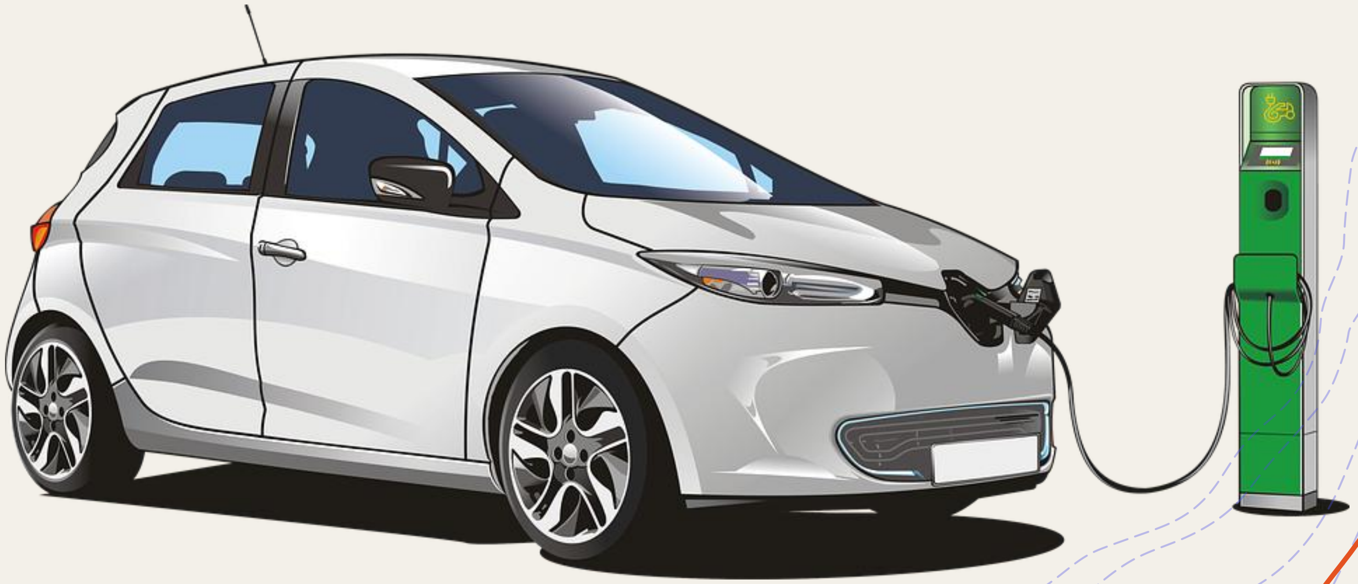
Electric Vehicle Use in Washington State



Chenchen Jiang

11/12/2023

Content

Introduction

Datasets:

- Data.WA.gov: Electric Vehicle Population Data (https://data.wa.gov/Transportation/Electric-Vehicle-Population-Data/f6w7-q2d2)
- Electric Vehicle Registration Activity by Year (https://data.wa.gov/Transportation/Electric-Vehicle-Registration-Activity-by-Year/tak8-xdcp)

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	Vehicle Location	Electric Utility	2020 Census Tract
0	KM8K33AGXL	King	Seattle	WA	98103.0	2020	HYUNDAI	KONA	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	258	0	43.0	249675142	POINT (-122.34301 47.659185)	CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)	5.303300e+10
1	1C4RJYB61N	King	Bothell	WA	98011.0	2022	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	1.0	233928502	POINT (-122.20578 47.762405)	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	5.303302e+10
2	1C4RJYD61P	Yakima	Yakima	WA	98908.0	2023	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	14.0	229675939	POINT (-120.6027202 46.5965625)	PACIFICORP	5.307700e+10
3	5YJ3E1EA7J	King	Kirkland	WA	98034.0	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	215	0	45.0	104714466	POINT (-122.209285 47.71124)	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	5.303302e+10

Make	Model	Vehicle Primary Use	Electric Range	Odometer Reading	Odometer Code	...	Meets 2019 HB 2042 Sale Price/Value Requirement	2019 HB 2042: Battery Range Requirement	2019 HB 2042: Purchase Date Requirement	2019 HB 2042: Sale Price/Value Requirement	Electric Vehicle Fee Paid	Transportation Electrification Fee Paid	Hybrid Vehicle Electrification Fee Paid	2020 Census Tract	Legislative District	Electric Utility
TESLA	Model Y	Passenger	0	15	Actual Mileage	...	False	No battery range	Meets purchase date requirement	Sale price too high	Not Applicable	Not Applicable	Not Applicable	5.303302e+10	48.0	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
TESLA	Model Y	Passenger	0	15	Actual Mileage	...	False	No battery range	Meets purchase date requirement	Sale price too high	Not Applicable	Not Applicable	Not Applicable	5.303301e+10	43.0	CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)
PORSCHE	Taycan	Passenger	0	37	Actual Mileage	...	False	No battery range	Meets purchase date requirement	Sale price too high	Not Applicable	Not Applicable	Not Applicable	5.303302e+10	48.0	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
TESLA	Model Y	Passenger	0	15	Actual Mileage	...	False	No battery range	Meets purchase date requirement	Sale price too high	Not Applicable	Not Applicable	Not Applicable	5.303303e+10	11.0	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
KIA	Niro	Passenger	26	57920	Actual Mileage	...	False	Low battery range	Meets purchase date requirement	Meets title transaction requirement	Not Applicable	Not Applicable	Not Applicable	5.303302e+10	41.0	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)

Use cases :

- Market Research
- Demand Forecasting
- Sales Strategies
- Charging Infrastructure Planning
- Environmental Impact Assessment

Business Objectives

- Make marketing strategies
- Enhance sales predictions
- Contribute to industry development
- Guide charging infrastructure planning

Coding Review–import packages

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd
from matplotlib.animation import FuncAnimation
from IPython.display import HTML
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.tree import plot_tree
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from dmba import plotDecisionTree, classificationSummary
```

Coding Review–import dataset & data clean

```python
# Import the datasets
df_EVPopulation = pd.read_csv('/Users/cheryl/Desktop/IE7275/Project/Dataset/1. Electric_Vehicle_Population_Data.csv')
df_EVRegistration = pd.read_csv('/Users/cheryl/Desktop/IE7275/Project/Dataset/2. Electric_Vehicle_Title_and_Registration_Activity.csv')


# Remove missing values
df_EVPopulation = df_EVPopulation.dropna()
df_EVRegistration = df_EVRegistration.dropna()
```

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	Vehicle Location	Electric Utility	2020 Census Tract
0	KM8K33AGXL	King	Seattle	WA	98103.0	2020	HYUNDAI	KONA	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	258	0	43.0	249675142	POINT (-122.34301 47.659185)	CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)	5.303300e+10
1	1C4RJYB61N	King	Bothell	WA	98011.0	2022	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	1.0	233928502	POINT (-122.20578 47.762405)	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	5.303302e+10
2	1C4RJYD61P	Yakima	Yakima	WA	98908.0	2023	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	14.0	229675939	POINT (-120.6027202 46.5965625)	PACIFICORP	5.307700e+10
3	5YJ3E1EA7J	King	Kirkland	WA	98034.0	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	215	0	45.0	104714466	POINT (-122.209285 47.71124)	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	5.303302e+10
4	WBY7Z8C5XJ	Thurston	Olympia	WA	98501.0	2018	BMW	I3	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	97	0	22.0	185498386	POINT (-122.89692 47.043535)	PUGET SOUND ENERGY INC	5.306701e+10

[150137 rows x 17 columns]

Make	Model	Vehicle Primary Use	Electric Range	Odometer Reading	Odometer Code	...	Meets 2019 HB 2042 Sale Price/Value Requirement	2019 HB 2042: Battery Range Requirement	2019 HB 2042: Purchase Date Requirement	2019 HB 2042: Sale Price/Value Requirement	Electric Vehicle Fee Paid	Transportation Electrification Fee Paid	Hybrid Vehicle Electrification Fee Paid	2020 Census Tract	Legislative District	Electric Utility
TESLA	Model Y	Passenger	0	15	Actual Mileage	...	False	No battery range	Meets purchase date requirement	Sale price too high	Not Applicable	Not Applicable	Not Applicable	5.303302e+10	48.0	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
TESLA	Model Y	Passenger	0	15	Actual Mileage	...	False	No battery range	Meets purchase date requirement	Sale price too high	Not Applicable	Not Applicable	Not Applicable	5.303301e+10	43.0	CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)
PORSCHE	Taycan	Passenger	0	37	Actual Mileage	...	False	No battery range	Meets purchase date requirement	Sale price too high	Not Applicable	Not Applicable	Not Applicable	5.303302e+10	48.0	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
TESLA	Model Y	Passenger	0	15	Actual Mileage	...	False	No battery range	Meets purchase date requirement	Sale price too high	Not Applicable	Not Applicable	Not Applicable	5.303303e+10	11.0	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
KIA	Niro	Passenger	26	57920	Actual Mileage	...	False	Low battery range	Meets purchase date requirement	Meets title transaction requirement	Not Applicable	Not Applicable	Not Applicable	5.303302e+10	41.0	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)

[100386 rows x 35 columns]

Coding Review–data processing

```
# Focus on the Electric Vehicle Type distribution condition
# Data processing

# Organize the data and keep necessary columns
df_EVPopulation_2 = df_EVPopulation.loc[:, 'County': 'Electric Range']
```

✓ 0.0s

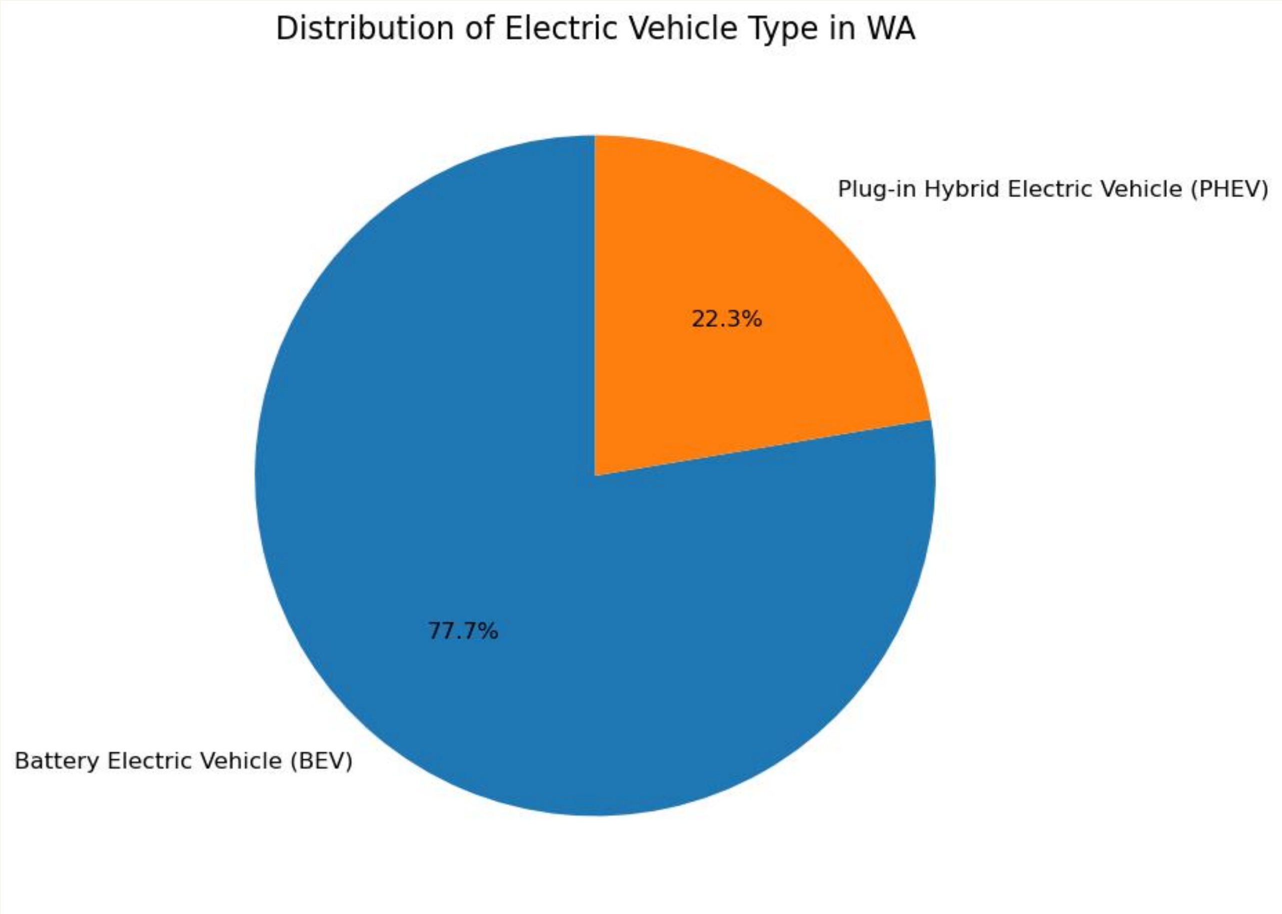
```
# Group the vehicle count by the Electric Vehicle Type
df_type = df_EVPopulation_2.groupby(['Electric Vehicle Type']).size().reset_index(name='Vehicle Count')
df_type
```

✓ 0.0s

	Electric Vehicle Type	Vehicle Count
0	Battery Electric Vehicle (BEV)	116583
1	Plug-in Hybrid Electric Vehicle (PHEV)	33554

```
# Plot the Electric Vehicle Type in pie chart
plt.figure(figsize=(12, 8))
plt.pie(df_type['Vehicle Count'], labels=df_type['Electric Vehicle Type'], autopct='%1.1f%%', startangle=90, textprops={'fontsize': 12})
plt.title('Distribution of Electric Vehicle Type in WA', fontsize=16)
plt.show()
```

Visualization & Insights



Battery Electric Vehicles (BEVs) dominates the market, 3.5 times market share as of Plug-in Hybrid Electric Vehicle (PHEV).

Coding Review–data processing

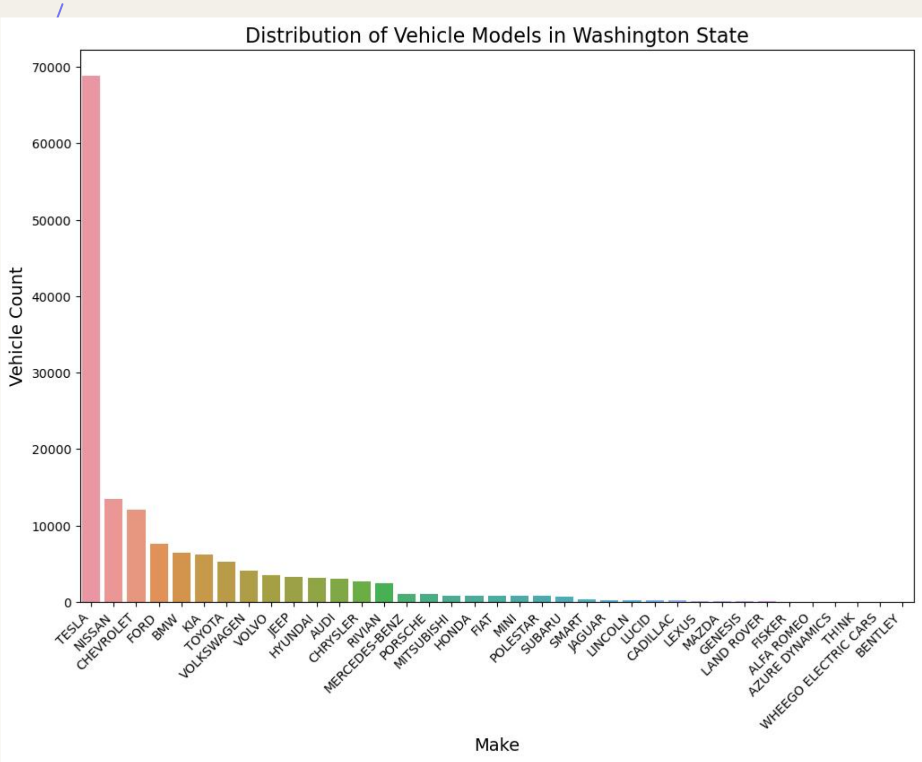
```python
# Focus on the Electric Vehicle Make's distribution
# Group the count by the Make
df_make = df_EVPopulation.groupby(['Make']).size().reset_index(name='Vehicle Count')

# Sort the DataFrame by Vehicle Count in descending order
df_make = df_make.sort_values(by='Vehicle Count', ascending=False)
df_make.head()
```

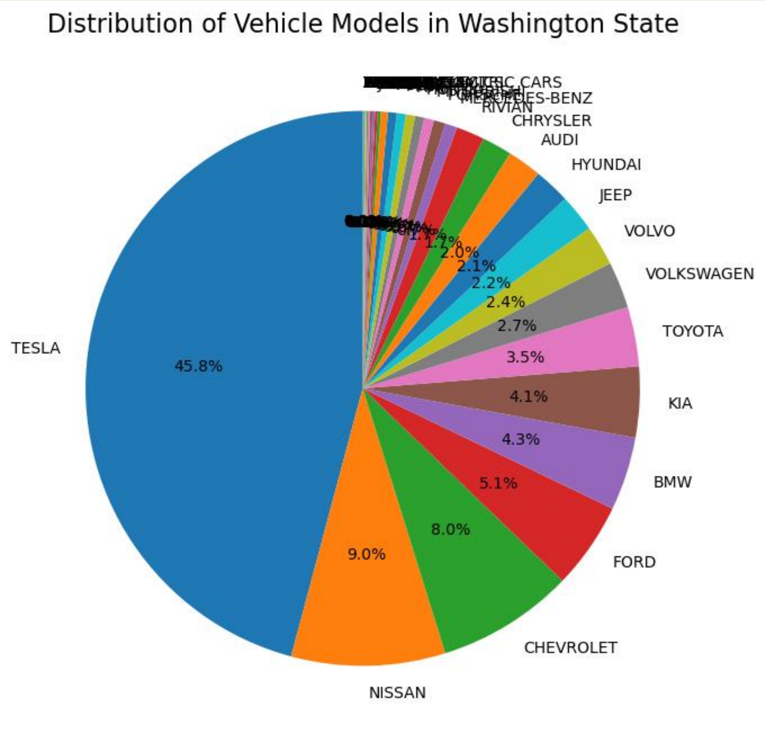
✓ 0.0s

	Make	Vehicle Count
31	TESLA	68821
25	NISSAN	13481
6	CHEVROLET	12003
10	FORD	7592
4	BMW	6426

```python
# Plot a bar chart to see the Make Distribution
plt.figure(figsize=(12, 8))
sns.barplot(x='Make', y='Vehicle Count', data=df_make, dodge=True)
plt.xlabel('Make', fontsize=14)
plt.ylabel('Vehicle Count', fontsize=14)
plt.title('Distribution of Vehicle Models in Washington State', fontsize=16)

# Adjust font size and rotation for x-axis tick labels
plt.xticks(rotation=45, ha='right', fontsize=10)

plt.show()

# Plot a pie chart for the Make Distribution
plt.figure(figsize=(12, 8))
plt.pie(df_make['Vehicle Count'], labels=df_make['Make'], autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Vehicle Models in Washington State', fontsize=16)
plt.show()
```

Visualization & Insights





- The top 5 Electric Vehicle (EV) manufacturers in Washington State include Tesla, Nissan, Chevrolet, Ford, and BMW. Tesla stands out as the market leader (45.8%).
- Traditional gasoline vehicle manufacturers such as BMW, Toyota, Mercedes-Benz, Porsche, and others are actively venturing into the EV market.

Coding Review–data processing

```
# Focus on the Electric Vehicle distribution by county and we would like to show it on the Washington Map
# Group the count by County
df_county = df_EVPopulation.groupby(['County']).size().reset_index(name='Vehicle Count')

# Sort the DataFrame by Vehicle Count in descending order
df_county = df_county.sort_values(by='Vehicle Count', ascending=False)
```

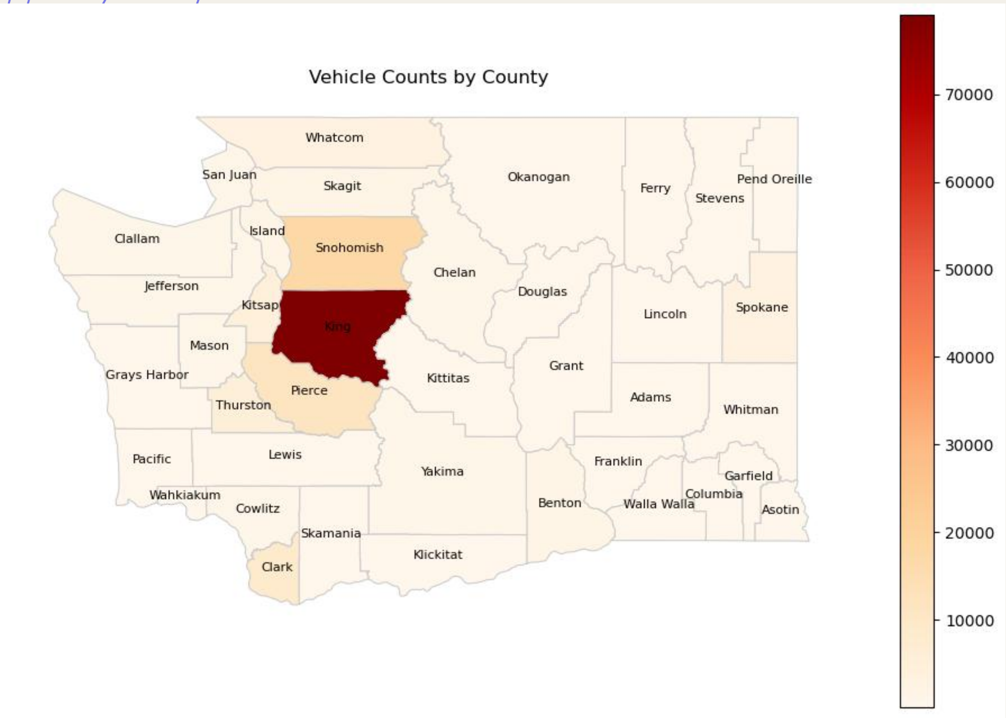
✓ 0.0s

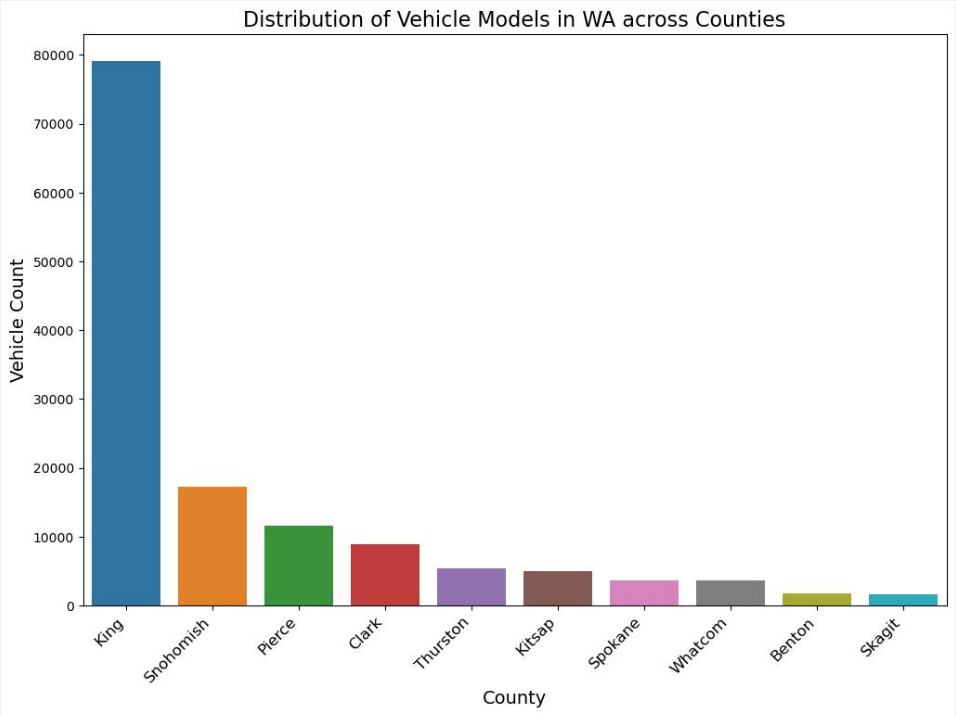
```
# import the GEOJson file of Washington State
geojason_path = '/Users/cheryl/Desktop/IE7275/Project/Dataset/WA_County_Boundaries.geojson'
wa_county = gpd.read_file(geojason_path)

# Rename columns
wa_county = wa_county.rename(columns={'JURISDICT_LABEL_NM': 'County'})
```

✓ 1.0s

```
# Merge the county dataset with the GEOJson file
df_merged = wa_county.merge(df_county, on='County', how = 'left')

# Plot the choropleth map
fig, ax = plt.subplots(1, 1, figsize=(12, 8))
df_merged.plot(column='Vehicle Count', cmap='OrRd', linewidth=0.8, ax=ax, edgecolor='0.8', legend=True)

# Annotate with county names
for x, y, label in zip(df_merged.geometry.centroid.x, df_merged.geometry.centroid.y, df_merged['County']):
    ax.text(x, y, label, fontsize=8, ha='center')

ax.set_title('Vehicle Counts by County')
ax.set_axis_off()

plt.show()

# Plot a bar chart
plt.figure(figsize=(12, 8))
sns.barplot(x='County', y='Vehicle Count', data=df_county10)
plt.xlabel('County', fontsize=14)
plt.ylabel('Vehicle Count', fontsize=14)
plt.title('Distribution of Vehicle Models in WA across Counties', fontsize=16)

# Adjust font size and rotation for x-axis tick labels
plt.xticks(rotation=45, ha='right', fontsize=12)

plt.show()
```

Visualization & Insights





The majority of Electric Vehicles are concentrated in King County, comprising nearly 80,000 units. This trend extends to its neighboring counties, Snohomish and Pierce, indicating an intense concentration of electric vehicles in specific areas within Washington State.

Coding Review–data processing

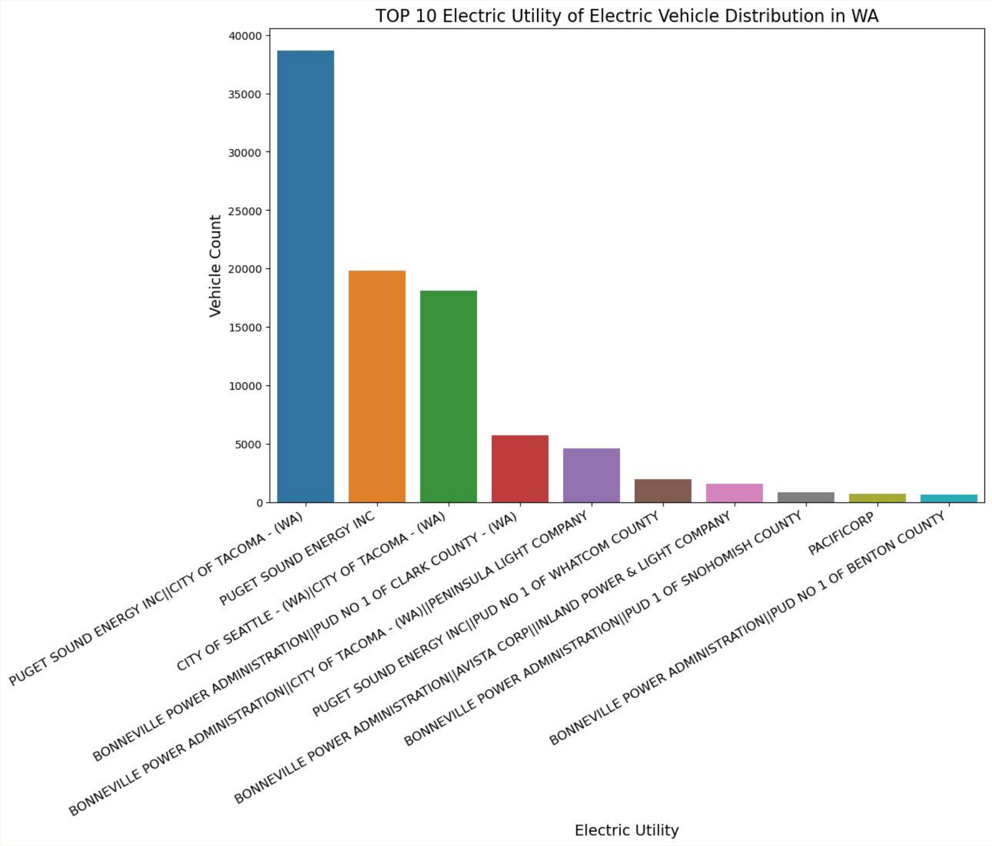
```
# Data processing
# Group the vehicle count by the Electric Utility
df_EUtility = df_EVRegistration.groupby(['Electric Utility']).size().reset_index(name='Vehicle Count')

# Sort the DataFrame by Vehicle Count in descending order
df_EUtility = df_EUtility.sort_values(by='Vehicle Count', ascending=False)

# Keep the top 10 Electric Utility Location
df_EUtility10 = df_EUtility.head(10)
df_EUtility10
```

✓ 0.0s

	Electric Utility	Vehicle Count
72	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	38672
71	PUGET SOUND ENERGY INC	19841
56	CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)	18097
35	BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF C...	5676
19	BONNEVILLE POWER ADMINISTRATION||CITY OF TACOM...	4604
73	PUGET SOUND ENERGY INC||PUD NO 1 OF WHATCOM CO...	1957
2	BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||...	1540
30	BONNEVILLE POWER ADMINISTRATION||PUD 1 OF SNOH...	830
63	PACIFICORP	707
33	BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF B...	659

```
# Plot a bar chart
plt.figure(figsize=(12, 8))
sns.barplot(x='Electric Utility', y='Vehicle Count', data=df_EUtility10)
plt.xlabel('Electric Utility', fontsize=14)
plt.ylabel('Vehicle Count', fontsize=14)
plt.title('TOP 10 Electric Utility of Electric Vehicle Distribution in WA', fontsize=16)

# Adjust font size and rotation for x-axis tick labels
plt.xticks(rotation=30, ha='right', fontsize=12)

plt.show()
```

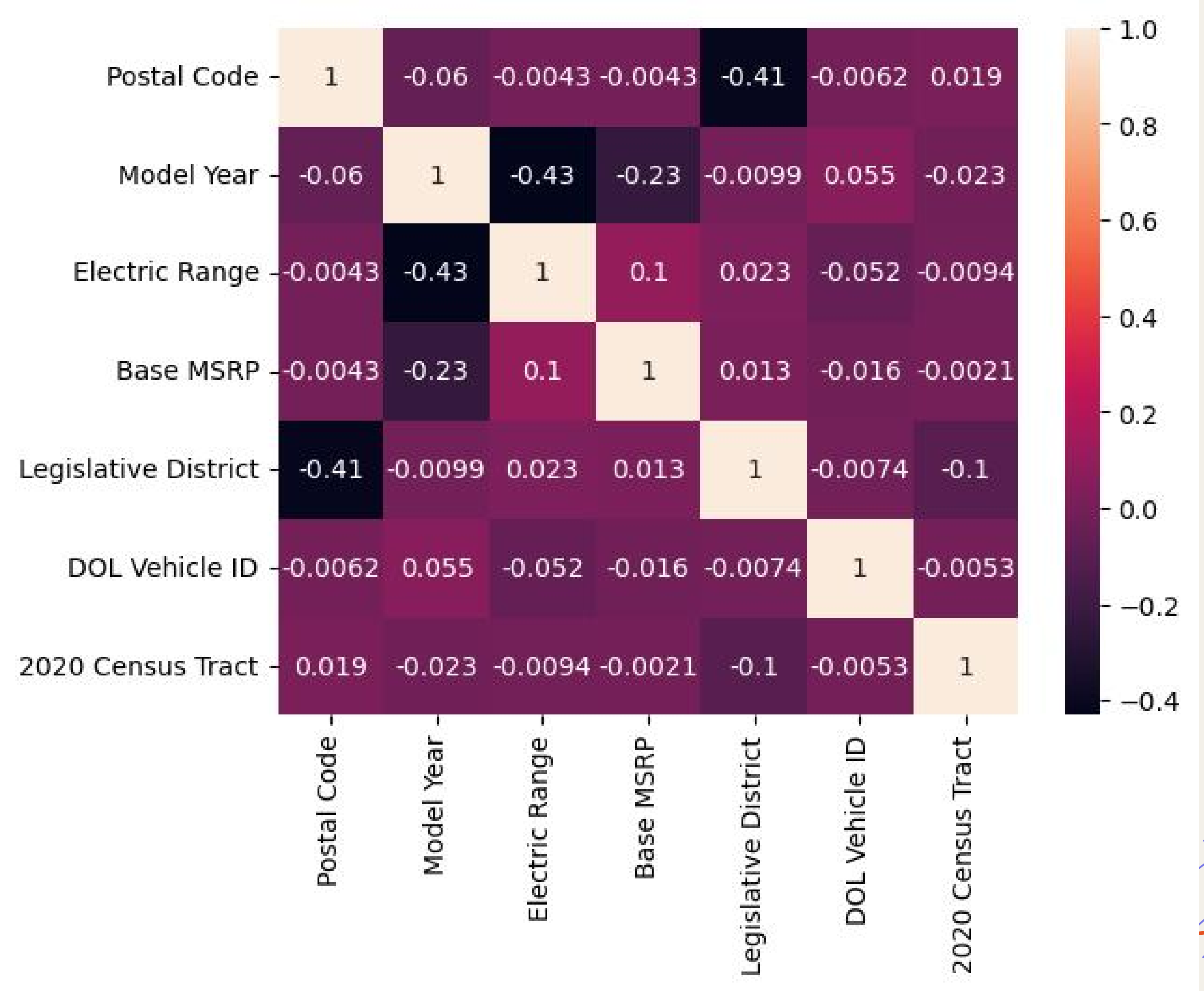
Visualization & Insights



The top three companies: Puget Sound Energy of Tacoma, Puget Sound Energy Inc, and City of Seattle. These companies are located in the three counties with the highest concentration of electric vehicles, reinforcing their significant role in the electric utility landscape.

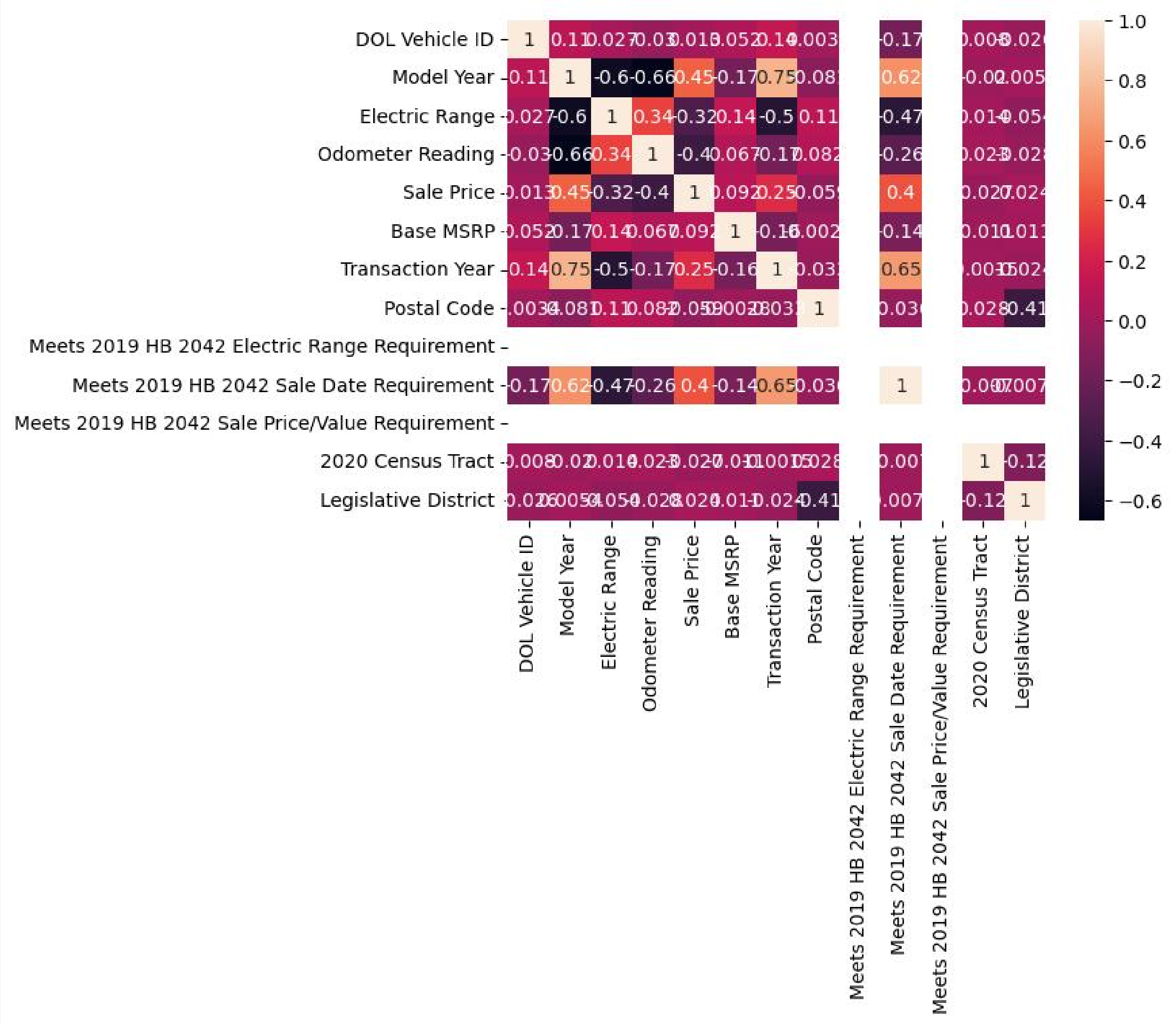
Coding Review–data correlation exploring

Heat Map



Coding Review–data correlation exploring

Heat Map



Modeling Exploration–PCA

```
# Take Odometer Reading and Model Year to apply PCA

X = df_EVRegistration[[ 'Odometer Reading']]
y = df_EVRegistration['Model Year']

# One-hot encode the categorical features 'Make'
encoder = OneHotEncoder(sparse=False, drop='first')
X_encoded = encoder.fit_transform(X)

# Encode the categorical target variable 'Model'
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

✓ 0.2s

```
/Users/cheryl/anaconda3/lib/python3.10/site-packages/sklearn/preprocessing/_encoders.py:975: FutureWarning: `sparse` was renamed to
  warnings.warn(
```

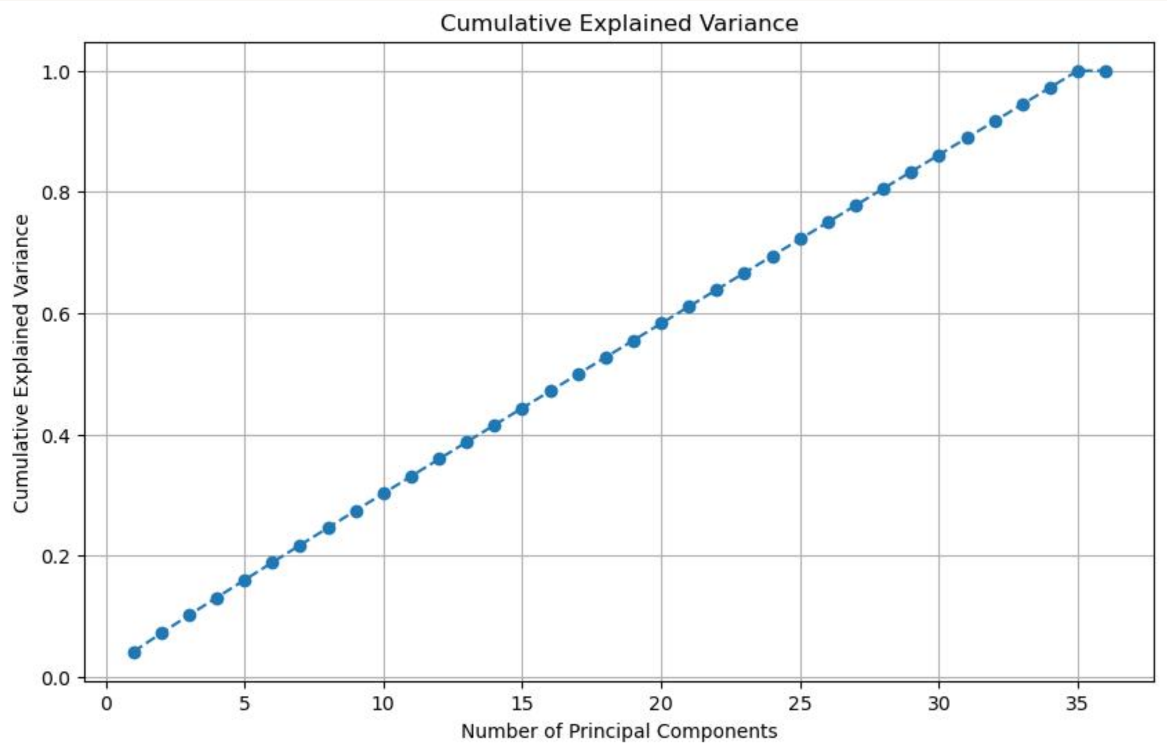
```
# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_encoded)
```

More...

✓ 35.6s

```
# plot the explained variance ratio
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_explained_variance = np.cumsum(explained_variance_ratio)

plt.figure(figsize=(10, 6))
plt.plot(range(1, len(cumulative_explained_variance) + 1), cumulative_explained_variance, marker='o', linestyle='--')
plt.title('Cumulative Explained Variance')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance')
plt.grid(True)
plt.show()
```

✓ 0.0s



Modeling Exploration–LDA

```
# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y_encoded, test_size=0.2, random_state=42)

# Standardize or normalize the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Create a Linear Discriminant Analysis classifier
lda_classifier = LinearDiscriminantAnalysis()

# Train the model on the scaled training set
lda_classifier.fit(X_train_scaled, y_train)
```

✓ 0.2s

```
▾ LinearDiscriminantAnalysis
LinearDiscriminantAnalysis()
```

```
# Evaluate the model on the scaled test set
accuracy = lda_classifier.score(X_test_scaled, y_test)
print(f"Accuracy: {accuracy}")
```

✓ 0.0s

```
Accuracy: 0.13810443585986412
```

Analysis

- **Complex Relationships:** Linear methods like PCA and LDA are effective when the relationships in the data are linear. If the data has complex, nonlinear relationships, PCA and LDA may not effectively capture the variation.

- **Data Not Well-Suited for Linear Techniques:** If the data has complex patterns or relationships that cannot be adequately captured by linear techniques, PCA and LDA may not be the most suitable method.

Modeling Exploration

```
# Keep the top 10 Makes
top_makes = df_EVPopulation['Make'].value_counts().index[:10]
df_top = df_EVPopulation[df_EVPopulation['Make'].isin(top_makes)]

X = df_EVPopulation[[ 'Make']]
y = df_EVPopulation['Model']

# One-hot encode the categorical features
encoder = OneHotEncoder(sparse=False, drop='first')
X_encoded = encoder.fit_transform(X)

# Encode the categorical target variable 'Model'
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y_encoded, test_size=0.2, random_state=42)
```

✓ 0.0s

Modeling Exploration– Decision Tree Classifier

```python
# Create a Classifier for multiclass classification
clf = RandomForestClassifier(random_state=42)

# Train the model on the training set
clf.fit(X_train, y_train)

# Extract an individual tree from the forest (for example, the first tree)
individual_tree = clf.estimators_[0]

# Debugging: Check the actual feature names used for training
feature_names = encoder.get_feature_names_out(['Make'])
print("Feature names:", feature_names)

# Display the individual Decision Tree
plt.figure(figsize=(15, 10))
plot_tree(individual_tree, filled=True, feature_names=feature_names,
          class_names=label_encoder.classes_, rounded=True, fontsize=10)
plt.show()

# Evaluate the model on both of training set and test set
accuracy_training = clf.score(X_train, y_train)
accuracy_test = clf.score(X_test, y_test)
print(f"Accuracy of training set: {accuracy_training}")
print(f"Accuracy of testing set: {accuracy_test}")

# Make predictions on the test set
y_pred = clf.predict(X_test)

# Print classification report and confusion matrix
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy of training set: 0.5127758952285008
Accuracy of testing set: 0.5150859198081791
```

Modeling Exploration– kNN

```
# Create a k-Nearest Neighbors Classifier for multiclass classification
knn = KNeighborsClassifier()

# Train the model on the training set
knn.fit(X_train, y_train)

# Evaluate the model on both of training set and test set
accuracy_training = knn.score(X_train, y_train)
accuracy_test = knn.score(X_test, y_test)
print(f"Accuracy of training set: {accuracy_training}")
print(f"Accuracy of testing set: {accuracy_test}")

# Make predictions on the test set
y_pred = knn.predict(X_test)

# Print classification report and confusion matrix
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy of training set: 0.44835541430540665
Accuracy of testing set: 0.44758002756930615
```

Analysis

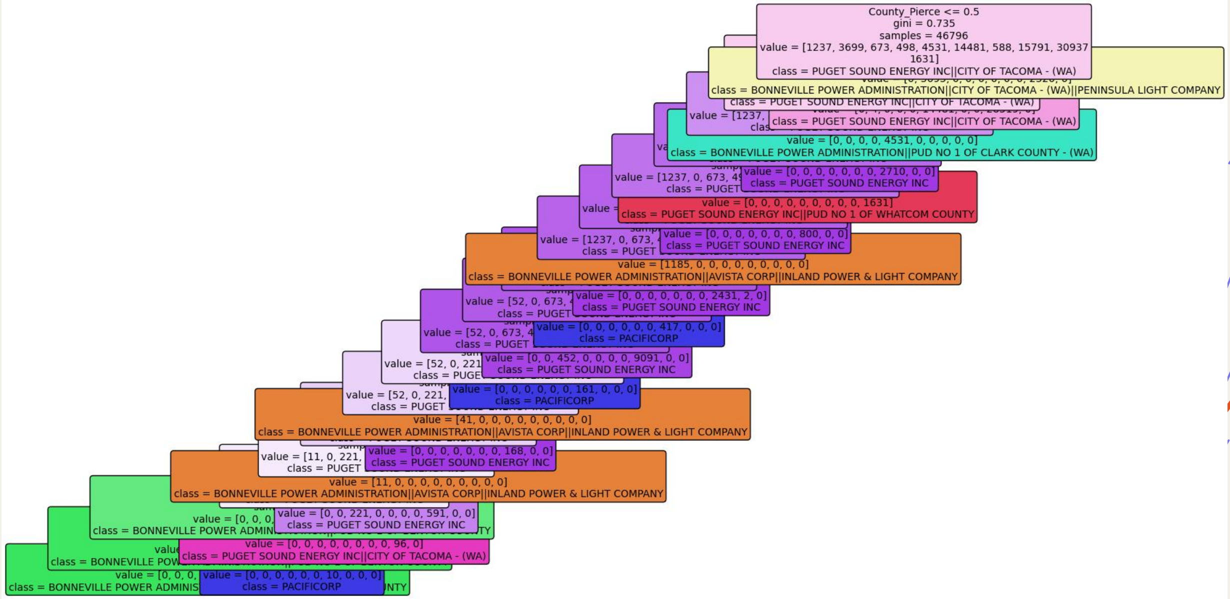
- **Imbalanced Classes:** If the classes in dataset are imbalanced, the models may have difficulty learning patterns in the minority class.
- **Nonlinearity in Data:** Decision trees and kNN models assume certain relationships between features and the target variable. If the relationships are highly nonlinear, these models might struggle to capture them effectively.

Modeling Exploration–Decision Tree Classifier

```python
# Keep the top 10 Utilities
top_Utility = df_EVRegistration['Electric Utility'].value_counts().index[:10]
df_topUtility = df_EVRegistration[df_EVRegistration['Electric Utility'].isin(top_Utility)]
```

✓ 0.0s

```python
X = df_topUtility[['County']]
y = df_topUtility['Electric Utility']

# One-hot encode the categorical features
encoder = OneHotEncoder(sparse=False, drop='first')
X_encoded = encoder.fit_transform(X)

# Encode the categorical target variable 'Model'
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y_encoded, test_size=0.2, random_state=42)

# Create a Random Forest Classifier for multiclass classification
clf = RandomForestClassifier(random_state=42)

# Train the model on the training set
clf.fit(X_train, y_train)

# Extract an individual tree from the forest (for example, the first tree)
individual_tree = clf.estimators_[0]

# Display the individual Decision Tree
plt.figure(figsize=(15, 10))
plot_tree(individual_tree, filled=True, feature_names=encoder.get_feature_names_out(['County']),
          class_names=label_encoder.classes_, rounded=True, fontsize=10)
plt.show()
```



Modeling Exploration–Decision Tree Classifier

```python
# Evaluate the model on both of training set and test set
accuracy_training = clf.score(X_train, y_train)
accuracy_test = clf.score(X_test, y_test)
print(f"Accuracy of training set: {accuracy_training}")
print(f"Accuracy of testing set: {accuracy_test}")

# Make predictions on the test set
y_pred = clf.predict(X_test)

# Print classification report and confusion matrix
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

✓ 0.4s

```
Accuracy of training set: 0.7634407150379391
Accuracy of testing set: 0.7627585462007884
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       308
           1       0.60      0.99      0.75       901
           2       0.00      0.00      0.00       154
           3       0.99      1.00      1.00       111
           4       1.00      1.00      1.00      1140
           5       0.00      0.00      0.00      3635
           6       1.00      0.99      1.00       145
           7       0.96      1.00      0.98      3966
           8       0.66      0.92      0.77      7791
           9       1.00      1.00      1.00       366

    accuracy                           0.76     18517
   macro avg       0.72      0.79      0.75     18517
weighted avg       0.63      0.76      0.68     18517
```

Conclusion

1. Limitations and restrictions faced with the dataset used.

- Most columns are categorical, so has limitation in the application of numerical modeling like Lenear and Multiple Regression.
- There is limitations in applying to PCA and LDA models, the possible reason is they assume linear relationships between variables. If the underlying relationships in data are nonlinear, PCA and LDA may not effectively capture the variation.
- If the data has complex patterns or relationships that cannot be adequately captured by linear techniques, PCA and LDA may not be the most suitable method.

2. Insights gathered from the data and which features were preferred over others

Key Findings

- The amount of Battery Electric Vehicles are **3.5 times** as that of Plug-in Hybrid Electric Vehicles.
- **Tesla dominates the Electric Vehicle market share (45.8%) in WA**, but most of the traditional gasoline vehicle manufacturers have been actively venturing into this market.
- There is an intense concentration of electric vehicles in **King County** and its neighborers **Snohomish and Pierce**, and the top three Electric Utility companies also locate around these areas.
- Model performance depends on the dataset type and also the factor selection.

Conclusion

3. Data Mining techniques used and why they were used?

PCA, LDA, DecisionTree and kNN data mining techniques have been used. The datasets are large so tried to used PCA and LDA for the dimentionality reduction. And they are more of categorical datasets, so applied Decision Tree and kNN for the classification.

4. Other information

Future Development Suggestions

- Battery Electric Vehicles manufacturers like Tesla can focus more on the **after-sales service** such as battery replacement, upgrade its digital system to maintain the customer relationship.
- Plug-in Hybrid Electric Vehicles manufacturers should pay more attention to the **technology innovation** in order to attract more customers.
- Charging infrastructure and the utility distribution can be more centered around the King county to facilitate the citizens vehicle charging.

Reference

1. Anon. 2023. “Electric Vehicle Population Data | Data.WA | State of Washington.” Retrieved (https://data.wa.gov/Transportation/Electric-Vehicle-Population-Data/f6w7-q2d2)
2. Anon. n.d.-b. “Electric Vehicle Registration Activity by Year | Data.WA | State of Washington.” Retrieved (https://data.wa.gov/Transportation/Electric-Vehicle-Registration-Activity-by-Year/tak8-xdcp).
3. Anon. n.d.-c. “How Can One Interpret a Heat Map Plot.” Cross Validated. Retrieved (https://stats.stackexchange.com/questions/392517/how-can-one-interpret-a-heat-map-plo).
4. Anon. n.d.-d. “Washington State Geospatial Open Data Portal.” Retrieved (https://geo.wa.gov/datasets/a77ee02fab3e4793b274ec52b7a523d8/explore).

Thank You!