In [8]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import KFold, cross_val_score
from sklearn.metrics import mean_squared_error
from sklearn.impute import SimpleImputer
```

In [9]:
```
# Load the train and test datasets
train_data = pd.read_csv('data/train.csv')
test_data = pd.read_csv('data/test.csv')
```

In [10]:
```
# Identify categorical variables and one-hot encode them
cat_cols = train_data.select_dtypes(include=['object']).columns
train_data = pd.get_dummies(train_data, columns=cat_cols)
test_data = pd.get_dummies(test_data, columns=cat_cols)
```

In [11]:
```
all_data = pd.concat([train_data.drop('SalePrice', axis=1), test_data], axi

# Identify columns with missing values
cols_with_missing = [col for col in all_data.columns if all_data[col].isnul
```

In [13]:
```
# Impute missing values in numeric columns
numeric_cols = all_data.select_dtypes(include=['float64', 'int64']).columns
numeric_cols_with_missing = list(set(cols_with_missing).intersection(numeri
imputer = SimpleImputer(strategy='mean')
all_data[numeric_cols_with_missing] = imputer.fit_transform(all_data[numeri
```

In [14]:
```
# Split the data back into training and test sets
X_train = all_data[:len(train_data)]
X_test = all_data[len(train_data):]
y_train = train_data['SalePrice']
```

In [15]:
```
# Define the model
model = RandomForestRegressor(n_estimators=100, random_state=0)
model.fit(X_train, y_train)
```

Out[15]:
```
RandomForestRegressor(random_state=0)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [16]:
```
predictions = model.predict(X_test)
```

In [18]:
```
# Create a submission file
submission = pd.DataFrame({'Id': test_data.Id, 'SalePrice': predictions})
submission.to_csv('submission.csv', index=False)
```