In [ ]:
```
import pandas as pd
```

In [ ]:
```
df = pd.read_csv("employee_churn_data_clean.csv")
df
```

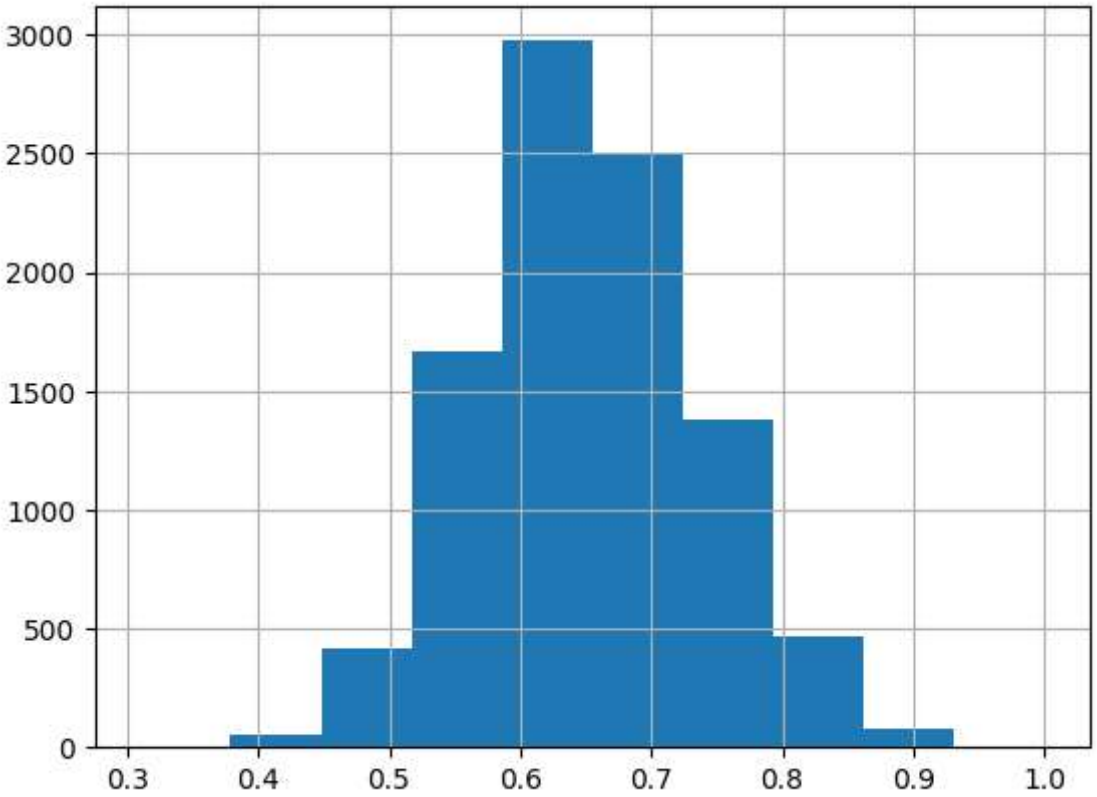
Out[ ]:

	department	promoted	review	projects	salary	tenure	satisfaction	bonus	avg_
0	operations	0	0.577569	3	0	5.0	0.626759	0	
1	operations	0	0.751900	3	1	6.0	0.443679	0	
2	support	0	0.722548	3	1	6.0	0.446823	0	
3	logistics	0	0.675158	4	2	8.0	0.440139	0	
4	sales	0	0.676203	3	2	5.0	0.577607	1	
...	...	...	...	...	...	...	...	...	
9535	operations	0	0.610988	4	1	8.0	0.543641	0	
9536	logistics	0	0.746887	3	1	8.0	0.549048	0	
9537	operations	0	0.557980	3	0	7.0	0.705425	0	
9538	IT	0	0.584446	4	1	8.0	0.607287	1	
9539	finance	0	0.626373	3	0	7.0	0.706455	1	

9540 rows × 10 columns

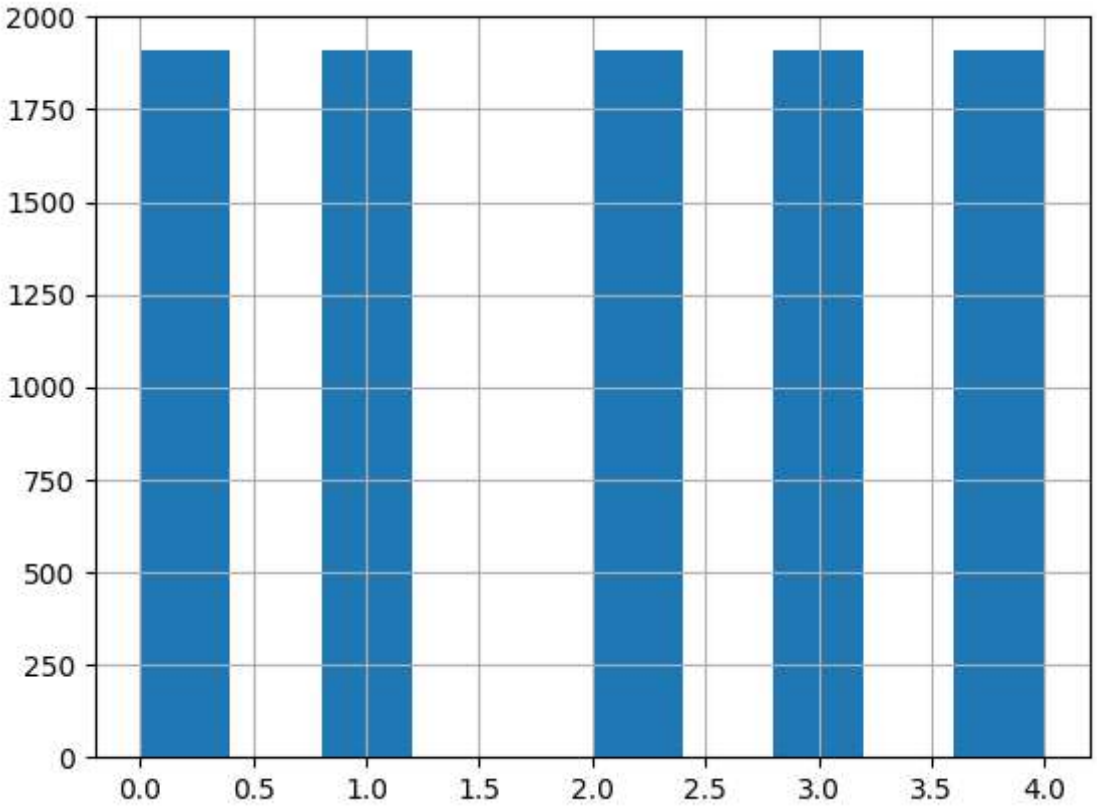
In [ ]:
```
df["review"].hist()
```

Out[ ]:
```
<Axes: >
```



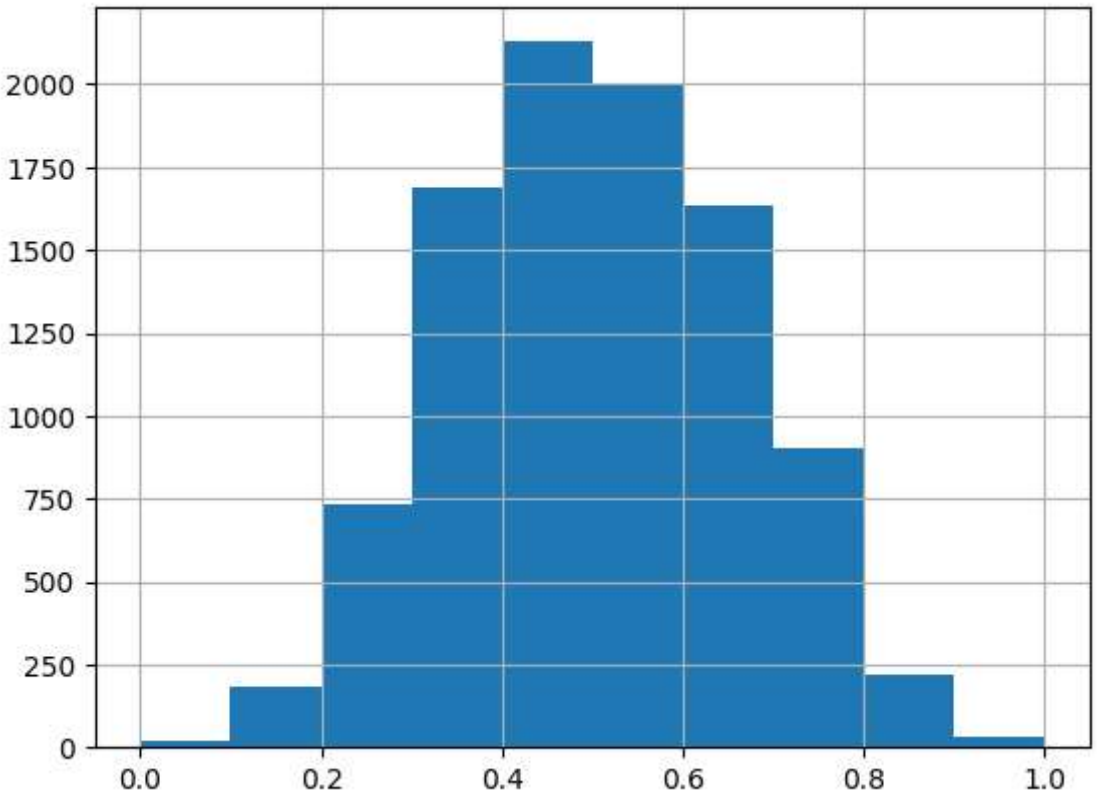
In [ ]:
```
from sklearn.preprocessing import KBinsDiscretizer

discretizer = KBinsDiscretizer(n_bins=5, encode="ordinal")
df["review"] = discretizer.fit_transform(df["review"].to_numpy().reshape(-1, 1))
df["review"].hist()
```

Out[ ]: <Axes: >



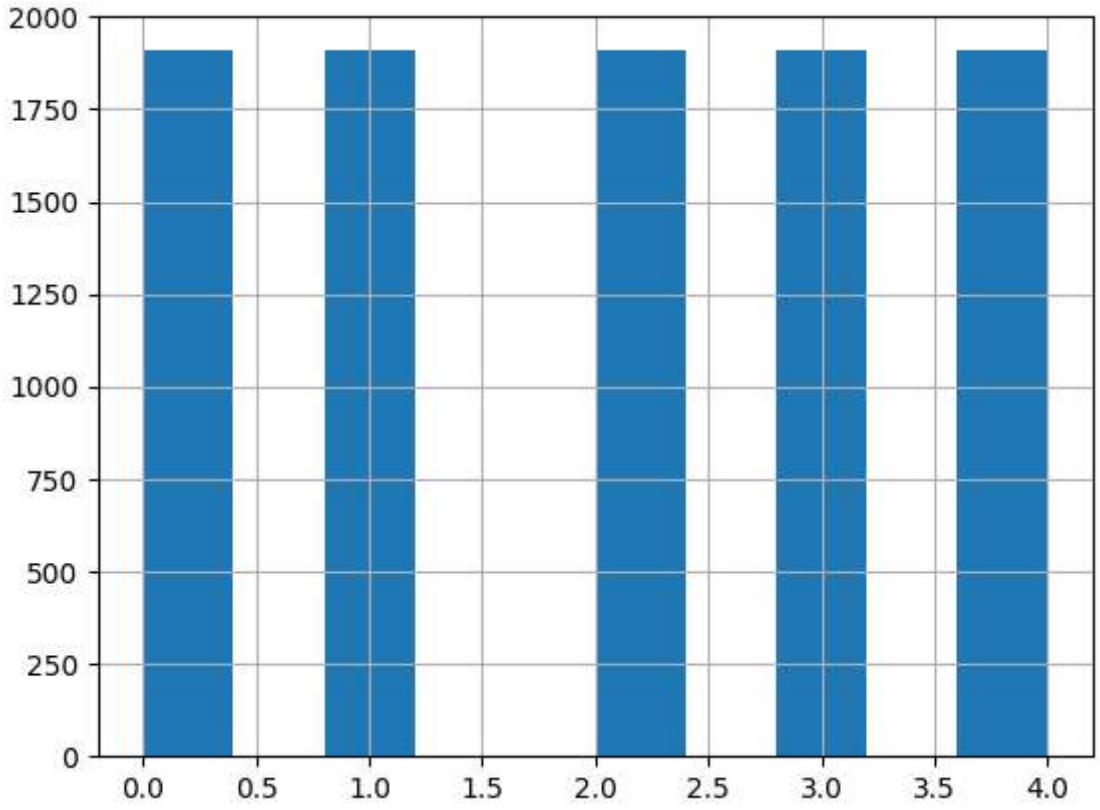
In [ ]:
```
df['satisfaction'].hist()
```

Out[ ]: <Axes: >



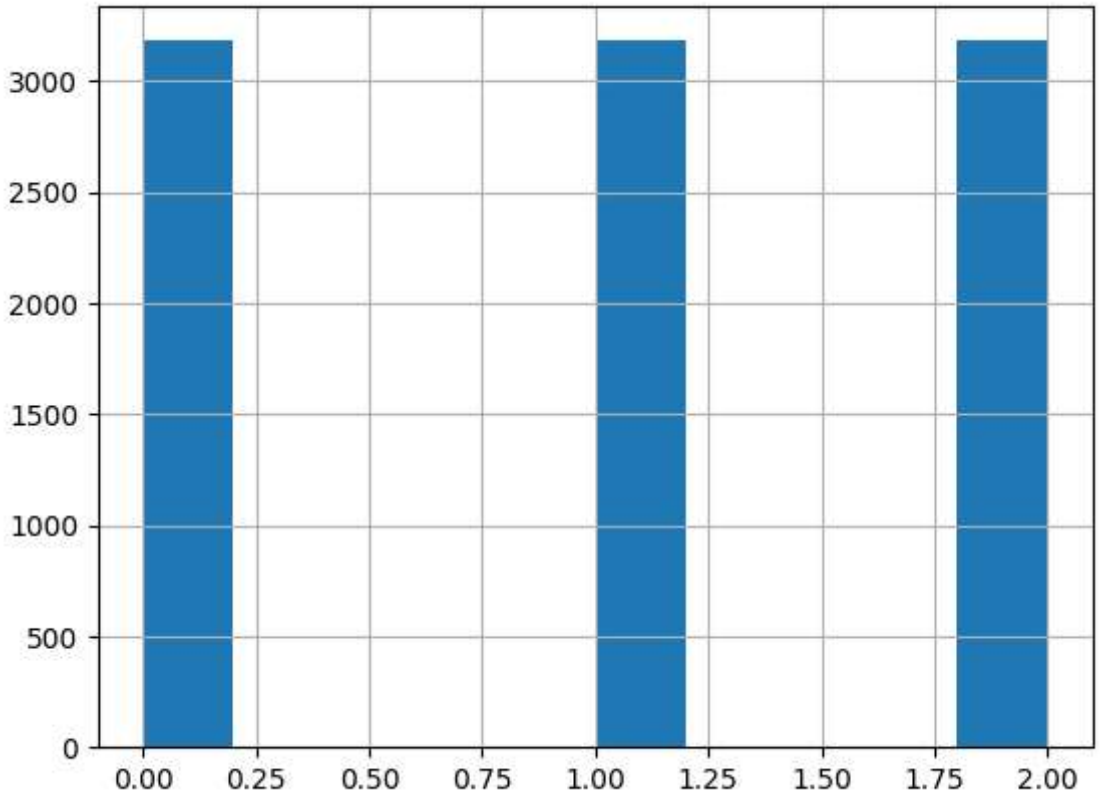
In [ ]:
```
discretizer = KBinsDiscretizer(n_bins=5, encode="ordinal")
df["satisfaction"] = discretizer.fit_transform(df["satisfaction"].to_numpy().reshap
df["satisfaction"].hist()
```

Out[ ]: <Axes: >



In [ ]:
```
discretizer = KBinsDiscretizer(n_bins=3, encode="ordinal")
df["avg_hrs_month"] = discretizer.fit_transform(df["avg_hrs_month"].to_numpy().resh
df["avg_hrs_month"].hist()
```

Out[ ]: <Axes: >



In [ ]:
```
most_dep = df["department"].value_counts().max()
lst = [df]
for class_index, group in df.groupby('department'):
    lst.append(group.sample(most_dep-len(group), replace=True))
df = pd.concat(lst)
```

In [ ]:
```
from sklearn.feature_extraction.text import HashingVectorizer

hasher = HashingVectorizer(n_features=10, binary=True)
encoded = hasher.fit_transform(df["department"])
encoded = pd.DataFrame(encoded.A, columns=[f"department_{i}" for i in range(hasher.
df = pd.concat([encoded.set_index(df.index),df], axis=1)
df = df.drop(["department"],axis=1)
df
```

Out[ ]:

	department_0	department_1	department_2	department_3	department_4	departme
0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	
2	0.0	1.0	0.0	0.0	0.0	
3	0.0	0.0	1.0	0.0	0.0	
4	0.0	0.0	1.0	0.0	0.0	
...	...	...	...	...	...	
6059	0.0	1.0	0.0	0.0	0.0	
3652	0.0	1.0	0.0	0.0	0.0	
2341	0.0	1.0	0.0	0.0	0.0	
9418	0.0	1.0	0.0	0.0	0.0	
7878	0.0	1.0	0.0	0.0	0.0	

18830 rows × 19 columns

In [ ]:
```
df.to_csv("employee_churn_data_clearned.csv", index= False)
```

In [ ]:
```
df
```

Out[ ]:

	department_0	department_1	department_2	department_3	department_4	departme
0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	
2	0.0	1.0	0.0	0.0	0.0	
3	0.0	0.0	1.0	0.0	0.0	
4	0.0	0.0	1.0	0.0	0.0	
...	...	...	...	...	...	
6059	0.0	1.0	0.0	0.0	0.0	
3652	0.0	1.0	0.0	0.0	0.0	
2341	0.0	1.0	0.0	0.0	0.0	
9418	0.0	1.0	0.0	0.0	0.0	
7878	0.0	1.0	0.0	0.0	0.0	

18830 rows × 19 columns

In [ ]:
```
df["avg_hrs_month"].describe()
```

Out[ ]:
```
count    18830.000000
mean         0.997663
std          0.814583
min          0.000000
25%          0.000000
50%          1.000000
75%          2.000000
max          2.000000
Name: avg_hrs_month, dtype: float64
```