Иниятуллин Р.М. ИУ5Ц-83Б

```
[11] import pandas as pd
     import seaborn as sb
     import numpy as np
     import matplotlib.pyplot as plt
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import LabelEncoder, StandardScaler
     from sklearn.tree import DecisionTreeRegressor
     from sklearn.ensemble import RandomForestRegressor
     from sklearn.metrics import mean_absolute_error, mean_squared_error

     # Загрузка датасета
     try:
         df = pd.read_csv('Billionaire.csv', delimiter=',')
         print('Загружен датасет')
     except Exception as ex:
         print('Отсутствует датасет. Проверьте путь файла')
         print('Error:', ex)

     # Приведение названий всех колонок к нижнему регистру
     df.columns = df.columns.str.lower()

     # Вывод информации о данных
     df.info()

     # Преобразование столбца "NetWorth" в числовой формат
     df['networth'] = df['networth'].str.replace('$', '').str.replace('B', '').astype(float)

     # Кодирование категориальных признаков
     le = LabelEncoder()
     df['country'] = le.fit_transform(df['country'])
     df['source'] = le.fit_transform(df['source'])
     df['industry'] = le.fit_transform(df['industry'])

     # Удаление ненужных колонок
     df = df.drop(columns=['name'])

     # Заполнение пропусков, если они есть
     df.fillna(df.mean(), inplace=True)
```

```
# Разделение на признаки и целевую переменную
X = df.drop(columns=['networth'])
y = df['networth']

# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
Загружен датасет
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2755 entries, 0 to 2754
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   name      2755 non-null   object
 1   networth  2755 non-null   object
 2   country   2755 non-null   object
 3   source    2755 non-null   object
 4   rank      2755 non-null   int64
 5   age       2676 non-null   float64
 6   industry  2755 non-null   object
dtypes: float64(1), int64(1), object(5)
memory usage: 150.8+ KB
```

```
[12] # Обучение модели дерева решений
     tree_reg = DecisionTreeRegressor(random_state=42)
     tree_reg.fit(X_train, y_train)

     # Предсказания для дерева решений
     y_pred_tree = tree_reg.predict(X_test)
```

```
# Обучение модели случайного леса
rf_reg = RandomForestRegressor(n_estimators=100, random_state=42)
rf_reg.fit(X_train, y_train)

# Предсказания для случайного леса
y_pred_rf = rf_reg.predict(X_test)
```

```
# Оценка качества для дерева решений
mae_tree = mean_absolute_error(y_test, y_pred_tree)
mse_tree = mean_squared_error(y_test, y_pred_tree)
print("Decision Tree Regressor:")
print("Mean Absolute Error (MAE):", mae_tree)
print("Mean Squared Error (MSE):", mse_tree)

# Оценка качества для случайного леса
mae_rf = mean_absolute_error(y_test, y_pred_rf)
mse_rf = mean_squared_error(y_test, y_pred_rf)
print("\nRandom Forest Regressor:")
print("Mean Absolute Error (MAE):", mae_rf)
print("Mean Squared Error (MSE):", mse_rf)
```

```
Decision Tree Regressor:
Mean Absolute Error (MAE): 0.027041742286751873
Mean Squared Error (MSE): 0.03268602540834845

Random Forest Regressor:
Mean Absolute Error (MAE): 0.01283666061706167
Mean Squared Error (MSE): 0.007170299455535008
```