',
    'enginesize',
    'boreratio',
    'stroke',
    'compressionratio',
    'horsepower',
    'peakrpm',
    'citympg',
    'highwaympg',
    'CarName_1_low',
    'drivewheel_1_low',
    'enginelocation_3_high',
    'aspiration_3_high',
    'fuelsystem_3_high',
    'enginetype_2_moderate',
    'fueltype_2_moderate',
    'carbody_1_low'
]
X = df[list_reduced]
y = df['price']
X_Train = df[list_reduced].loc[df['TrainTest'] == 'Train']
y_Train = df['price'].loc[df['TrainTest'] == 'Train']
df['price'].hist()
```

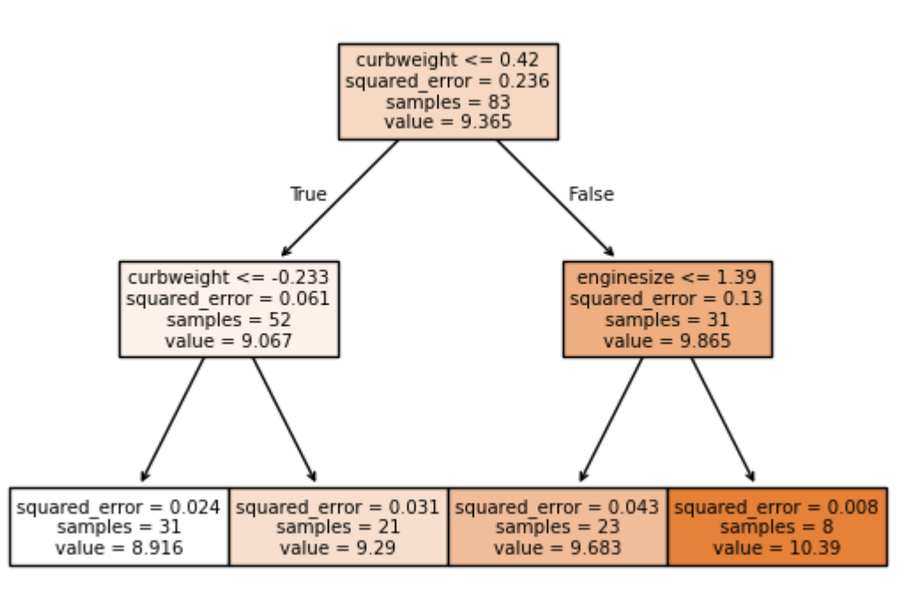
```
<>:4: SyntaxWarning: invalid escape sequence '\o'
<>:4: SyntaxWarning: invalid escape sequence '\o'
C:\Users\nandi\AppData\Local\Temp\ipykernel_25980\2286807562.py:4: SyntaxWarning: invalid escape sequence '\o'
  filepath_or_buffer = "D:\output\cars_prepared.csv",
```

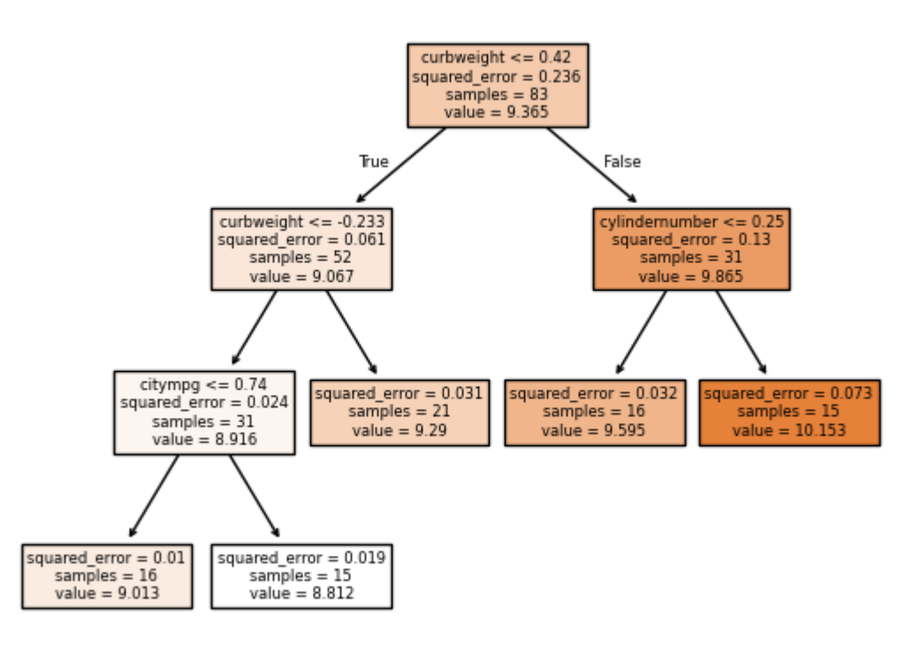
Out[17]: <Axes: >

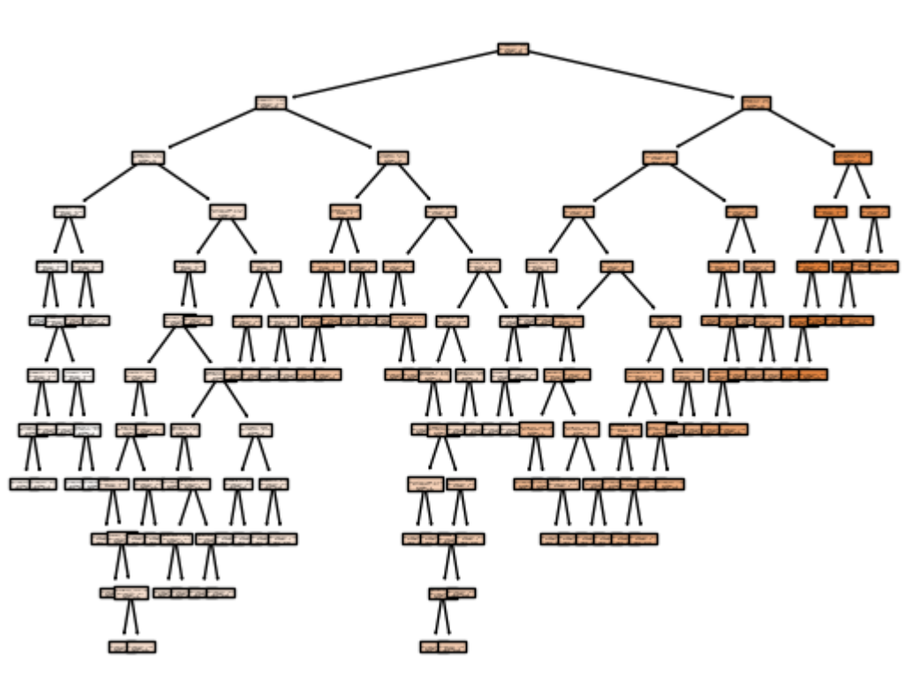


In [18]:
```python
list_max_depth = [2,None]
list_min_samples_leaf = [15,1]
from sklearn.tree import DecisionTreeRegressor
list_DecisionTreeRegressor = [DecisionTreeRegressor(max_depth = j,min_samples_leaf = k) for j in list_max_depth for k in list_min_samples_leaf]
list_DecisionTreeRegressor
list_DecisionTreeRegressor_fit = [model.fit(
    X = X_Train,
    y = y_Train
) for model in list_DecisionTreeRegressor]
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
for fit in list_DecisionTreeRegressor_fit:
    plot_tree(
        decision_tree = fit,
        feature_names = list_reduced,
        filled = True
    )
    plt.show()
```

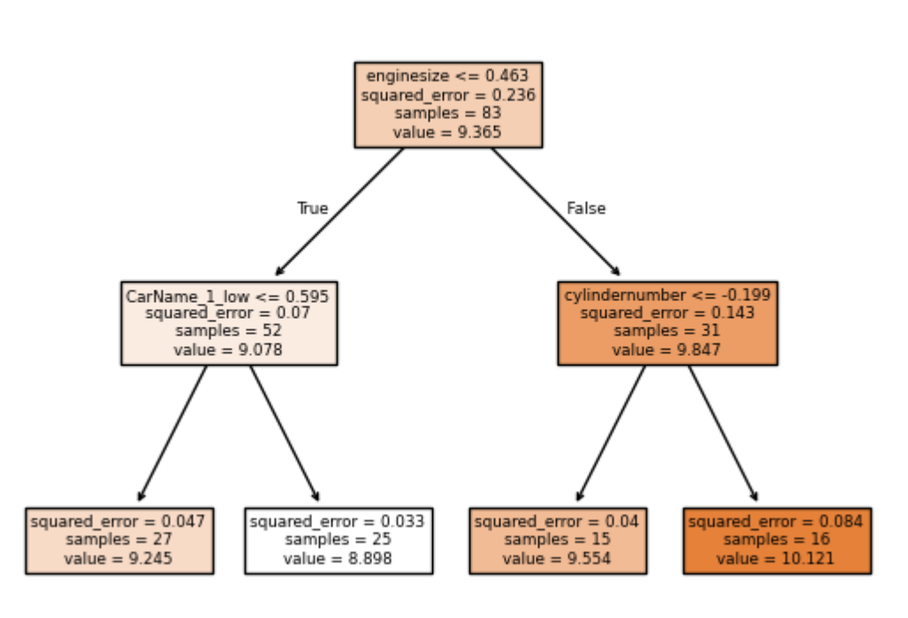


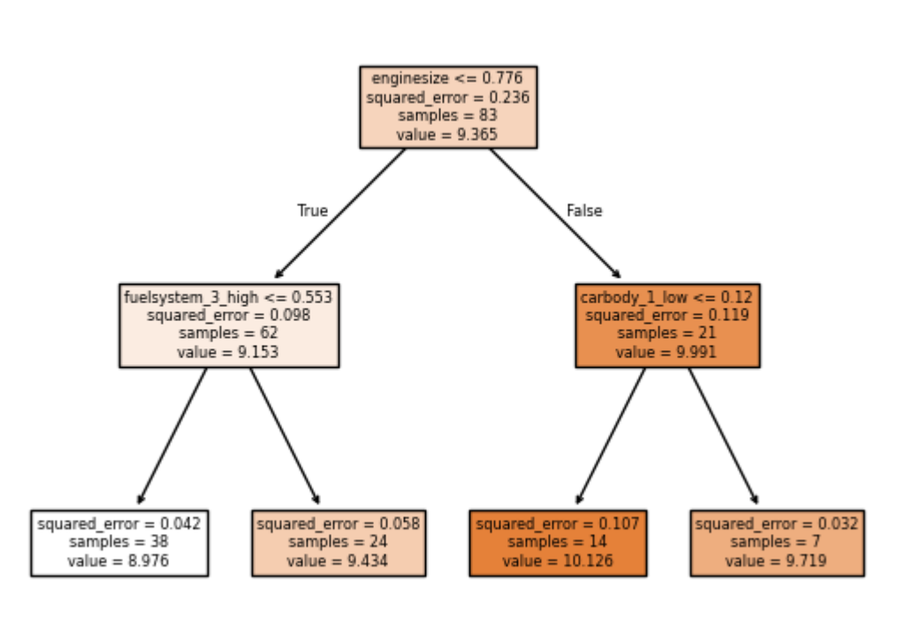


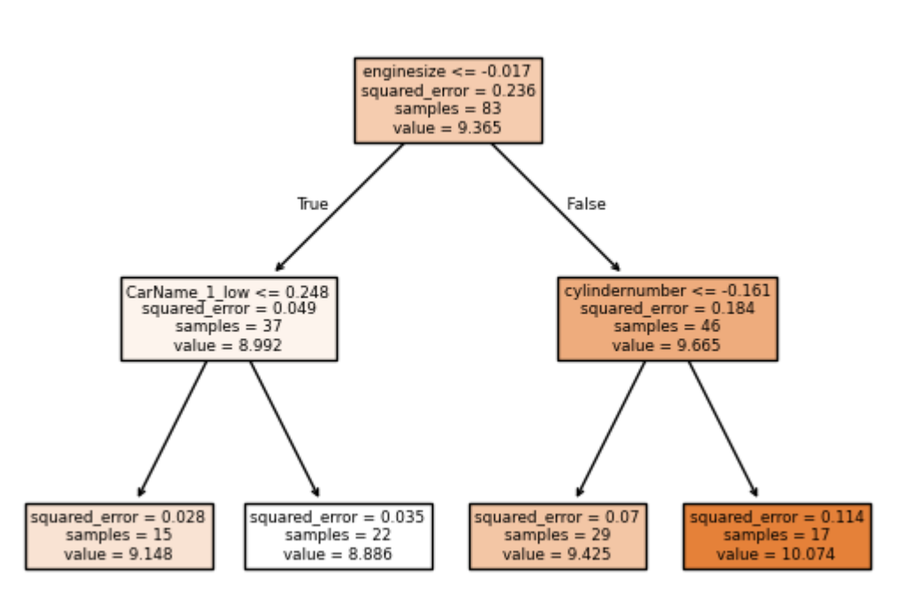


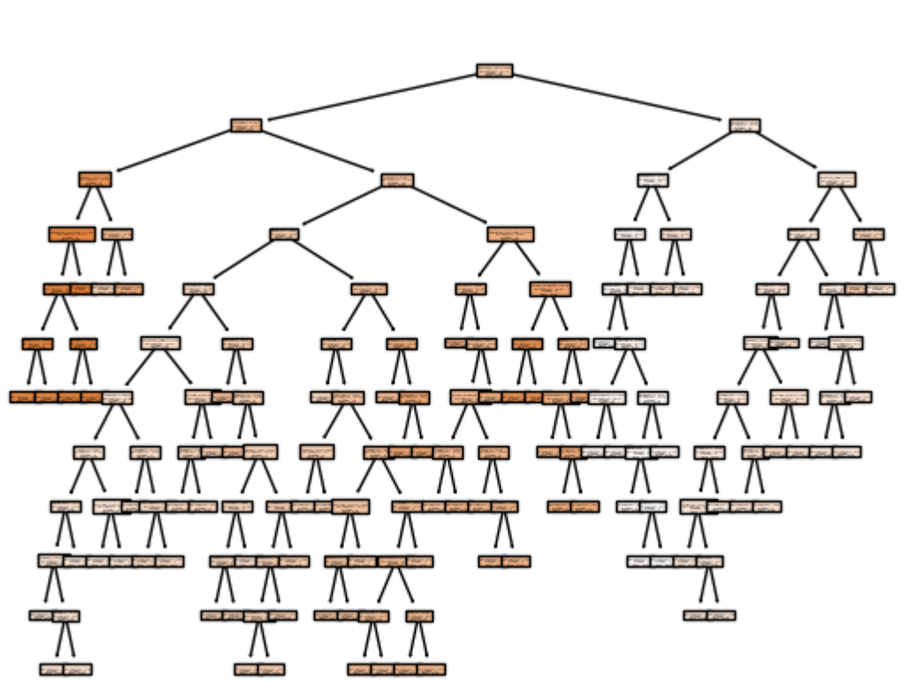


In [19]:
```python
list_max_depth = [2,None]
list_min_samples_leaf = [15,1]
from sklearn.tree import ExtraTreeRegressor
list_ExtraTreeRegressor = [ExtraTreeRegressor(max_depth = j,min_samples_leaf = k) for j in list_max_depth for k in list_min_samples_leaf]
list_ExtraTreeRegressor
list_ExtraTreeRegressor_fit = [model.fit(
    X = X_Train,
    y = y_Train
) for model in list_ExtraTreeRegressor]
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
for fit in list_ExtraTreeRegressor_fit:
    plot_tree(
        decision_tree = fit,
        feature_names = list_reduced,
        filled = True
    )
    plt.show()
```









In [20]:
```python
list_fit = list_DecisionTreeRegressor_fit + list_ExtraTreeRegressor_fit
```

In [21]:
```python
list_predict = [fit.predict(X = X[list_reduced]) for fit in list_fit]
df_predict = pd.DataFrame(list_predict).T
df_predict.head()
```

Out[21]:

	0	1	2	3	4	5	6	7
0	9.290163	9.290163	9.290163	9.510075	9.244562	9.433748	9.425308	9.510075
1	9.290163	9.290163	9.290163	9.510075	9.244562	9.433748	9.425308	9.510075
2	10.153301	9.682777	10.153301	9.999616	10.120741	9.718682	10.074106	9.632335
3	8.915628	8.915628	9.012965	9.206332	9.244562	9.433748	9.147842	9.296335
4	10.153301	9.682777	10.153301	9.999616	10.120741	9.433748	10.074106	9.632335

In [22]:
```python
from sklearn.metrics import mean_squared_error



list_mean_squared_error = [[j,k,mean_squared_error(
    y_true = df['price'].loc[df['TrainTest'] == k],
    y_pred = df_predict.loc[df['TrainTest'] == k,j]
)] for j in range(df_predict.shape[1]) for k in ['Train','Validation','Test']]
df_mean_squared_error = pd.DataFrame(list_mean_squared_error)
df_mean_squared_error.columns = ['model','TrainTest','mean_squared_error']
df_mean_squared_error = df_mean_squared_error.pivot(
    index='model',
    columns='TrainTest',
    values='mean_squared_error'
)
df_mean_squared_error['TreeRegressor'] = list_DecisionTreeRegressor + list_ExtraTreeRegressor
df_mean_squared_error.sort_values(['Test','Validation','Train'])[['TreeRegressor','Test','Validation','Train']]
```

Out[22]:

TrainTest	TreeRegressor	Test	Validation	Train
model				
1	DecisionTreeRegressor(max_depth=2)	0.052570	0.034449	0.029554
3	DecisionTreeRegressor()	0.053117	0.039021	0.000012
7	ExtraTreeRegressor()	0.082040	0.032838	0.000012
2	DecisionTreeRegressor(min_samples_leaf=15)	0.115600	0.055697	0.032515
5	ExtraTreeRegressor(max_depth=2)	0.116271	0.085055	0.056679
0	DecisionTreeRegressor(max_depth=2, min_samples...	0.122290	0.061539	0.036290

4	ExtraTreeRegressor(max_depth=7, min_samples_s...	0.127234	0.051499	0.226100
5	ExtraTreeRegressor(min_samples_leaf=10)	0.119238	0.072670	0.269540