Analysis of Power Generation and Classification of Census Regions of United States Power Plants

Vedant Gannu (gannuv@rpi.edu)

[1]Rensselaer Polytechnic Institute, Tetherless World Constellation, Troy, NY, United States,

Abstract

Power plants are an engineering marvel that are extremely vital to the general population as they are the primary generators of manmade electricity. According to the EIA, about 4, 116 billion kilowatthours (kWh) or 4.12 trillion kWh were generated by United States utility-scale electricity generation facilities in 2021. Around 61% of this electricity generation was from fossil fuels—coal, natural gas, petroleum, and other gases. About 19% was from nuclear energy, and about 20% was from renewable energy sources. The three major fossil fuels that contribute to this 61% are Natural Gas, Coal, and Petroleum whereas the renewable sources are contributed by Wind, Hydropower, Solar (including photovoltaic and thermal), Biomass (Wood, Landfill Gas, Biogenic waste, other kinds of biomass), and Geothermal. With this incredible generation capacity of just the United States alone to be able to supply electricity to the continually growing electric demands of the general population and technology, the major overhead cost the planet pays for is in the form of CO_2, NO_2, and SO_2 emissions into the atmosphere. In 2020 the EIA reported that "all energy sources resulted in the emission of 1.55 billion metric tons—1.71 billion short tons—of carbon dioxide (CO_2)" [2] of which coal, natural gas, and petroleum fuels accounted for 99% of those emissions. Given this capacity to generate electricity, analyzing the efficiency and capacity of net electricity generation is very important.

Motivation

1. Analyze the net electricity generation of United States power plants attributed to their overall fuel consumption and total fuel consumption towards electricity generation, and to see if there is regression can be used to model it.
2. The second goal of this project is to try and determine if certain power plant generation information such as net electricity generation, fuel consumed for electricity generation, fuel type, and even the year of the data can be used to classify where the specific power plant was surveyed.

Clean

- Gather data from spreadsheet files for years 2018-2021. (Sheet: Generation and Fuel Data)
- Drop rows with 0 net generation and nonzero Elec Fuel Consumption MMBtu

Inspect

- Gather subset of fuel types
- Visualize the distributions for each AER fuel type by year:
  - Total Fuel Consumption MMBtu
  - Elec Fuel Consumption MMBtu

Outliers

- Aggregate net generation, Elec Fuel Consumption MMBtu, and Total Fuel Consumption MMBtu by plant id, year, and fuel
- Consider Net Generation within (Q3-Q3)*0.5 and (Q3-Q3)*1.5

Model

- Multiple Regression
- Random Forest

Sponsors:

Resources:

Python packages**:**

Sci-kit Learn – Linear Regression and RandomForest
Seaborn – Boxplot and Linear visualizations
Statsmodels – Linear Regression Scoring
Pandas – Data cleaning and aggregation

Exploratory Data Analysis and Modeling

1. Import data:

A. Spreadsheet files

```
gen_fuel_dataset2018 = pd.read_excel(r"EIA923_Schedules_2_3_4_5_M_12_2018_Final_Revision.xlsx", sheet_name="Page 1 Generation and Fuel Data",
                                     header=5)
gen_fuel_dataset2019 = pd.read_excel(r"EIA923_2019_Final_Revision.xlsx", sheet_name="Page 1 Generation and Fuel Data",
                                     header=5)
gen_fuel_dataset2020 = pd.read_excel(r"EIA923_Schedules_2_3_4_5_M_12_2020_Final_Revision.xlsx", sheet_name="Page 1 Generation and Fuel Data",
                                     header=5)
gen_fuel_dataset2021 = pd.read_excel(r"EIA923_Schedules_2_3_4_5_M_12_2021_18FEB2022.xlsx", sheet_name="Page 1 Generation and Fuel Data",
                                     header=5)
```

2. Inspect:

A. Subset and Aggregate

```
renewable_sources_aer = ["SUN", "GEO", "HPS", "HYC", "MLP", "NUC", "ORW", "WND", "WWW"]
nonrenewable_sources_aer = ["COL", "DFO", "NG", "OOG", "PC", "RFO", "WOC", "WOO", "OTH"]
agg_gen_fuel_dataset_2018 = gen_fuel_dataset2018[~(gen_fuel_dataset2018["aer_fuel_type_code"].isin(["OTH"])
                                                 & gen_fuel_dataset2018["reported_fuel_type_code"].isin(["OTH", "PUR", "MWH"]))]
agg_gen_fuel_dataset_2019 = gen_fuel_dataset2019[~(gen_fuel_dataset2019["aer_fuel_type_code"].isin(["OTH"])
                                                 & gen_fuel_dataset2019["reported_fuel_type_code"].isin(["OTH", "PUR", "MWH"]))]
agg_gen_fuel_dataset_2020 = gen_fuel_dataset2020[~(gen_fuel_dataset2020["aer_fuel_type_code"].isin(["OTH"])
                                                 & gen_fuel_dataset2020["reported_fuel_type_code"].isin(["OTH", "PUR", "MWH"]))]
agg_gen_fuel_dataset_2021 = gen_fuel_dataset2021[~(gen_fuel_dataset2021["aer_fuel_type_code"].isin(["OTH"])
                                                 & gen_fuel_dataset2021["reported_fuel_type_code"].isin(["OTH", "PUR", "MWH"]))]


columns_interest = [colname  for colname in gen_fuel_dataset2019.columns if colname.find("elec_quantity") == -1\
                    and colname.find("quantity_") == -1 and colname.find("netgen_") == -1 and colname.find("elec_mmbtu") == -1\
                       and colname.find("mmbtuper_") == -1 and colname.find("tot_mmbtu") == -1 and colname.find("reserved") == -1]
#Selecting columns of interest and removing records where electricity was not generated even though fuel was consumed
agg_gen_fuel_dataset_2018 = agg_gen_fuel_dataset_2018.loc[
    ~(agg_gen_fuel_dataset_2018["net_generation_(megawatthours)"] == 0) & (agg_gen_fuel_dataset_2018["elec_fuel_consumption_mmbtu"] != 0),
agg_gen_fuel_dataset_2019 = agg_gen_fuel_dataset_2019.loc[
    ~(agg_gen_fuel_dataset_2019["net_generation_(megawatthours)"] == 0) & (agg_gen_fuel_dataset_2019["elec_fuel_consumption_mmbtu"] != 0),
agg_gen_fuel_dataset_2020 = agg_gen_fuel_dataset_2020.loc[
    ~(agg_gen_fuel_dataset_2020["net_generation_(megawatthours)"] == 0) & (agg_gen_fuel_dataset_2020["elec_fuel_consumption_mmbtu"] != 0),
agg_gen_fuel_dataset_2021 = agg_gen_fuel_dataset_2021.loc[
    ~(agg_gen_fuel_dataset_2021["net_generation_(megawatthours)"] == 0) & (agg_gen_fuel_dataset_2021["elec_fuel_consumption_mmbtu"] != 0),
```

B. Visualize

```
plt.subplots(figsize=(30,15))
plt.ticklabel_format(style='plain')
sns.boxplot(x="aer_fuel_type_code", y="net_generation_(megawatthours)", hue="year",
                data=\
               agg_gen_fuel_dataset[agg_gen_fuel_dataset["aer_fuel_type_code"].isin(nonrenewable_sources_aer)], palette="Set3")

plt.subplots(figsize=(30,15))
plt.ticklabel_format(style='plain')
sns.boxplot(x="aer_fuel_type_code", y="net_generation_(megawatthours)", hue="year",
                data=\
               agg_gen_fuel_dataset[agg_gen_fuel_dataset["aer_fuel_type_code"].isin(renewable_sources_aer)], palette="Set3")
```

3. Outlier:

A. Re-Aggregate

```
agg_gen_fuel_dataset_grouped = agg_gen_fuel_dataset.groupby(["year", "plant_id",
                               "aer_fuel_type_code"], as_index=False).agg({'net_generation_(megawatthours)':'sum', 'total_fuel_consumption_mmbtu':'sum',
```

B. Visualize

```
plt.subplots(figsize=(30,15))
plt.ticklabel_format(style='plain')
sns.boxplot(x="aer_fuel_type_code", y="net_generation_(megawatthours)", hue="year",
                data=\
               agg_gen_fuel_dataset_grouped[agg_gen_fuel_dataset_grouped["aer_fuel_type_code"].isin(nonrenewable_sources_aer)],

plt.subplots(figsize=(30,15))
plt.ticklabel_format(style='plain')
sns.boxplot(x="aer_fuel_type_code", y="net_generation_(megawatthours)", hue="year",
                data=\
               agg_gen_fuel_dataset_grouped[agg_gen_fuel_dataset_grouped["aer_fuel_type_code"].isin(renewable_sources_aer)], pa
```

C. Remove Outliers

```
describe1 = agg_gen_fuel_dataset_grouped.groupby(["year", "aer_fuel_type_code"])["net_generation_(megawatthours)"].describe()
outlier_range1 = ((describe1["75%"] - describe1["25%"]) * 0.8).reset_index()
outlier_range1[1] = ((describe1["75%"] - describe1["25%"]) * 1.8).reset_index()[0]
outlier_range1.rename(columns={0:"lower", 1:"upper"}, inplace=True)

merge1 = pd.merge(testing, outlier_range1, on=["year", "aer_fuel_type_code"], how="inner")
final1 = merge1[(merge1["net_generation_(megawatthours)"] < merge1["upper"] ) & (merge1["net_generation_(megawatthours)"] > merge1["lower"])]
```

D. Revisualize

```
plt.subplots(figsize=(30,15))
plt.ticklabel_format(style='plain')
sns.boxplot(x="aer_fuel_type_code", y="net_generation_(megawatthours)", hue="year",
                data=\
               final1[final1["aer_fuel_type_code"].isin(nonrenewable_sources_aer)], palette="Set3")

plt.subplots(figsize=(30,15))
plt.ticklabel_format(style='plain')
sns.boxplot(x="aer_fuel_type_code", y="net_generation_(megawatthours)", hue="year",
                data=\
               final1[final1["aer_fuel_type_code"].isin(renewable_sources_aer)], palette="Set3")
```

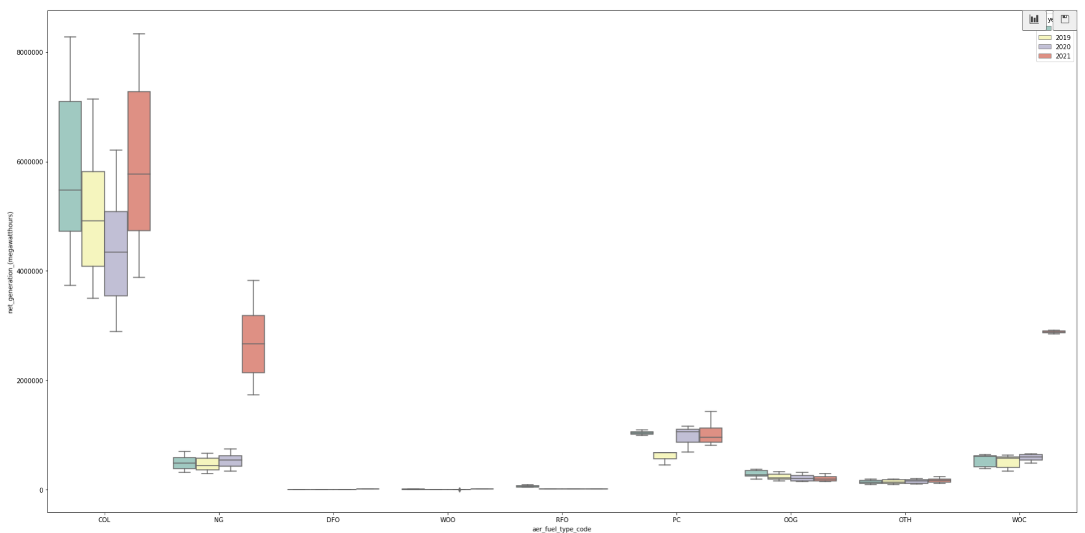
4. Model

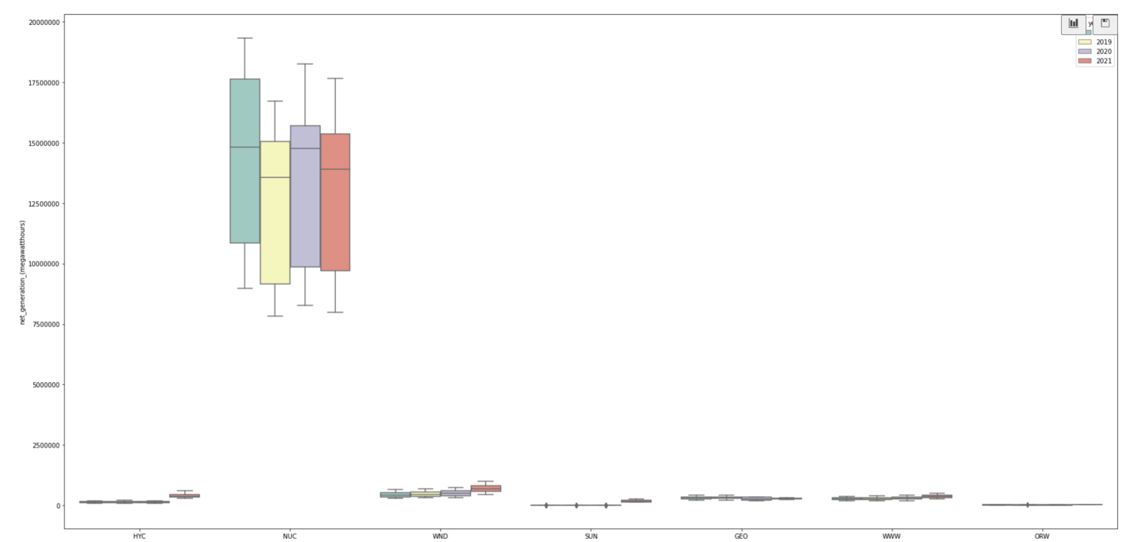
```
model = LinearRegression().fit(x_train, y_train)
results = model.predict(x_test)
print(metrics.mean_squared_error(y_test, results))
print(model.score(x_train, y_train))
model1 = sm.OLS(y_train, sm.add_constant(x_train)).fit()
print(model1.summary())
```

```
clf1=RandomForestClassifier(n_estimators=200)

kf = KFold(n_splits=5, shuffle=True)

scores = cross_val_score(clf1, x_train1, y_train1, scoring="f1_micro", cv=kf)

print(scores)

clf1.fit(x_train1,y_train1)
y_pred1 = clf1.predict(x_test1)
```

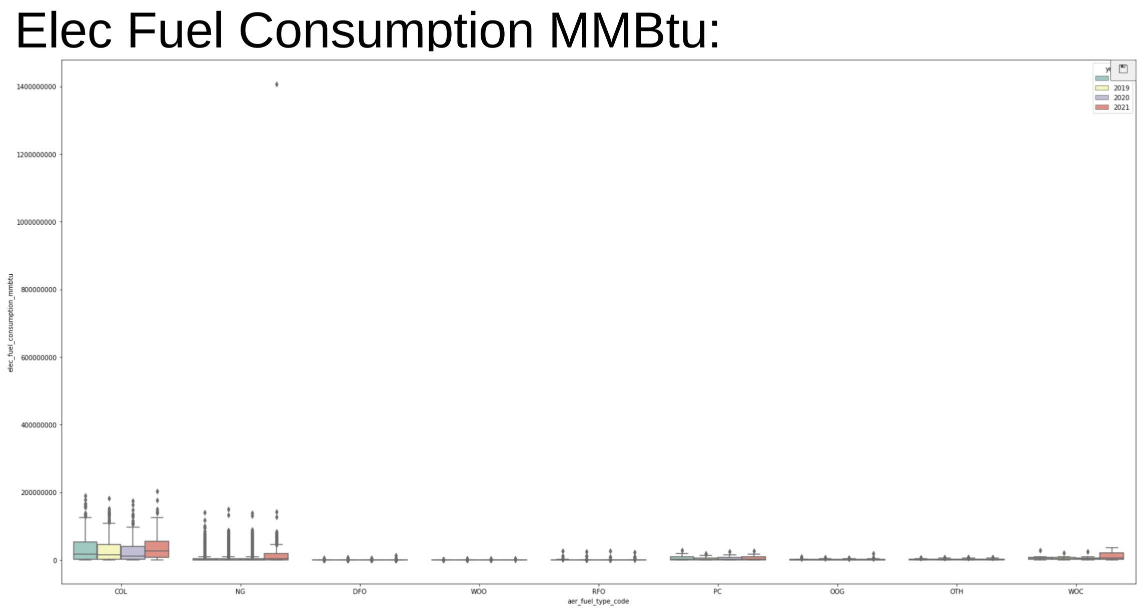
Visualizations

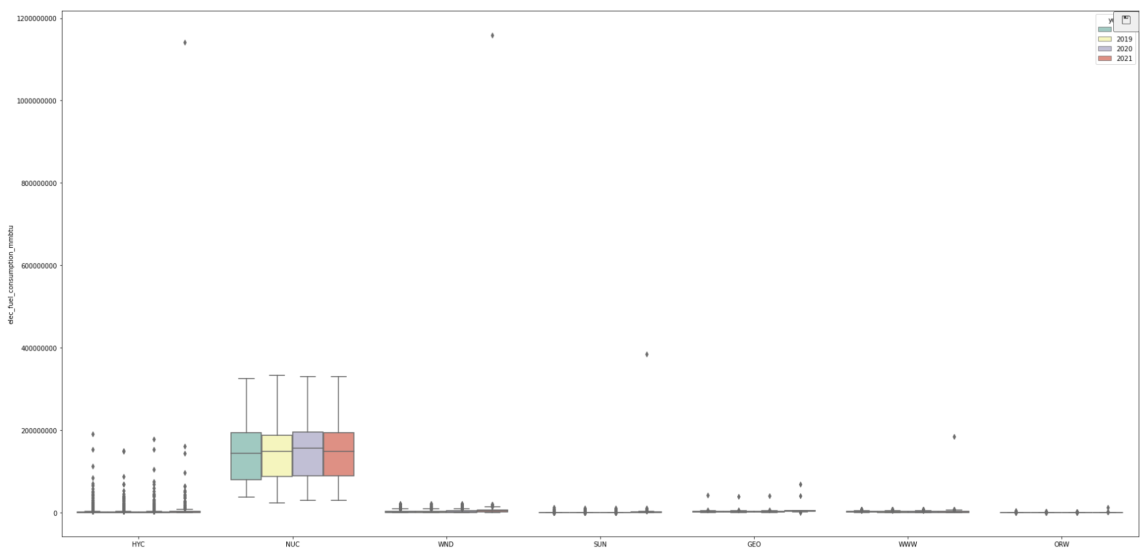
Net Generation (Non-outliers):



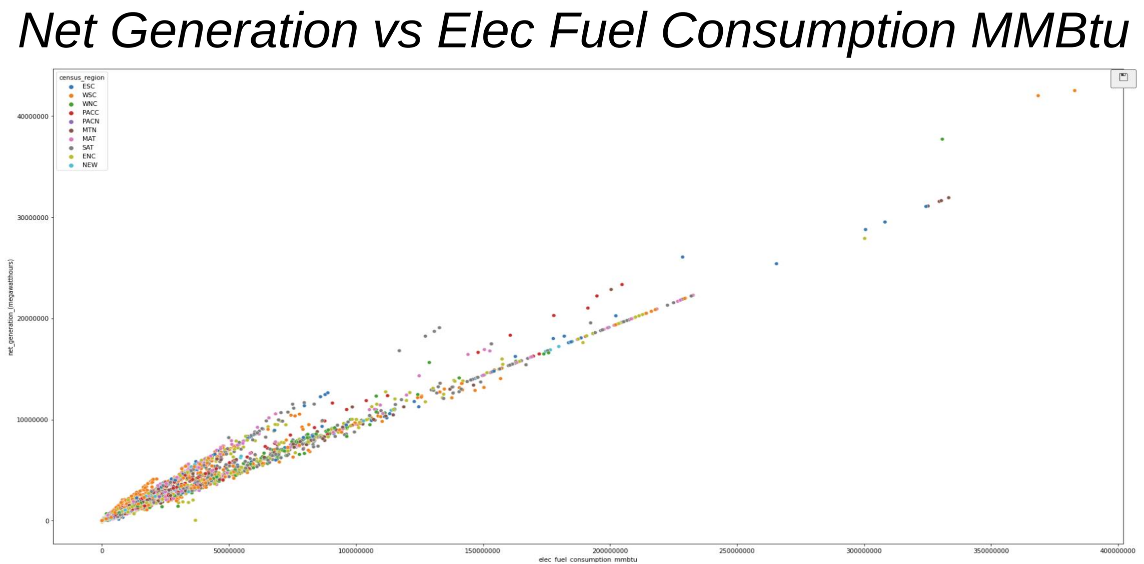


Elec Fuel Consumption MMBtu:





Net Generation vs Elec Fuel Consumption MMBtu



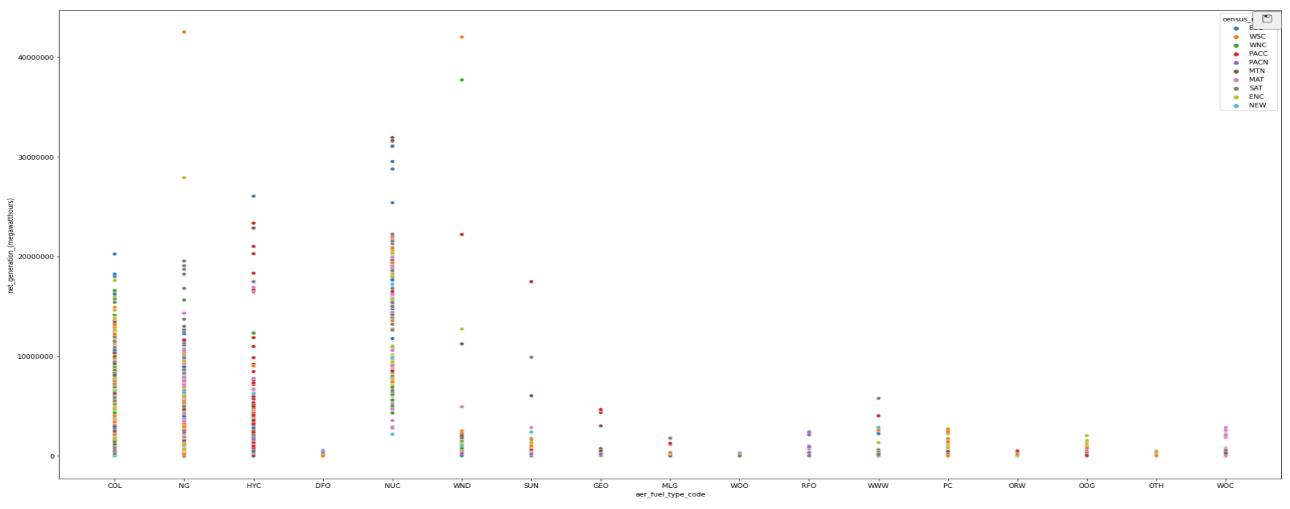
Results

Multiple Regression

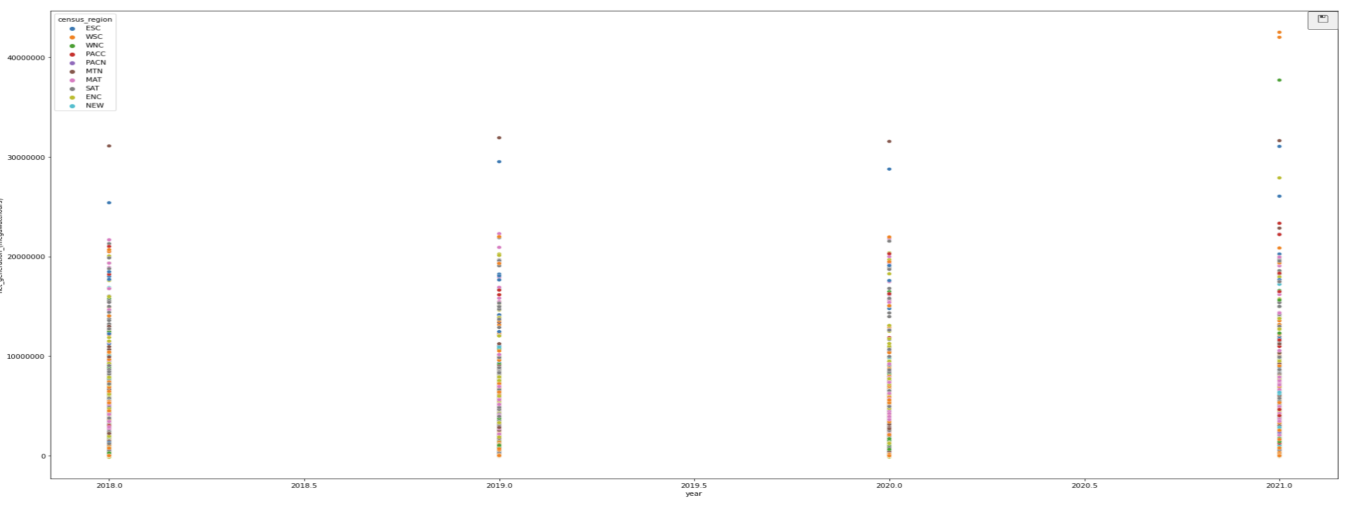
```
                                    coef
-----------------------------------------
const                           1.91e+04
total_fuel_consumption_mmbtu      0.0211
elec_fuel_consumption_mmbtu       0.0747
```

```
R-squared:                   0.995
Adj. R-squared:              0.995
F-statistic:             4.720e+05
Prob (F-statistic):           0.00
Log-Likelihood:            -57538.
AIC:                     1.151e+05
BIC:                     1.151e+05
```

Random Forest:
Net Generation vs AER Fuel Type



Net Generation vs Year



5 Fold Cross Validation F1 scores and Testing F1 score

```
[0.25621022 0.26201232 0.27043121 0.26940452 0.26324435]
```

```
print(metrics.accuracy_score(y_test2, y_pred2))
✓ 0.4s

0.275887943971986
```