campusx-official / 100-days-of-machine-learning Public

Code Issues 3 Pull requests Actions Projects Security Insights

main

100-days-of-machine-learning / day37-handling-missing-categorical-data / frequent-value-imputation.ipynb

campusx-official Add files via upload History

1 contributor

601 lines (601 sloc) | 114 KB

In [37]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [38]:
```
df = pd.read_csv('train.csv',usecols=['GarageQual','FireplaceQu','SalePrice'
```

In [39]:
```
df.head()
```

Out[39]:

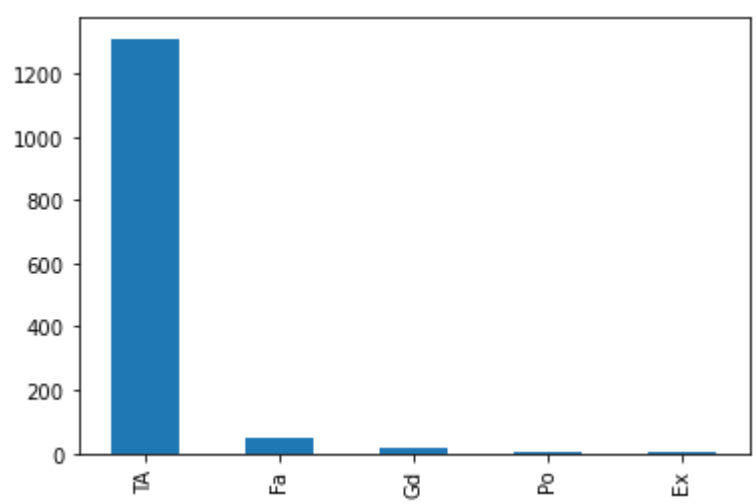
	FireplaceQu	GarageQual	SalePrice
0	NaN	TA	208500
1	TA	TA	181500
2	TA	TA	223500
3	Gd	TA	140000
4	TA	TA	250000

In [41]:
```
df.isnull().mean()*100
```

Out[41]:
```
FireplaceQu    47.260274
GarageQual      5.547945
SalePrice       0.000000
dtype: float64
```

In [42]:
```
df['GarageQual'].value_counts().plot(kind='bar')
```

Out[42]:



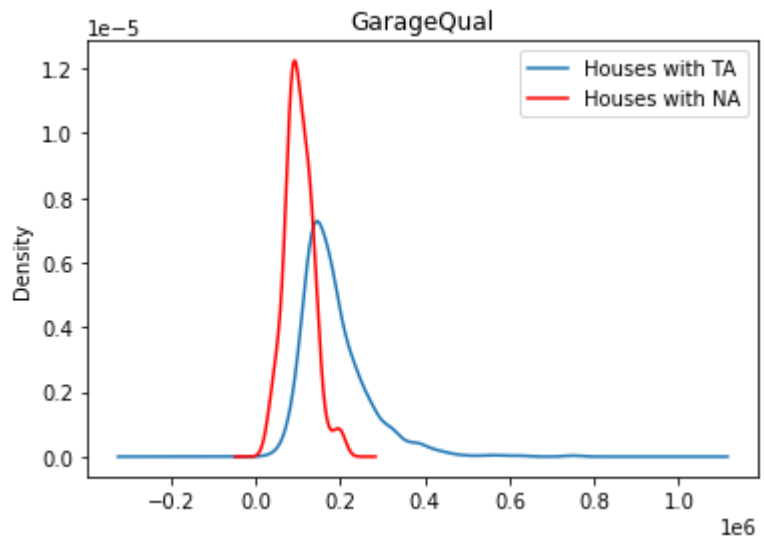
In [43]:
```
df['GarageQual'].mode()
```

Out[43]:
```
0    TA
dtype: object
```

In [44]:

```
fig = plt.figure()
ax = fig.add_subplot(111)

df[df['GarageQual']=='TA']['SalePrice'].plot(kind='kde', ax=ax)

df[df['GarageQual'].isnull()]['SalePrice'].plot(kind='kde', ax=ax, color='re

lines, labels = ax.get_legend_handles_labels()
labels = ['Houses with TA', 'Houses with NA']
ax.legend(lines, labels, loc='best')

plt.title('GarageQual')
```

Out[44]: Text(0.5, 1.0, 'GarageQual')



In [45]:

```
temp = df[df['GarageQual']=='TA']['SalePrice']
```

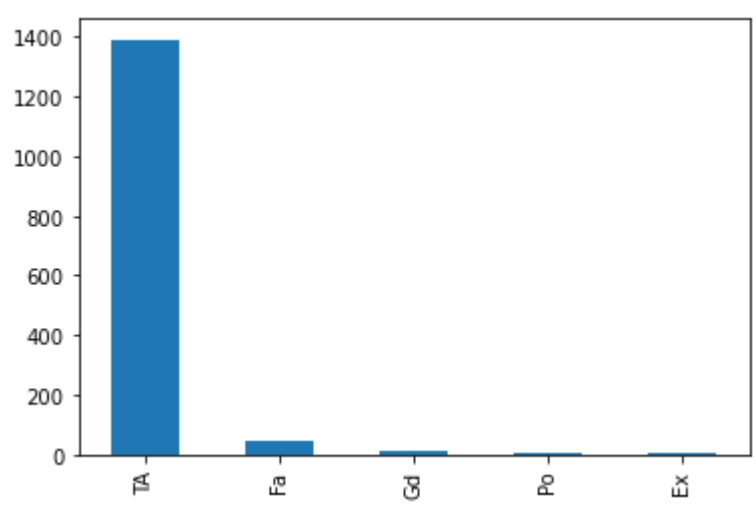
In [46]:

```
df['GarageQual'].fillna('TA', inplace=True)
```

In [47]:

```
df['GarageQual'].value_counts().plot(kind='bar')
```

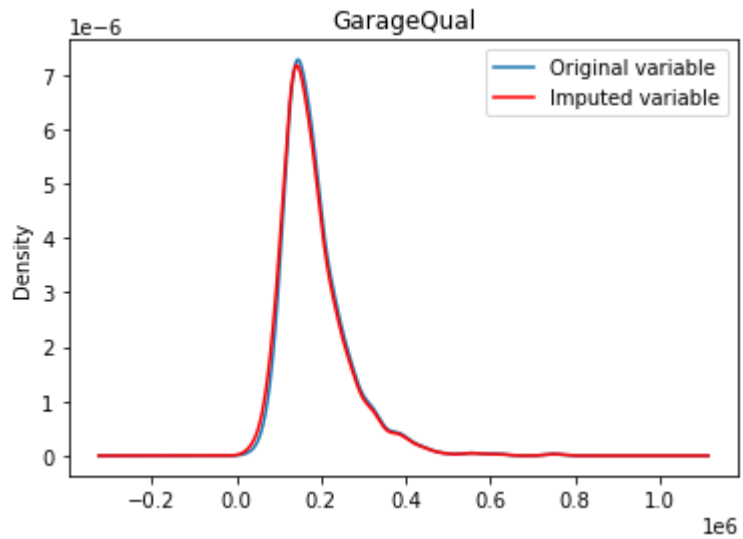
Out[47]:



In [48]:

```
fig = plt.figure()
ax = fig.add_subplot(111)


temp.plot(kind='kde', ax=ax)

# distribution of the variable after imputation
df[df['GarageQual'] == 'TA']['SalePrice'].plot(kind='kde', ax=ax, color='red

lines, labels = ax.get_legend_handles_labels()
labels = ['Original variable', 'Imputed variable']
ax.legend(lines, labels, loc='best')

# add title
plt.title('GarageQual')
```

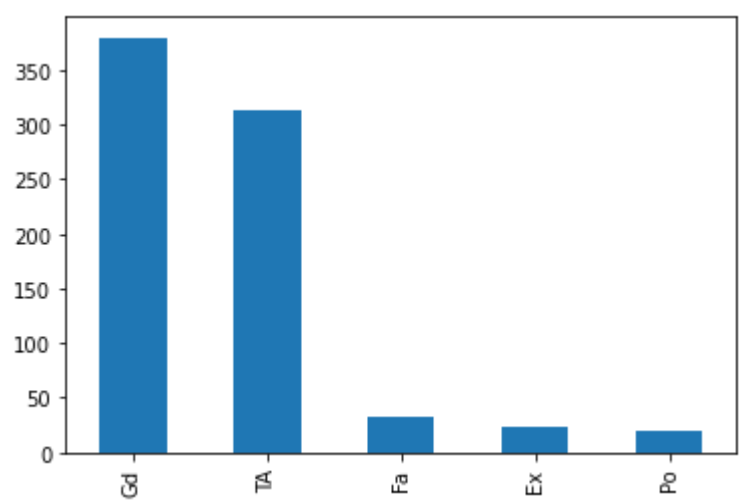
Out[48]: Text(0.5, 1.0, 'GarageQual')



In [49]:

```
df['FireplaceQu'].value_counts().plot(kind='bar')
```

Out[49]:



In [50]:

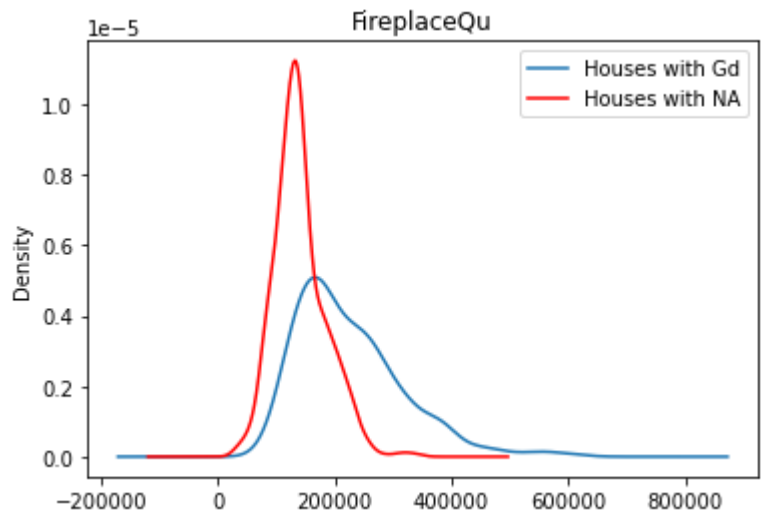
```
df['FireplaceQu'].mode()
```

```
df['FireplaceQu'].mode()
```

Out[50]:
```
0    Gd
dtype: object
```

In [51]:
```
fig = plt.figure()
ax = fig.add_subplot(111)

df[df['FireplaceQu']=='Gd']['SalePrice'].plot(kind='kde', ax=ax)

df[df['FireplaceQu'].isnull()]['SalePrice'].plot(kind='kde', ax=ax, color='r

lines, labels = ax.get_legend_handles_labels()
labels = ['Houses with Gd', 'Houses with NA']
ax.legend(lines, labels, loc='best')

plt.title('FireplaceQu')
```

Out[51]:
```
Text(0.5, 1.0, 'FireplaceQu')
```

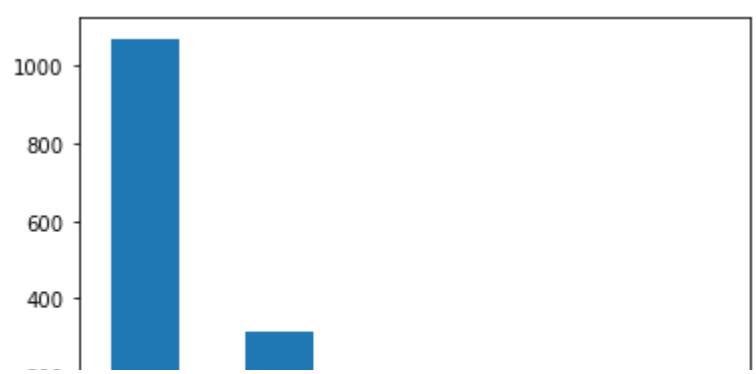


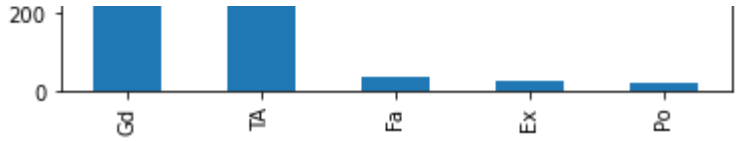
In [52]:
```
temp = df[df['FireplaceQu']=='Gd']['SalePrice']
```

In [53]:
```
df['FireplaceQu'].fillna('Gd', inplace=True)
```

In [54]:
```
df['FireplaceQu'].value_counts().plot(kind='bar')
```

Out[54]:

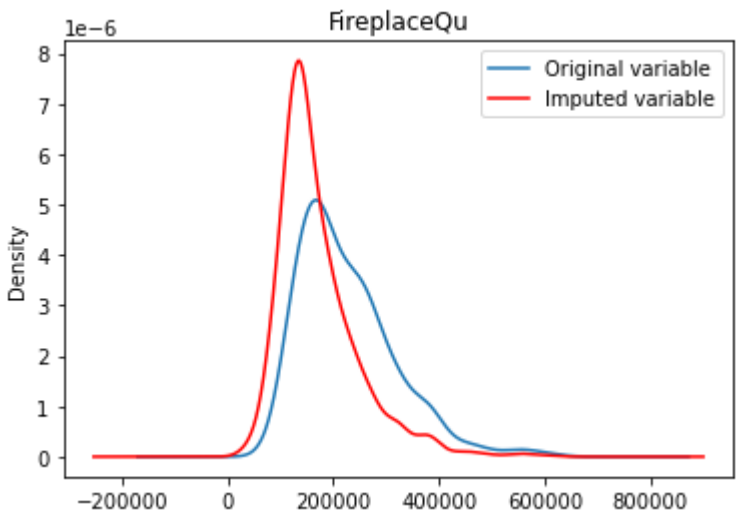




In [55]:

```
fig = plt.figure()
ax = fig.add_subplot(111)


temp.plot(kind='kde', ax=ax)

# distribution of the variable after imputation
df[df['FireplaceQu'] == 'Gd']['SalePrice'].plot(kind='kde', ax=ax, color='re

lines, labels = ax.get_legend_handles_labels()
labels = ['Original variable', 'Imputed variable']
ax.legend(lines, labels, loc='best')

# add title
plt.title('FireplaceQu')
```

Out[55]: Text(0.5, 1.0, 'FireplaceQu')



In [56]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(df.drop(columns=['SalePrice
```

In [57]:

```
from sklearn.impute import SimpleImputer
```

In [58]:

```
imputer = SimpleImputer(strategy='most_frequent')
```

In [59]:

```
X_train = imputer.fit_transform(X_train)
X_test = imputer.transform(X_train)
```

In [60]:

```
imputer.statistics_
```

Out[60]: array(['Gd', 'TA'], dtype=object)

Out[60]: array(['Gd', 'TA'], dtype=object)

In [ ]: