Dog Breed Rank Prediction using Random Forest Regressor

Best Fit a Dog for me?

https://www.kaggle.com/datasets/sarahmichaela/dog-breed-rank-dog-dot-me

Preprocessing

```
# Import necessary libraries
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import math
```

```
# Import dataset
df = pd.read_csv('breed_rank.csv')
df_test = pd.read_csv('train.csv')
```

```
df.rank.info()
print()
df_test.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Series name:: rank, Type: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 9 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Breed    195 non-null    object
 1   2013 Rank  195 non-null  int64
 2   2014 Rank  195 non-null  int64
 3   2015 Rank  195 non-null  int64
 4   2016 Rank  195 non-null  int64
 5   2017 Rank  195 non-null  int64
 6   2018 Rank  195 non-null  int64
 7   2019 Rank  195 non-null  int64
 8   2020 Rank  195 non-null  int64
dtype: int64
memory usage: 13.8 KB
```

	Breed	2013 Rank	2014 Rank	2015 Rank	2016 Rank	2017 Rank	2018 Rank	2019 Rank	2020 Rank
0	Retrievers (Labrador)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1	German Shepherd Dogs	2.0	2.0	2.0	2.0	2.0	2.0	2.0	3.0
2	Retrievers (Golden)	3.0	3.0	3.0	3.0	3.0	3.0	3.0	4.0
3	Beagles	4.0	5.0	5.0	5.0	5.0	6.0	7.0	7.0
4	Bulldogs	5.0	4.0	4.0	5.0	5.0	5.0	5.0	5.0

```
df.info()
print()
df_test.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Breed                  195 non-null    object
 1   Affectionate With Family 195 non-null  int64
 2   Good With Young Children 195 non-null  int64
 3   Good With Other Dogs   195 non-null    int64
 4   Shedding Level         195 non-null    int64
 5   Coat Grooming Frequency 195 non-null   int64
 6   Drooling Level         195 non-null    int64
 7   Coat Type              195 non-null    int64
 8   Coat Length            195 non-null    int64
 9   Openness To Strangers  195 non-null    int64
dtype: int64
memory usage: 15.3 KB
```

	Breed	Affectionate With Family	Good With Young Children	Good With Other Dogs	Shedding Level	Coat Grooming Frequency	Drooling Level	Coat Type	Coat Length	Openness To Strangers
0	Retrievers (Labrador)	5	5	5	4	2	3	Double	Short	5
1	French Bulldogs	5	5	4	3	1	2	Smooth	Short	5
2	German Shepherd Dogs	4	5	4	4	3	2	Double	Medium	4
3	Retrievers (Golden)	5	5	4	4	3	2	Double	Medium	5
4	Bulldogs	5	4	3	1	1	4	Smooth	Short	4

```
# Combine train and values
df = pd.concat([df, df_test])[['breed', 'rank']].set_index('')
```

```
# Combine the dataset based on breed column
df = df.merge(df_train, on='breed').drop('breed').dropna()
df.head()
```

	Breed	Affectionate With Family	Good With Young Children	Good With Other Dogs	Shedding Level	Coat Grooming Frequency	Drooling Level	Coat Type	Coat Length	Openness To Strangers	Playfulness Level	Watchdog/Protective Nature	Adaptability Level	Trainability Level	Energy Level	Barking Level	Mental Stimulation Needs	rank
0	Retrievers (Labrador)	5.0	5.0	5.0	4.0	2.0	3.0	Double	Short	5.0	5.0	3.0	5.0	5.0	4.0	3.0	5.0	1.0
1	French Bulldogs	5.0	5.0	4.0	3.0	1.0	2.0	Smooth	Short	5.0	5.0	3.0	5.0	4.0	3.0	1.0	2.0	22.0
2	German Shepherd Dogs	4.0	5.0	4.0	4.0	3.0	2.0	Double	Medium	4.0	5.0	5.0	4.0	5.0	4.0	3.0	4.0	2.0
3	Retrievers (Golden)	5.0	5.0	4.0	4.0	3.0	2.0	Double	Medium	5.0	4.0	4.0	5.0	4.0	4.0	3.0	4.0	3.0
4	Bulldogs	5.0	4.0	3.0	1.0	1.0	4.0	Smooth	Short	4.0	3.0	5.0	3.0	1.0	1.0	5.0	5.0	4.0

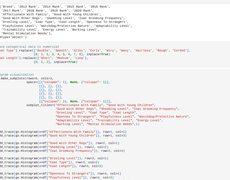
5 rows × 25 columns

```
# Replace NaN with 0
df = df.fillna(0)
```

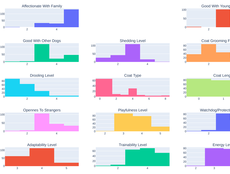
Visualization

```
# Create a boxplot for the first breed "Retrievers (Labrador)"
df[ df['Affectionate With Family', 'Good With Young Children', 'Good With Other Dogs', 'Shedding Level',
    'Coat Grooming Frequency', 'Drooling Level', 'Coat Type', 'Coat Length', 'Openness To Strangers',
    'Playfulness Level', 'Watchdog/Protective Nature', 'Adaptability Level', 'Trainability Level', 'Energy Level',
    'Barking Level', 'Mental Stimulation Needs']
    ].iloc[0].T.hvplot.line('Retrievers (Labrador)')[:15], ylim=(1, 5), barmode='group']
fig.show()
```

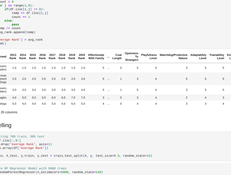
```
# Fig 1 as DataFrame
fig = df[ df['Breed'].isin(['Retrievers (Labrador)'])].T
fig.show()
```




```
# Create a boxplot for the fixed breed "Pumis"
fig = df[ df['Breed'].isin(['Pumis'])].iloc[:5].T, barmode='group']
fig.show()
```




```
# Fig 2 as DataFrame
fig = df[ df['Breed'].isin(['Pumis'])], y=[0:7], barmode='group']
fig.show()
```




```
# Correlation heatmap between variables
df[df.columns[1:-1]].corr().sort("PRC0", axis=False, inplace=True)
df[:-1].T
```

```
df.columns
```

```
Index(['Breed', '2013 Rank', '2014 Rank', '2015 Rank', '2016 Rank',
       '2017 Rank', '2018 Rank', '2019 Rank', '2020 Rank',
       'Affectionate With Family', 'Good With Young Children',
       'Good With Other Dogs', 'Shedding Level', 'Coat Grooming Frequency',
       'Drooling Level', 'Coat Type', 'Coat Length', 'Openness To Strangers',
       'Playfulness Level', 'Watchdog/Protective Nature', 'Adaptability Level',
       'Trainability Level', 'Energy Level', 'Barking Level',
       'Mental Stimulation Needs', 'rank'],
      dtype='object')
```

```
# Replace categorical data in numerical
df['Coat Type'] = df[['Coat Type', 'Coat', 'Hair', 'Wire', 'Hairless', 'Rough', 'Curled'
    ].T.idxmax(axis=1).map({ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
```