Neural Network Modeling

This notebook contains the logic for neural network regression and classification tasks.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
!ls
```

```
drive  sample_data
```

```
cleaned = pd.read_csv("/content/drive/Shareddrives/Data Science 303 Group Project/csv/cleaned_fires_data_with_four_closest_stations_nov21.csv")
cleaned = cleaned.replace([np.inf, -np.inf], np.nan)
cleaned = cleaned.dropna()
```

```
display(cleaned.head())
display(cleaned.info(verbose=True))

STATION_LIST = ["BODIE", "BROOKS", "COHASSET", "EEL_RIVER", "HELL_HOLE", "HERNANDEZ"]
STATION_LIST += ["HUNTER_MOUNTAIN", "JUANITA_LAKE", "LADDER_BUTTE", "LAS_TABLAS", "LA_HONDA", "OAK_CREEK"]
STATION_LIST += ["PANAMINT", "PILOT_HILL", "SCORPION", "SOLDIER_MOUNTAIN", "SQUAW_LAKE", "STAMPEDE", "VAN_BREMMER", "WOLVERTON"]
STATION_LIST = set(STATION_LIST)
```

	Unnamed: 0	BODIE	BROOKS	COHASSET	CONTAINMENT_MONTH	CONT_DATE	CONT_DOY	CONT_TIME	DISCOVERY_DATE	DISCOVERY_DOY	DISCOVERY_MONTH	DISCOVERY_TIME	EEL_RIVER	FIRE_SIZE	FIRE_SIZE_CLASS	FIRE_YEAR	FOD_ID	HELL_HOLE	HERNANDEZ	HUNTER_MOUNTAIN	JUANITA_LAKE	LADDER_BUTTE	LAS_TABLAS	LATITUDE	LA_HONDA	LONGITUDE	NWCG_RE
0	0	-0.501752	-0.841998	-1.128433	2	2453403.5	33.0	1730.0	2453403.5	33	2	1300.0	-0.848324	0.10	A	2005	1	-1.000659	0.195610	0.434939	-0.886032	-1.114178	0.448456	40.036944	-0.394781	-121.005833	
1	1	-1.237103	-0.939629	-0.802799	5	2453137.5	133.0	1530.0	2453137.5	133	5	845.0	-0.597768	0.25	A	2004	2	-1.471873	-0.562875	-0.243079	-0.487813	-0.669480	-0.281374	38.933056	-0.869095	-120.404444	
2	2	-1.070104	-1.059637	-0.891098	6	2453156.5	152.0	2024.0	2453156.5	152	6	1921.0	-0.695749	0.10	A	2004	3	-1.410920	-0.543260	-0.109734	-0.533929	-0.714976	-0.262071	38.984167	-0.950563	-120.735556	
3	3	-1.588084	-0.752356	-0.586044	7	2453189.5	185.0	1400.0	2453184.5	180	6	1600.0	-0.401895	0.10	A	2004	4	-1.213089	-0.755069	-0.564467	-0.322165	-0.486222	-0.477673	38.559167	-0.836997	-119.913333	
4	4	-1.578681	-0.759860	-0.590604	7	2453189.5	185.0	1200.0	2453184.5	180	6	1600.0	-0.407106	0.10	A	2004	5	-1.217813	-0.758604	-0.557924	-0.324379	-0.488626	-0.480494	38.559167	-0.845502	-119.933056	

5 rows × 301 columns

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 87106 entries, 0 to 87105
Data columns (total 301 columns):
 #   Column                                                                Dtype
---  ------                                                                -----
 0   Unnamed: 0                                                            int64
 1   BODIE                                                                 float64
 2   BROOKS                                                                float64
 3   COHASSET                                                              float64
 4   CONTAINMENT_MONTH                                                     int64
 5   CONT_DATE                                                             float64
 6   CONT_DOY                                                              float64
 7   CONT_TIME                                                             float64
 8   DISCOVERY_DATE                                                        float64
 9   DISCOVERY_DOY                                                         int64
 10  DISCOVERY_MONTH                                                       int64
 11  DISCOVERY_TIME                                                        float64
 12  EEL_RIVER                                                             float64
 13  FIRE_SIZE                                                             float64
 14  FIRE_SIZE_CLASS                                                       object
 15  FIRE_YEAR                                                             int64
 16  FOD_ID                                                                int64
 17  HELL_HOLE                                                             float64
 18  HERNANDEZ                                                             float64
 19  HUNTER_MOUNTAIN                                                       float64
 20  JUANITA_LAKE                                                          float64
 21  LADDER_BUTTE                                                          float64
 22  LAS_TABLAS                                                            float64
 23  LATITUDE                                                              float64
 24  LA_HONDA                                                              float64
 25  LONGITUDE                                                             float64
 26  NWCG_REPORTING_AGENCY                                                 object
 27  NWCG_REPORTING_UNIT_NAME                                              object
 28  OAK_CREEK                                                             float64
 29  OWNER_DESCR                                                           object
 30  PANAMINT                                                              float64
 31  PILOT_HILL                                                            float64
 32  PRIMARY_STATION_10_MONTHS_PRIOR_ELEVATION                             float64
 33  PRIMARY_STATION_10_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH     float64
 34  PRIMARY_STATION_10_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH float64
 35  PRIMARY_STATION_10_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE           float64
 36  PRIMARY_STATION_10_MONTHS_PRIOR_LATITUDE                              float64
 37  PRIMARY_STATION_10_MONTHS_PRIOR_LONGITUDE                             float64
 38  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                float64
 39  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE     float64
 40  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT float64
 41  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT float64
 42  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT float64
 43  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT float64
 44  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT float64
 45  PRIMARY_STATION_10_MONTHS_PRIOR_TEMPERATURE_AVERAGE                   float64
 46  PRIMARY_STATION_10_MONTHS_PRIOR_TEMPERATURE_MAX                       float64
 47  PRIMARY_STATION_10_MONTHS_PRIOR_TEMPERATURE_MIN                       float64
 48  PRIMARY_STATION_11_MONTHS_PRIOR_ELEVATION                             float64
 49  PRIMARY_STATION_11_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH     float64
 50  PRIMARY_STATION_11_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH float64
 51  PRIMARY_STATION_11_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE           float64
 52  PRIMARY_STATION_11_MONTHS_PRIOR_LATITUDE                              float64
 53  PRIMARY_STATION_11_MONTHS_PRIOR_LONGITUDE                             float64
 54  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                float64
 55  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE     float64
 56  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT float64
 57  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT float64
 58  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT float64
 59  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT float64
 60  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT float64
 61  PRIMARY_STATION_11_MONTHS_PRIOR_TEMPERATURE_AVERAGE                   float64
 62  PRIMARY_STATION_11_MONTHS_PRIOR_TEMPERATURE_MAX                       float64
 63  PRIMARY_STATION_11_MONTHS_PRIOR_TEMPERATURE_MIN                       float64
 64  PRIMARY_STATION_12_MONTHS_PRIOR_ELEVATION                             float64
 65  PRIMARY_STATION_12_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH     float64
 66  PRIMARY_STATION_12_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH float64
 67  PRIMARY_STATION_12_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE           float64
 68  PRIMARY_STATION_12_MONTHS_PRIOR_LATITUDE                              float64
 69  PRIMARY_STATION_12_MONTHS_PRIOR_LONGITUDE                             float64
 70  PRIMARY_STATION_12_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                float64
 71  PRIMARY_STATION_12_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE     float64
 72  PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT float64
 73  PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT float64
 74  PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT float64
 75  PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT float64
 76  PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT float64
 77  PRIMARY_STATION_12_MONTHS_PRIOR_TEMPERATURE_AVERAGE                   float64
 78  PRIMARY_STATION_12_MONTHS_PRIOR_TEMPERATURE_MAX                       float64
 79  PRIMARY_STATION_12_MONTHS_PRIOR_TEMPERATURE_MIN                       float64
 80  PRIMARY_STATION_13_MONTHS_PRIOR_ELEVATION                             float64
 81  PRIMARY_STATION_13_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH     float64
 82  PRIMARY_STATION_13_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH float64
 83  PRIMARY_STATION_13_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE           float64
 84  PRIMARY_STATION_13_MONTHS_PRIOR_LATITUDE                              float64
 85  PRIMARY_STATION_13_MONTHS_PRIOR_LONGITUDE                             float64
 86  PRIMARY_STATION_13_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                float64
 87  PRIMARY_STATION_13_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE     float64
 88  PRIMARY_STATION_13_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT float64
 89  PRIMARY_STATION_13_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT float64
 90  PRIMARY_STATION_13_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT float64
 91  PRIMARY_STATION_13_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT float64
 92  PRIMARY_STATION_13_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT float64
 93  PRIMARY_STATION_13_MONTHS_PRIOR_TEMPERATURE_AVERAGE                   float64
 94  PRIMARY_STATION_13_MONTHS_PRIOR_TEMPERATURE_MAX                       float64
 95  PRIMARY_STATION_13_MONTHS_PRIOR_TEMPERATURE_MIN                       float64
 96  PRIMARY_STATION_14_MONTHS_PRIOR_ELEVATION                             float64
 97  PRIMARY_STATION_14_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH     float64
 98  PRIMARY_STATION_14_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH float64
 99  PRIMARY_STATION_14_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE           float64
 100 PRIMARY_STATION_14_MONTHS_PRIOR_LATITUDE                              float64
 101 PRIMARY_STATION_14_MONTHS_PRIOR_LONGITUDE                             float64
```

```
102 PRIMARY_STATION_14_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                      float64
103 PRIMARY_STATION_14_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE           float64
104 PRIMARY_STATION_14_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT float64
105 PRIMARY_STATION_14_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT  float64
106 PRIMARY_STATION_14_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT  float64
107 PRIMARY_STATION_14_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT   float64
108 PRIMARY_STATION_14_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT  float64
109 PRIMARY_STATION_14_MONTHS_PRIOR_TEMPERATURE_AVERAGE                         float64
110 PRIMARY_STATION_14_MONTHS_PRIOR_TEMPERATURE_MAX                             float64
111 PRIMARY_STATION_14_MONTHS_PRIOR_TEMPERATURE_MIN                             float64
112 PRIMARY_STATION_15_MONTHS_PRIOR_ELEVATION                                   float64
113 PRIMARY_STATION_15_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH           float64
114 PRIMARY_STATION_15_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH       float64
115 PRIMARY_STATION_15_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE                 float64
116 PRIMARY_STATION_15_MONTHS_PRIOR_LATITUDE                                    float64
117 PRIMARY_STATION_15_MONTHS_PRIOR_LONGITUDE                                   float64
118 PRIMARY_STATION_15_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                      float64
119 PRIMARY_STATION_15_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE           float64
120 PRIMARY_STATION_15_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT float64
121 PRIMARY_STATION_15_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT  float64
122 PRIMARY_STATION_15_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT  float64
123 PRIMARY_STATION_15_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT   float64
124 PRIMARY_STATION_15_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT  float64
125 PRIMARY_STATION_15_MONTHS_PRIOR_TEMPERATURE_AVERAGE                         float64
126 PRIMARY_STATION_15_MONTHS_PRIOR_TEMPERATURE_MAX                             float64
127 PRIMARY_STATION_15_MONTHS_PRIOR_TEMPERATURE_MIN                             float64
128 PRIMARY_STATION_16_MONTHS_PRIOR_ELEVATION                                   float64
129 PRIMARY_STATION_16_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH           float64
130 PRIMARY_STATION_16_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH       float64
131 PRIMARY_STATION_16_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE                 float64
132 PRIMARY_STATION_16_MONTHS_PRIOR_LATITUDE                                    float64
133 PRIMARY_STATION_16_MONTHS_PRIOR_LONGITUDE                                   float64
134 PRIMARY_STATION_16_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                      float64
135 PRIMARY_STATION_16_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE           float64
136 PRIMARY_STATION_16_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT float64
137 PRIMARY_STATION_16_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT  float64
138 PRIMARY_STATION_16_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT  float64
139 PRIMARY_STATION_16_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT   float64
140 PRIMARY_STATION_16_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT  float64
141 PRIMARY_STATION_16_MONTHS_PRIOR_TEMPERATURE_AVERAGE                         float64
142 PRIMARY_STATION_16_MONTHS_PRIOR_TEMPERATURE_MAX                             float64
143 PRIMARY_STATION_16_MONTHS_PRIOR_TEMPERATURE_MIN                             float64
144 PRIMARY_STATION_1_MONTHS_PRIOR_ELEVATION                                    float64
145 PRIMARY_STATION_1_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH            float64
146 PRIMARY_STATION_1_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH        float64
147 PRIMARY_STATION_1_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE                  float64
148 PRIMARY_STATION_1_MONTHS_PRIOR_LATITUDE                                     float64
149 PRIMARY_STATION_1_MONTHS_PRIOR_LONGITUDE                                    float64
150 PRIMARY_STATION_1_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                       float64
151 PRIMARY_STATION_1_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE            float64
152 PRIMARY_STATION_1_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT  float64
153 PRIMARY_STATION_1_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT   float64
154 PRIMARY_STATION_1_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT   float64
155 PRIMARY_STATION_1_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT    float64
156 PRIMARY_STATION_1_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT   float64
157 PRIMARY_STATION_1_MONTHS_PRIOR_TEMPERATURE_AVERAGE                          float64
158 PRIMARY_STATION_1_MONTHS_PRIOR_TEMPERATURE_MAX                              float64
159 PRIMARY_STATION_1_MONTHS_PRIOR_TEMPERATURE_MIN                              float64
160 PRIMARY_STATION_2_MONTHS_PRIOR_ELEVATION                                    float64
161 PRIMARY_STATION_2_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH            float64
162 PRIMARY_STATION_2_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH        float64
163 PRIMARY_STATION_2_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE                  float64
164 PRIMARY_STATION_2_MONTHS_PRIOR_LATITUDE                                     float64
165 PRIMARY_STATION_2_MONTHS_PRIOR_LONGITUDE                                    float64
166 PRIMARY_STATION_2_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                       float64
167 PRIMARY_STATION_2_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE            float64
168 PRIMARY_STATION_2_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT  float64
169 PRIMARY_STATION_2_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT   float64
170 PRIMARY_STATION_2_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT   float64
171 PRIMARY_STATION_2_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT    float64
172 PRIMARY_STATION_2_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT   float64
173 PRIMARY_STATION_2_MONTHS_PRIOR_TEMPERATURE_AVERAGE                          float64
174 PRIMARY_STATION_2_MONTHS_PRIOR_TEMPERATURE_MAX                              float64
175 PRIMARY_STATION_2_MONTHS_PRIOR_TEMPERATURE_MIN                              float64
176 PRIMARY_STATION_3_MONTHS_PRIOR_ELEVATION                                    float64
177 PRIMARY_STATION_3_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH            float64
178 PRIMARY_STATION_3_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH        float64
179 PRIMARY_STATION_3_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE                  float64
180 PRIMARY_STATION_3_MONTHS_PRIOR_LATITUDE                                     float64
181 PRIMARY_STATION_3_MONTHS_PRIOR_LONGITUDE                                    float64
182 PRIMARY_STATION_3_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                       float64
183 PRIMARY_STATION_3_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE            float64
184 PRIMARY_STATION_3_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT  float64
185 PRIMARY_STATION_3_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT   float64
186 PRIMARY_STATION_3_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT   float64
187 PRIMARY_STATION_3_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT    float64
188 PRIMARY_STATION_3_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT   float64
189 PRIMARY_STATION_3_MONTHS_PRIOR_TEMPERATURE_AVERAGE                          float64
190 PRIMARY_STATION_3_MONTHS_PRIOR_TEMPERATURE_MAX                              float64
191 PRIMARY_STATION_3_MONTHS_PRIOR_TEMPERATURE_MIN                              float64
192 PRIMARY_STATION_4_MONTHS_PRIOR_ELEVATION                                    float64
193 PRIMARY_STATION_4_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH            float64
194 PRIMARY_STATION_4_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH        float64
195 PRIMARY_STATION_4_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE                  float64
196 PRIMARY_STATION_4_MONTHS_PRIOR_LATITUDE                                     float64
197 PRIMARY_STATION_4_MONTHS_PRIOR_LONGITUDE                                    float64
198 PRIMARY_STATION_4_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                       float64
199 PRIMARY_STATION_4_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE            float64
200 PRIMARY_STATION_4_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT  float64
201 PRIMARY_STATION_4_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT   float64
202 PRIMARY_STATION_4_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT   float64
203 PRIMARY_STATION_4_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT    float64
204 PRIMARY_STATION_4_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT   float64
205 PRIMARY_STATION_4_MONTHS_PRIOR_TEMPERATURE_AVERAGE                          float64
206 PRIMARY_STATION_4_MONTHS_PRIOR_TEMPERATURE_MAX                              float64
207 PRIMARY_STATION_4_MONTHS_PRIOR_TEMPERATURE_MIN                              float64
208 PRIMARY_STATION_5_MONTHS_PRIOR_ELEVATION                                    float64
209 PRIMARY_STATION_5_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH            float64
210 PRIMARY_STATION_5_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH        float64
211 PRIMARY_STATION_5_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE                  float64
212 PRIMARY_STATION_5_MONTHS_PRIOR_LATITUDE                                     float64
213 PRIMARY_STATION_5_MONTHS_PRIOR_LONGITUDE                                    float64
214 PRIMARY_STATION_5_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                       float64
215 PRIMARY_STATION_5_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE            float64
216 PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT  float64
217 PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT   float64
218 PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT   float64
219 PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT    float64
220 PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT   float64
221 PRIMARY_STATION_5_MONTHS_PRIOR_TEMPERATURE_AVERAGE                          float64
```

Final Part of Normalization: MinMax scaling latitude and longitude

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()

lat = scaler.fit_transform(np.array(cleaned["LATITUDE"]).reshape(-1, 1))
long = scaler.fit_transform(np.array(cleaned["LONGITUDE"]).reshape(-1, 1))

# DROP ALL LATITUDE LONGITUDE COLUMNS, THEN READD THE SCALED LATITUDE AND LONGITUDE
drops = []
for col in cleaned.columns:
  if "LATITUDE" in col or "LONGITUDE" in col:
    drops.append(col)

for col in drops:
  del cleaned[col]

cleaned["S_LATITUDE"] = lat
cleaned["S_LONGITUDE"] = long

```

```
246 PRIMARY_STATION_7_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                        float64
```

```
# Elevation for the primary station is redundant. Drop all elevation columns from the primary station, except for one
elevation = cleaned["PRIMARY_STATION_1_MONTHS_PRIOR_ELEVATION"]

drops = []
for col in cleaned.columns:
  if "PRIMARY_STATION" in col and "ELEVATION" in col:
    drops.append(col)

for col in drops:
  del cleaned[col]

cleaned["PRIMARY_STATION_ELEVATION"] = elevation

```

```
262 PRIMARY_STATION_8_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                        float64
```

```
# We should have a large number of fires. If not, fail
print(len(cleaned))
assert(len(cleaned) > 70000)
```

```
87106
```

```
269 PRIMARY_STATION_8_MONTHS_PRIOR_TEMPERATURE_AVERAGE                          float64
```

```
print( (1 - 360) % 365)
```

```
6
```

```
274 PRIMARY_STATION_9_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH        float64
```

```
cleaned['DURATION'] = (cleaned['CONT_DOY']-cleaned['DISCOVERY_DOY'])%365
```

```
277 PRIMARY_STATION_9_MONTHS_PRIOR_LONGITUDE                                    float64
```

```
cleaned['DURATION'].max()
```

```
335.0
```

```
282 PRIMARY_STATION_9_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT   float64
```

```
from sklearn.linear_model import LinearRegression, Ridge
```

```
285 PRIMARY_STATION_9_MONTHS_PRIOR_TEMPERATURE_AVERAGE                          float64
```

```
# Working off of closest_station_1 for now, we can replicate later with other stations if we want
# 1. Get number of days that the fire was going (containment date - discovery date)
# Model 1a: fire duration
```

```
290 SOURCE_REPORTING_UNIT_NAME                                                 object
```

```
numerical = cleaned.select_dtypes(include="number")
numerical.head()
numerical.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 87106 entries, 0 to 87105
Data columns (total 246 columns):
 #   Column                                                                  Dtype
---  ------                                                                  -----
 0   Unnamed: 0                                                              int64
 1   BODIE                                                                   float64
 2   BROOKS                                                                  float64
 3   CONASSET                                                                float64
 4   CONTAINMENT_MONTH                                                       int64
 5   CONT_DATE                                                               float64
 6   CONT_DOY                                                                float64
 7   CONT_TIME                                                               float64
 8   DISCOVERY_DATE                                                          float64
 9   DISCOVERY_DOY                                                           int64
 10  DISCOVERY_MONTH                                                         int64
 11  DISCOVERY_TIME                                                          float64
 12  EEL_RIVER                                                               float64
 13  FIRE_SIZE                                                               float64
 14  FIRE_YEAR                                                               int64
 15  FOD_ID                                                                  int64
 16  HELL_HOLE                                                               float64
 17  HERNANDEZ                                                               float64
 18  HUNTER_MOUNTAIN                                                         float64
 19  JUANITA_LAKE                                                            float64
 20  LADDER_BUTTE                                                            float64
 21  LAS_TABLAS                                                              float64
 22  LA_HONDA                                                                float64
 23  OAK_CREEK                                                               float64
 24  PANAMINT                                                                float64
 25  PILOT_HILL                                                              float64
 26  PRIMARY_STATION_10_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH       float64
 27  PRIMARY_STATION_10_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH   float64
 28  PRIMARY_STATION_10_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE             float64
 29  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                  float64
 30  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE       float64
 31  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT  float64
 32  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT  float64
 33  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT  float64
 34  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT   float64
 35  PRIMARY_STATION_10_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT  float64
 36  PRIMARY_STATION_10_MONTHS_PRIOR_TEMPERATURE_AVERAGE                     float64
 37  PRIMARY_STATION_10_MONTHS_PRIOR_TEMPERATURE_MAX                         float64
 38  PRIMARY_STATION_10_MONTHS_PRIOR_TEMPERATURE_MIN                         float64
 39  PRIMARY_STATION_11_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH       float64
 40  PRIMARY_STATION_11_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH   float64
 41  PRIMARY_STATION_11_MONTHS_PRIOR_HEATING_DEGREE_DAYS_TO_DATE             float64
 42  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY                  float64
 43  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_COOLING_DEGREE_DAY_CUMULATIVE       float64
 44  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT  float64
```

```
45  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT   float64
46  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT   float64
47  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_0_FAHRENHEIT    float64
48  PRIMARY_STATION_11_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT   float64
49  PRIMARY_STATION_11_MONTHS_PRIOR_TEMPERATURE_AVERAGE                          float64
50  PRIMARY_STATION_11_MONTHS_PRIOR_TEMPERATURE_MAX                              float64
51  PRIMARY_STATION_11_MONTHS_PRIOR_TEMPERATURE_MIN                              float64
52  PRIMARY STATION 12 MONTHS PRIOR EXTREME MAXIMUM TEMPERATURE MONTH            float64
```

```
cols = numerical.columns
closest = []

for col in cols:
  if (col[:16] == 'CLOSEST_STATION_' or col in STATION_LIST or col == "S_LATITUDE" or col == "S_LONGITUDE"):
    closest.append(col)
```

```
primary_station_data_cols = []
for col in cols:
  if ("PRIMARY_STATION" in col or col in STATION_LIST or col == "S_LATITUDE" or col == "S_LONGITUDE"):
    primary_station_data_cols.append(col)

primary_station_df = pd.DataFrame()
for col in primary_station_data_cols:
  primary_station_df[col] = numerical[col]

primary_station_df["FIRE_SIZE"] = numerical["FIRE_SIZE"]
primary_station_df["DURATION"] = numerical["DURATION"]

```

```
NUM_FEATURES_TO_KEEP = 300 + 2
correlation = primary_station_df.corr(method='pearson')
highest_correlation = (correlation.nlargest(NUM_FEATURES_TO_KEEP, 'FIRE_SIZE').index)
print(f"TOP {NUM_FEATURES_TO_KEEP} FEATURES WITH HIGHEST CORRELATION TO FIRE SIZE")
print(list(highest_correlation))

print(f"TOP 10 FEATURES WITH HIGHEST CORRELATION TO FIRE SIZE")
display(highest_correlation[2:12])
display(correlation.nlargest(NUM_FEATURES_TO_KEEP, 'FIRE_SIZE')[2:12]["FIRE_SIZE"])

fire_size_prediction_df = primary_station_df[highest_correlation]
del fire_size_prediction_df["FIRE_SIZE"]
del fire_size_prediction_df["DURATION"]
```

```
TOP 302 FEATURES WITH HIGHEST CORRELATION TO FIRE SIZE
['FIRE_SIZE', 'DURATION', 'PRIMARY_STATION_13_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH', 'PRIMARY_STATION_1_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH', 'PRIMARY_STATION_13_MONTHS_PRIOR_TEMPERATURE_MAX', 'PRIMARY_STATION_1_MON
TOP 10 FEATURES WITH HIGHEST CORRELATION TO FIRE SIZE
Index(['PRIMARY_STATION_13_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH',
       'PRIMARY_STATION_1_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH',
       'PRIMARY_STATION_13_MONTHS_PRIOR_TEMPERATURE_MAX',
       'PRIMARY_STATION_1_MONTHS_PRIOR_TEMPERATURE_MAX',
       'PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT',
       'PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT',
       'PRIMARY_STATION_13_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT',
       'PRIMARY_STATION_1_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT',
       'PRIMARY_STATION_12_MONTHS_PRIOR_TEMPERATURE_MAX',
       'PRIMARY_STATION_12_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH'],
      dtype='object')
PRIMARY_STATION_13_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH             0.022363
PRIMARY_STATION_1_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH              0.022363
PRIMARY_STATION_13_MONTHS_PRIOR_TEMPERATURE_MAX                               0.020201
PRIMARY_STATION_1_MONTHS_PRIOR_TEMPERATURE_MAX                                0.020201
PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT    0.019643
PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT    0.019516
PRIMARY_STATION_13_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT    0.019503
PRIMARY_STATION_1_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_90_FAHRENHEIT     0.019503
PRIMARY_STATION_12_MONTHS_PRIOR_TEMPERATURE_MAX                               0.019122
PRIMARY_STATION_12_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH             0.018295
Name: FIRE_SIZE, dtype: float64
```

```
display(fire_size_prediction_df)
```

ITH_MIN_TEMP_BELOW_32_FAHRENHEIT	PRIMARY_STATION_ELEVATION	PRIMARY_STATION_12_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT	PRIMARY_STATION_2_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT	PRIMARY_STATION_14_MONTHS
1.574436	1.192259	1.643791	1.877825	
-0.935826	0.525056	0.501069	0.161437	
-0.935826	0.525056	-0.935826	1.374855	
1.088624	1.808604	1.088624	1.563838	
1.088624	1.808604	1.088624	1.563838	
...	...	...	...	
-0.935826	-1.008744	-0.935826	-0.278662	
-0.935826	0.180489	-0.935826	-1.143144	
-0.935826	-1.008744	-0.935826	-1.130569	
-0.935826	-1.266260	-0.935826	-0.354108	
-0.935826	0.180489	-0.935826	-1.143144	

```
NUM_FEATURES_TO_KEEP = 300 + 2
correlation = primary_station_df.corr(method='pearson')
highest_correlation = (correlation.nlargest(NUM_FEATURES_TO_KEEP, 'DURATION').index)
print(f"TOP {NUM_FEATURES_TO_KEEP} FEATURES WITH HIGHEST CORRELATION TO FIRE DURATION")
display(highest_correlation)

print(f"TOP 10 FEATURES WITH HIGHEST CORRELATION TO FIRE DURATION")
display(highest_correlation[2:12])
display(correlation.nlargest(NUM_FEATURES_TO_KEEP, 'DURATION')[2:12]["DURATION"])

fire_duration_prediction_df = primary_station_df[highest_correlation]
del fire_duration_prediction_df["FIRE_SIZE"]
del fire_duration_prediction_df["DURATION"]
```

```
TOP 302 FEATURES WITH HIGHEST CORRELATION TO FIRE DURATION
Index(['DURATION', 'FIRE_SIZE',
       'PRIMARY_STATION_6_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT',
       'PRIMARY_STATION_6_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT',
       'PRIMARY_STATION_7_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT',
       'PRIMARY_STATION_ELEVATION',
       'PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT',
       'PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT',
       'PRIMARY_STATION_7_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT',
       'PRIMARY_STATION_16_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT',
       ...
       'PRIMARY_STATION_6_MONTHS_PRIOR_NUM_DAYS_WITH_MAX_TEMP_ABOVE_70_FAHRENHEIT',
       'PRIMARY_STATION_5_MONTHS_PRIOR_TEMPERATURE_MIN',
       'PRIMARY_STATION_5_MONTHS_PRIOR_TEMPERATURE_AVERAGE',
       'PRIMARY_STATION_7_MONTHS_PRIOR_TEMPERATURE_AVERAGE',
       'PRIMARY_STATION_7_MONTHS_PRIOR_TEMPERATURE_MIN',
       'PRIMARY_STATION_6_MONTHS_PRIOR_TEMPERATURE_MAX',
       'PRIMARY_STATION_6_MONTHS_PRIOR_TEMPERATURE_MIN',
       'PRIMARY_STATION_7_MONTHS_PRIOR_EXTREME_MINIMUM_TEMPERATURE_FOR_MONTH',
       'PRIMARY_STATION_6_MONTHS_PRIOR_TEMPERATURE_AVERAGE',
       'PRIMARY_STATION_6_MONTHS_PRIOR_EXTREME_MAXIMUM_TEMPERATURE_MONTH'],
      dtype='object', length=233)
TOP 10 FEATURES WITH HIGHEST CORRELATION TO FIRE DURATION
Index(['PRIMARY_STATION_6_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT',
       'PRIMARY_STATION_6_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT',
       'PRIMARY_STATION_7_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT',
       'PRIMARY_STATION_ELEVATION',
       'PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT',
       'PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT',
       'PRIMARY_STATION_7_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT',
       'PRIMARY_STATION_16_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT',
       'PRIMARY_STATION_4_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT',
       'PRIMARY_STATION_3_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT'],
      dtype='object')
PRIMARY_STATION_6_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT    0.072410
PRIMARY_STATION_6_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT     0.070551
PRIMARY_STATION_7_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT    0.066327
PRIMARY_STATION_ELEVATION                                                     0.065809
PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT     0.064962
PRIMARY_STATION_5_MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT    0.064641
PRIMARY_STATION_7_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT     0.064181
PRIMARY_STATION_16_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT    0.059995
PRIMARY_STATION_4_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT     0.059995
```

```
display(fire_duration_prediction_df)
```

MONTHS_PRIOR_NUM_DAYS_WHERE_AVG_TEMP_BELOW_65_FAHRENHEIT	PRIMARY_STATION_7_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT	PRIMARY_STATION_16_MONTHS_PRIOR_NUM_DAYS_WITH_MIN_TEMP_BELOW_32_FAHRENHEIT	PRIMARY_STATION_4_MONTHS_PRIC
-0.288093	-0.935826	1.031302	
1.506884	-0.232580	1.243994	
1.321414	1.291149	1.609640	
2.798891	1.643791	1.643791	
2.798891	1.643791	1.643791	
...	...	...	
0.425497	-0.232580	-0.232580	
-0.093192	-0.028466	-0.935826	
-0.335247	0.837331	0.136035	
0.060843	-0.935826	-0.935826	
-0.093192	-0.028466	-0.935826	

```
closest_df = pd.DataFrame()

for col in closest:
  closest_df[col] = numerical[col]
```

```
closest_df.head()
```

	BODIE	BROOKS	COHASSET	EEL_RIVER	HELL_HOLE	HERNANDEZ	HUNTER_MOUNTAIN	JUANITA_LAKE	LADDER_BUTTE	LAS_TABLAS	LA_HONDA	OAK_CREEK	PANAMINT	PILOT_HILL	SCORPION	SOLDIER_MOUNTAIN	SQUAW_LAKE	STAMPEDE	VAN_BREMMER	WOLVERTON	S_
0	-0.501752	-0.841998	-1.128433	-0.848324	-1.000659	0.195610	0.434939	-0.886032	-1.114178	0.448456	-0.394781	0.297877	0.545114	-0.914818	-0.846454	-1.044163	0.687200	-1.139269	-0.918627	0.410736	
1	-1.237103	-0.939629	-0.802799	-0.597768	-1.471873	-0.562875	-0.243079	-0.487813	-0.669480	-0.281374	-0.869095	-0.451833	-0.057172	-1.280961	-0.473194	-0.612901	0.315528	-1.287797	-0.513504	-0.311271	
2	-1.070104	-1.059637	-0.891098	-0.695749	-1.410920	-0.543260	-0.109734	-0.533929	-0.714976	-0.262071	-0.950563	-0.317057	0.059122	-1.392229	-0.537641	-0.663661	0.382606	-1.233892	-0.557916	-0.204482	
3	-1.588084	-0.752356	-0.586044	-0.401895	-1.213089	-0.755069	-0.564467	-0.322165	-0.486222	-0.477673	-0.836997	-0.795000	-0.341413	-1.070171	-0.296636	-0.431126	0.143227	-1.125369	-0.346639	-0.618217	
4	-1.578681	-0.759860	-0.590604	-0.407106	-1.217813	-0.758604	-0.557924	-0.324379	-0.488626	-0.480494	-0.845502	-0.788869	-0.335760	-1.077584	-0.299859	-0.433722	0.146406	-1.126335	-0.348787	-0.614275	

Model 1: Linear Regression

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error, median_absolute_error, r2_score
from sklearn.metrics import explained_variance_score
from sklearn import svm
from sklearn.preprocessing import PowerTransformer
from sklearn.ensemble import RandomForestRegressor

# def log_transform(column):
#   """
#   Transforms a feature so that it is scaled logarithmically. Useful for correcting floating point errors
#   """


# def min_max_transform(column):
#   """
#   Transforms a feature using min-max scaling
#   """
#   pass

def run_full_linear_regression_with_accuracy(features, response, n_highest = 0):
```

```
  print(f"Response Minimum: {response.min()}")
  print(f"Response Maximum: {response.max()}")

  model = LinearRegression()
  # model = Ridge()
  # model = svm.SVR()
  # Split into test and training set
  X_train, X_test, y_train, y_test = train_test_split(features, response, test_size=0.2, random_state=2020)
  model.fit(X_train, y_train)

  training_predictions = model.predict(X_train)
  test_predictions = model.predict(X_test)
  print(f"Linear Regression Coefficients: { model.coef_ }\nLinear Regression Intercept: { model.intercept_}\n")

  print("Training: mean absolute error: ", mean_absolute_error(y_train, training_predictions))
  print("Test mean absolute error: ", mean_absolute_error(y_test, test_predictions))

  print("Training average error rate: ", median_absolute_error(y_train, training_predictions))
  print("Test average error rate: ", median_absolute_error(y_test, test_predictions))

  print("Explained variance in training set is: ", explained_variance_score(y_train, training_predictions))
  print("Explained variance in test set is: ", explained_variance_score(y_test, test_predictions))

  print("R2 Score training set: ", r2_score(y_train, training_predictions))
  print("R2 Score testing set: ", r2_score(y_test, test_predictions))

  fig = plt.figure()
  ax1 = fig.add_subplot()
  ax1.set_xlabel("True Value from Test Set")
  ax1.set_ylabel('Prediction from Test Set')
  ax1.set_title('True Value vs Predicted Value for Test Set: Linear Regression')
  ax1.scatter(y_test, test_predictions)
  plt.show()

  fig = plt.figure()
  ax1 = fig.add_subplot()
  ax1.set_xlabel("Predicted Value from Test Set")
  ax1.set_ylabel('Residual Value from Test Set')
  ax1.set_title('Test Set Residual Values vs Predicted Values')
  ax1.scatter(y_train, training_predictions)
  plt.show()


  # Residual plot against predictor
  fig = plt.figure()
  ax1 = fig.add_subplot()
  ax1.set_xlabel("Test Set Prediction")
  ax1.set_ylabel('Prediction from Training Set')
  ax1.set_title('Residual Graph: Linear Regression')
  ax1.scatter(training_predictions, y_train - training_predictions)
  plt.show()



def run_full_ridge_regression_with_accuracy(features, response, n_highest = 0):
  print(f"Response Minimum: {response.min()}")
  print(f"Response Maximum: {response.max()}")


  model = Ridge()
  # Split into test and training set
  X_train, X_test, y_train, y_test = train_test_split(features, response, test_size=0.2, random_state=2020)
  model.fit(X_train, y_train)

  training_predictions = model.predict(X_train)
  test_predictions = model.predict(X_test)
  print(f"Ridge Regression Coefficients: { model.coef_ }\Ridge Regression Intercept: { model.intercept_}\n")

  print("Training: mean absolute error: ", mean_absolute_error(y_train, training_predictions))
  print("Test mean absolute error: ", mean_absolute_error(y_test, test_predictions))

  print("Training average error rate: ", median_absolute_error(y_train, training_predictions))
  print("Test average error rate: ", median_absolute_error(y_test, test_predictions))

  print("Explained variance in training set is: ", explained_variance_score(y_train, training_predictions))
  print("Explained variance in test set is: ", explained_variance_score(y_test, test_predictions))

  print("R2 Score training set: ", r2_score(y_train, training_predictions))
  print("R2 Score testing set: ", r2_score(y_test, test_predictions))

  fig = plt.figure()
  ax1 = fig.add_subplot()
  ax1.set_xlabel("True Value from Test Set")
  ax1.set_ylabel('Prediction from Test Set')
  ax1.set_title('True Value vs Predicted Value for Test Set: Ridge Regression')
  ax1.scatter(y_test, test_predictions)
  plt.show()

  fig = plt.figure()

  ax1 = fig.add_subplot()
  ax1.set_xlabel("True Value from Training Set")
  ax1.set_ylabel('Prediction from Training Set')
  ax1.set_title('True Value vs Predicted Value for Training Set: Ridge Regression')
  ax1.scatter(y_train, training_predictions)
  plt.show()

  fig = plt.figure()
  ax1 = fig.add_subplot()
  ax1.set_xlabel("Test Set Prediction")
  ax1.set_ylabel('Prediction from Training Set')
  ax1.set_title('Residual Graph: Ridge Regression')
  ax1.scatter(training_predictions, y_train - training_predictions)
  plt.show()

def run_full_SVM_regression_with_accuracy(features, response, n_highest = 0):
  print(f"Response Minimum: {response.min()}")
  print(f"Response Maximum: {response.max()}")


  model = svm.SVR()
```

```
  # Split into test and training set
  X_train, X_test, y_train, y_test = train_test_split(features, response, test_size=0.2, random_state=2020)
  model.fit(X_train, y_train)

  training_predictions = model.predict(X_train)
  test_predictions = model.predict(X_test)

  print("Training: mean absolute error: ", mean_absolute_error(y_train, training_predictions))
  print("Test mean absolute error: ", mean_absolute_error(y_test, test_predictions))

  print("Training average error rate: ", median_absolute_error(y_train, training_predictions))
  print("Test average error rate: ", median_absolute_error(y_test, test_predictions))

  print("Explained variance in training set is: ", explained_variance_score(y_train, training_predictions))
  print("Explained variance in test set is: ", explained_variance_score(y_test, test_predictions))

  fig = plt.figure()
  ax1 = fig.add_subplot()
  ax1.set_xlabel("True Value from Test Set")
  ax1.set_ylabel('Prediction from Test Set')
  ax1.set_title('True Value vs Predicted Value for Test Set: SVM Regression')
  ax1.scatter(y_test, test_predictions)
  plt.show()

  fig = plt.figure()

  ax1 = fig.add_subplot()
  ax1.set_xlabel("True Value from Training Set")
  ax1.set_ylabel('Prediction from Training Set')
  ax1.set_title('True Value vs Predicted Value for Training Set: SVM Regression')
  ax1.scatter(y_train, training_predictions)
  plt.show()

def run_full_random_forest_regression_with_accuracy(features, response, n_highest = 0):
  print(f"Response Minimum: {response.min()}")
  print(f"Response Maximum: {response.max()}")


  model = RandomForestRegressor(n_estimators=100)
  # Split into test and training set
  X_train, X_test, y_train, y_test = train_test_split(features, response, test_size=0.2, random_state=2020)
  model.fit(X_train, y_train)

  training_predictions = model.predict(X_train)
  test_predictions = model.predict(X_test)

  print("Training: mean absolute error: ", mean_absolute_error(y_train, training_predictions))
  print("Test mean absolute error: ", mean_absolute_error(y_test, test_predictions))

  print("Training average error rate: ", median_absolute_error(y_train, training_predictions))
  print("Test average error rate: ", median_absolute_error(y_test, test_predictions))

  print("Explained variance in training set is: ", explained_variance_score(y_train, training_predictions))
  print("Explained variance in test set is: ", explained_variance_score(y_test, test_predictions))

  fig = plt.figure()
  ax1 = fig.add_subplot()
  ax1.set_xlabel("True Value from Test Set")
  ax1.set_ylabel('Prediction from Test Set')
  ax1.set_title('True Value vs Predicted Value for Test Set: Random Forest Regression')
  ax1.scatter(y_test, test_predictions)
  plt.show()

  fig = plt.figure()

  ax1 = fig.add_subplot()
  ax1.set_xlabel("True Value from Training Set")
  ax1.set_ylabel('Prediction from Training Set')
  ax1.set_title('True Value vs Predicted Value for Training Set: Random Forest Regression')
  ax1.scatter(y_train, training_predictions)
  plt.show()
```

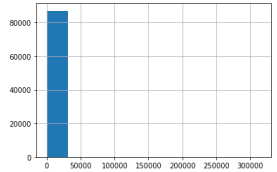
```
SUBSET = -1
```

Fire Size Prediction using Linear Regression and Ridge Regression

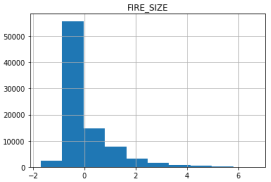
Here, we predict the size of a fire using linear regression

```
FIRE_SIZE = numerical["FIRE_SIZE"]
FIRE_SIZE.hist()
```

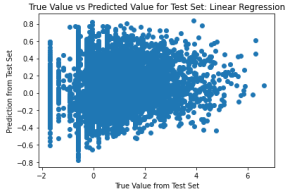
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1061ebab10>
```

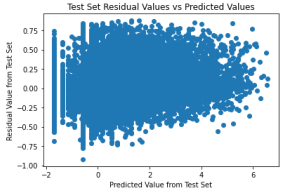


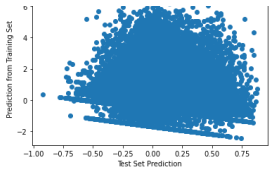

```
log_fire_size = pd.DataFrame(np.log(FIRE_SIZE))
log_fire_size = log_fire_size.replace([np.inf, -np.inf], np.nan)
log_fire_size = log_fire_size.fillna(0)
log_fire_size = (log_fire_size - log_fire_size.mean()) / log_fire_size.std()
log_fire_size.hist()
plt.show()

run_full_linear_regression_with_accuracy(fire_size_prediction_df[:SUBSET], log_fire_size[:SUBSET])
run_full_ridge_regression_with_accuracy(fire_size_prediction_df[:SUBSET], log_fire_size[:SUBSET])
```



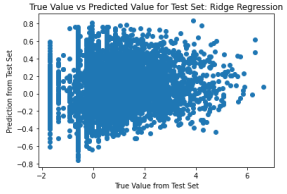

```
Response Minimum: FIRE_SIZE   -1.698422
dtype: float64
Response Maximum: FIRE_SIZE    6.638909
dtype: float64
Linear Regression Coefficients: [[ 1.03704077e+08 -1.03704077e+08  1.47492378e+10 -1.47492378e+10
  -3.64134423e-02  1.77510045e-02 -2.09909761e+10  2.09909761e+10
   3.66966471e-01 -5.58446348e-03  2.70166660e+10 -2.70166660e+10
  -1.42716001e+10  1.42716001e+10 -1.72787955e+10  1.72787955e+10
  -1.78996530e+10  1.78996530e+10  5.89141995e-02  1.57028880e+10
  -1.57028880e+10 -3.12987808e-02 -9.36800763e-02 -2.84259652e-01
  -8.77977069e-03  3.94475441e+09 -3.94475441e+09 -6.43242757e+09
   6.43242757e+09  4.95082214e-02  7.37310301e+10 -7.37310301e+10
  -1.06289991e+09  1.06289990e+09  7.54375549e+09 -7.54375549e+09
   3.59283332e+09 -3.59283332e+09 -1.90976778e+07  1.90976777e+07
   3.10789346e+09 -3.10789346e+09  1.30289919e+08 -1.30289920e+08
   3.96357816e+08 -3.96357816e+08 -8.76759070e+09  8.76759070e+09
   4.60180044e-02  2.22729209e+08 -2.22729209e+08  1.47006300e+09
  -1.47006300e+09 -3.30738068e-01  2.42208195e+00 -7.91138649e-01
  -3.46548065e+09  3.46548065e+09 -1.96505081e+09  1.96505081e+09
  -1.37278080e+00 -6.74629211e-02  2.78862157e+08 -2.78862157e+08
  -4.31046438e+07  4.31046439e+07  2.18398702e+07 -2.18398700e+07
   8.70097172e+06 -8.70097177e+06  1.07774734e-02  4.79607582e-02
  -1.27549458e+00 -5.22060394e-02  2.23098469e+00  4.82935905e-02
  -2.77605724e+00 -5.61192088e+08  5.61192088e+08  2.32509041e+00
  -1.50121772e+00 -1.33900642e-02  3.94990921e-01  3.89124393e-01
  -5.48233986e-02  1.06481999e+09 -1.06481999e+09  3.97219682e+09
  -3.97219682e+09 -1.22138149e+09  1.22138149e+09 -3.83896915e+09
   3.83896915e+09 -2.81248093e-02  1.35421753e-03  5.50117493e-02
   2.52781029e+09 -2.52781029e+09  9.65213776e-02  2.13899612e-02
   1.07672501e+00 -3.47785950e-02 -3.58829498e-02 -3.50656509e-02
  -9.05838013e-02  8.71037347e+08 -8.71037347e+08  2.07498927e+09
  -2.07498927e+09  2.34832764e-02  6.14059794e+08 -6.14059794e+08
   1.48298740e-02 -7.13648796e-02 -3.11574936e-02  4.96406555e-02
   4.19883901e+00  1.31343365e-01 -2.62155533e-02 -2.04658508e-02
  -7.95698166e-03  1.06210709e-02 -7.92312622e-02 -3.80210876e-02
   6.57653809e-03 -5.98640442e-02 -2.41565704e-03 -3.71236801e-02
   9.07802582e-03  1.43840313e-01 -5.66997528e-02  1.06575775e+00
  -9.81597900e-02  3.21168900e-02 -4.14981842e-02  1.47008896e-03
  -4.26416397e-02  6.22899256e+08 -6.22899256e+08  2.93273926e-02
   5.68656921e-02  7.85408020e-02 -4.91542816e-01  3.04281712e-01
  -3.46946716e-02 -4.57048416e-03 -5.71763064e+08  5.71763064e+08
   8.95407200e-02  2.28118896e-03 -4.97293949e-01 -7.67070770e-01
  -6.68535233e-02 -2.65884399e-02  4.46219444e-02  5.38037300e-01
  -1.23427391e-01  4.80651855e-04 -5.65294743e-01 -1.05566120e+00
   3.51715088e-03 -1.26923597e+08  1.26923597e+08  4.25128937e-01
  -3.40239882e-01 -2.53200004e+09  2.53200004e+09 -2.10371017e-02
   7.10717866e+08 -7.10717866e+08  2.51319885e-01  4.42872524e-01
   1.09748840e-02  9.76252174e+07 -9.76252174e+07  8.23974609e-04
  -5.34629822e-03  2.08721161e-02  5.14984131e-05  1.78281094e+09
  -1.78281094e+09  2.61077881e-02  6.81924820e-03 -1.70917511e-02
   5.40506840e-02 -1.64546967e-02  8.01086426e-04 -1.01178408e+10
   1.01178408e+10  2.04987216e+09 -2.04987216e+09 -1.92438835e+09
   1.92438835e+09 -1.17696285e-01  4.97193336e-02 -5.83720207e-03
  -5.07464170e-01  6.06273651e-01  8.73763388e+08 -8.73763387e+08
   1.32496488e+00 -3.09833527e-01  1.93934441e-02  1.87454224e-02
   2.61116028e-02 -1.68700218e-02  2.69813538e-02  6.30855560e-03
   3.98845673e-02  2.10609436e-02 -7.80124664e-02 -6.96214497e-01
   9.76101449e+08 -9.76101450e+08 -2.87990570e-02 -3.52430344e-02
   5.19652367e-02 -6.83453011e+08  6.83453011e+08  1.45538918e+07
  -1.45538918e+07  3.06806564e-02 -6.37801996e+08  6.37801996e+08
   1.27232552e-01 -4.67758179e-02  3.44344351e+08 -3.44344351e+08
  -4.59327698e-02 -1.02490783e-01  6.69097900e-03]]
Linear Regression Intercept: [-1.41717782]

Training: mean absolute error:  0.7022413703548958
Test mean absolute error:  0.700650561286869
Training average error rate:  0.5400948672942414
Test average error rate:  0.538445372454226
Explained variance in training set is:  0.049671639216195884
Explained variance in test set is:  0.04495804308808271
R2 Score training set:  0.04967163909210026
R2 Score testing set:  0.044928275789205
```

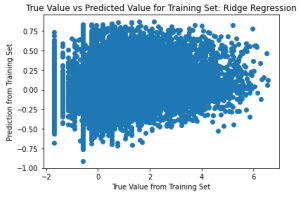


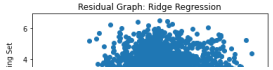


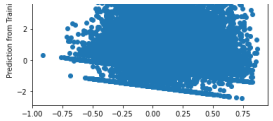



```
Response Minimum: FIRE_SIZE    -1.698422
dtype: float64
Response Maximum: FIRE_SIZE    6.638909
dtype: float64
Ridge Regression Coefficients: [[ 4.33594698e-02  4.33594698e-02  1.03918029e-01  1.03918029e-01
  -3.55139592e-02  1.94372343e-02 -2.37730649e-03 -2.37730649e-03
   2.95197670e-01 -5.50421501e-03 -1.64275460e-02 -1.64275460e-02
   2.09743454e-02  2.09743454e-02 -2.73514574e-03 -2.73514574e-03
  -2.32735103e-01 -2.32735103e-01  5.67128831e-02  1.03293300e-01
   1.03293300e-01 -3.10742784e-02 -9.37498845e-02 -1.59790510e-01
  -8.65886988e-03 -2.02195149e-03 -2.02195149e-03  1.63489531e-01
   1.63489531e-01  4.85216754e-02 -7.13064955e-03 -7.13064954e-03
  -1.53354316e-01 -1.53354316e-01 -1.89397625e-03 -1.89397625e-03
   1.25155780e-01  1.25155780e-01 -3.62207874e-02 -3.62207874e-02
   3.87649977e-02  3.87649977e-02 -2.95336422e-01 -2.95336422e-01
  -7.49515622e-03 -7.49515622e-03 -1.50014523e-02 -1.50014523e-02
   4.71990775e-02  1.57950987e-02  1.57950987e-02  3.95388936e-02
   3.95388936e-02 -2.96773002e-01  2.17586250e+00 -6.20496107e-01
   8.02527227e-03  8.02527227e-03  1.92007303e-02  1.92007303e-02
  -1.21936264e+00 -1.25427278e-01  3.96608944e-02  3.96608944e-02
   2.29678042e-02  2.29678042e-02  8.13495083e-02  8.13495083e-02
  -2.46745638e-02 -2.46745638e-02  1.22420091e-02 -2.83519348e-02
  -1.29031438e+00 -4.49127615e-02  1.43418219e+00  4.66073026e-02
  -1.64123986e+00  6.40358408e-03  6.40358408e-03  1.08638129e+00
  -4.82500336e-01 -1.10822859e-02  3.25933961e-01  1.96258308e-01
   3.96117084e-02 -5.51759323e-02 -5.51759323e-02 -4.70122684e-03
  -4.70122683e-03  1.17971539e-02  1.17971539e-02 -1.06617077e-02
  -1.06617077e-02 -2.67885150e-02  1.70111535e-03  5.34686540e-02
  -1.99321352e-02 -1.99321352e-02  9.68986125e-02  1.97705340e-02
   1.08665133e+00 -3.54806340e-02 -3.58292019e-02 -2.88611820e-02
  -6.10815315e-02  2.84367189e-02  2.84367189e-02 -1.93343131e-02
  -1.93343131e-02  2.27118163e-02 -6.85355161e-04 -6.85355161e-04
   1.43410472e-02 -7.64684468e-02 -3.35278327e-02  4.95580216e-02
   3.39656156e+00  1.31859290e-01 -2.77583729e-02 -1.97469144e-02
  -7.52814716e-03  1.16471480e-02 -7.92131730e-02 -3.61492458e-02
   6.83011830e-03 -6.34140872e-02 -2.91079150e-03 -3.86656895e-02
   8.66979561e-03  1.46205124e-01 -5.60352074e-02  9.68871324e-01
  -9.80762352e-02  3.17596654e-02 -4.23048647e-02  1.02074299e-03
  -4.34179526e-02 -7.49110514e-03 -7.49110514e-03  2.83423971e-02
   5.44439329e-02  7.70594034e-02 -5.35758603e-01  2.60590537e-01
  -3.45819843e-02 -6.60734163e-03 -1.77984248e-02 -1.77984248e-02
   6.42804807e-02  2.33412963e-03 -4.30558901e-01 -7.29125279e-01
  -6.79223733e-02 -2.72924161e-02  4.29888154e-02  4.11278431e-01
  -1.21747881e-01  1.33964358e-04 -4.82635951e-01 -8.23258453e-01
   2.90044843e-03  1.04770930e-02  1.04770930e-02  3.16474061e-01
  -2.91603393e-01 -4.48059053e-04 -4.48059053e-04 -2.11956074e-02
   1.14757576e-02  1.14757576e-02  2.13991992e-01  4.25428791e-01
   9.23604515e-03  1.41535283e-02  1.41535283e-02 -6.99421468e-05
  -7.03892942e-03  2.11925108e-02  5.42289730e-04  1.32431686e-02
   1.32431686e-02  2.52032969e-02  6.70447767e-03 -1.57652797e-02
   3.29509529e-02 -1.76445543e-02 -4.99090504e-04 -5.62255052e-04
  -5.62255052e-04 -9.27821626e-04 -9.27821626e-04  2.52165762e-02
   2.52165762e-02 -6.20699629e-02  4.77016420e-02 -6.73406909e-03
  -4.34379424e-01  4.80174177e-01  2.66107395e-02  2.66107395e-02
   1.06272342e+00 -2.54940299e-01  1.76320769e-02  1.83135442e-02
   2.58548703e-02 -1.71273569e-02  2.77324148e-02  6.47875586e-03
   3.91878129e-02  2.19523807e-02 -7.91176431e-02 -3.95938707e-01
  -5.57587801e-02 -5.57587801e-02 -2.79974557e-02 -3.79393033e-02
   5.09759789e-02 -1.27408458e-04 -1.27408458e-04  5.09244746e-05
   5.09244749e-05  3.14949159e-02 -9.60952552e-03 -9.60952552e-03
   1.21714090e-01 -4.72507300e-02  6.10308126e-04  6.10308126e-04
  -4.65120694e-02 -9.99586537e-02  5.44285772e-03]]\Ridge Regression Intercept: [-1.24172231]

Training: mean absolute error:  0.7025343078227769
Test mean absolute error:  0.7009320623969 89
Training average error rate:  0.5407735980566852
Test average error rate:  0.5394821140280315
Explained variance in training set is:  0.049534174468021 33
Explained variance in test set is:  0.04486034715527376
R2 Score training set:  0.04953417446802111
R2 Score testing set:  0.04482933788042309
```









Basic Neural Network Model

```
import tensorflow as tf
```

```
log_fire_size = pd.DataFrame(np.log(FIRE_SIZE))
log_fire_size = log_fire_size.replace([np.inf, -np.inf], np.nan)
log_fire_size = log_fire_size.fillna(0)
log_fire_size = (log_fire_size - log_fire_size.mean()) / log_fire_size.std()

X_train, X_test, y_train, y_test = train_test_split(fire_size_prediction_df, log_fire_size, test_size=0.2, random_state=2020)
```

```
from tensorflow.keras.layers import *
from tensorflow.keras.losses import MeanSquaredError
print(X_train.shape)


model = tf.keras.models.Sequential([Dense(231, activation="relu"),
                                    Dropout(0.2),
                                    Dense(231, activation="relu"),
                                    Dense(231, activation="relu"),
                                    Dense(1)])

loss_function = MeanSquaredError()
model.compile(optimizer="adam", loss=loss_function, metrics=["mean_squared_error", "mean_absolute_error"])
model.fit(X_train, y_train, epochs = 300)
model.evaluate(X_test, y_test, verbose=2)


test_predictions = model.predict(X_test)
fig = plt.figure()
ax1 = fig.add_subplot()
ax1.set_xlabel("True Value from Test Set")
ax1.set_ylabel('Prediction from Test Set')
ax1.set_title('True Value vs Predicted Value for Test Set: Simple Neural Network')
ax1.scatter(y_test, test_predictions)
plt.show()

print(model.summary())
```

```
(69684, 231)
Epoch 1/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.9930 - mean_squared_error: 0.9930 - mean_absolute_error: 0.7215
Epoch 2/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9731 - mean_squared_error: 0.9731 - mean_absolute_error: 0.7130
Epoch 3/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9692 - mean_squared_error: 0.9692 - mean_absolute_error: 0.7113
Epoch 4/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9647 - mean_squared_error: 0.9647 - mean_absolute_error: 0.7087
Epoch 5/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9621 - mean_squared_error: 0.9621 - mean_absolute_error: 0.7080
Epoch 6/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9584 - mean_squared_error: 0.9584 - mean_absolute_error: 0.7058
Epoch 7/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9554 - mean_squared_error: 0.9554 - mean_absolute_error: 0.7046
Epoch 8/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9539 - mean_squared_error: 0.9539 - mean_absolute_error: 0.7047
Epoch 9/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9513 - mean_squared_error: 0.9513 - mean_absolute_error: 0.7026
Epoch 10/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9495 - mean_squared_error: 0.9495 - mean_absolute_error: 0.7023
Epoch 11/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9483 - mean_squared_error: 0.9483 - mean_absolute_error: 0.7020
Epoch 12/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9466 - mean_squared_error: 0.9466 - mean_absolute_error: 0.7011
Epoch 13/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9457 - mean_squared_error: 0.9457 - mean_absolute_error: 0.7005
Epoch 14/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9438 - mean_squared_error: 0.9438 - mean_absolute_error: 0.6997
Epoch 15/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9429 - mean_squared_error: 0.9429 - mean_absolute_error: 0.6992
Epoch 16/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9409 - mean_squared_error: 0.9409 - mean_absolute_error: 0.6985
Epoch 17/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9407 - mean_squared_error: 0.9407 - mean_absolute_error: 0.6981
Epoch 18/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9387 - mean_squared_error: 0.9387 - mean_absolute_error: 0.6972
Epoch 19/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9379 - mean_squared_error: 0.9379 - mean_absolute_error: 0.6969
Epoch 20/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9371 - mean_squared_error: 0.9371 - mean_absolute_error: 0.6966
Epoch 21/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9352 - mean_squared_error: 0.9352 - mean_absolute_error: 0.6965
Epoch 22/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9344 - mean_squared_error: 0.9344 - mean_absolute_error: 0.6955
Epoch 23/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9329 - mean_squared_error: 0.9329 - mean_absolute_error: 0.6946
Epoch 24/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9334 - mean_squared_error: 0.9334 - mean_absolute_error: 0.6945
Epoch 25/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9320 - mean_squared_error: 0.9320 - mean_absolute_error: 0.6941
Epoch 26/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9306 - mean_squared_error: 0.9306 - mean_absolute_error: 0.6933
Epoch 27/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9305 - mean_squared_error: 0.9305 - mean_absolute_error: 0.6933
Epoch 28/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9296 - mean_squared_error: 0.9296 - mean_absolute_error: 0.6936
Epoch 29/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9289 - mean_squared_error: 0.9289 - mean_absolute_error: 0.6931
Epoch 30/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9274 - mean_squared_error: 0.9274 - mean_absolute_error: 0.6920
Epoch 31/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9279 - mean_squared_error: 0.9279 - mean_absolute_error: 0.6925
Epoch 32/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9240 - mean_squared_error: 0.9240 - mean_absolute_error: 0.6907
Epoch 33/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9234 - mean_squared_error: 0.9234 - mean_absolute_error: 0.6905
Epoch 34/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9240 - mean_squared_error: 0.9240 - mean_absolute_error: 0.6908
Epoch 35/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9238 - mean_squared_error: 0.9238 - mean_absolute_error: 0.6905
Epoch 36/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9235 - mean_squared_error: 0.9235 - mean_absolute_error: 0.6905
Epoch 37/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9222 - mean_squared_error: 0.9222 - mean_absolute_error: 0.6890
Epoch 38/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9208 - mean_squared_error: 0.9208 - mean_absolute_error: 0.6890
Epoch 39/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9195 - mean_squared_error: 0.9195 - mean_absolute_error: 0.6881
Epoch 40/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.9193 - mean_squared_error: 0.9193 - mean_absolute_error: 0.6878
Epoch 41/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9189 - mean_squared_error: 0.9189 - mean_absolute_error: 0.6877
Epoch 42/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9188 - mean_squared_error: 0.9188 - mean_absolute_error: 0.6880
Epoch 43/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9166 - mean_squared_error: 0.9166 - mean_absolute_error: 0.6870
Epoch 44/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9154 - mean_squared_error: 0.9154 - mean_absolute_error: 0.6867
Epoch 45/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9159 - mean_squared_error: 0.9159 - mean_absolute_error: 0.6864
Epoch 46/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9127 - mean_squared_error: 0.9127 - mean_absolute_error: 0.6849
Epoch 47/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.9136 - mean_squared_error: 0.9136 - mean_absolute_error: 0.6856
Epoch 48/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9139 - mean_squared_error: 0.9139 - mean_absolute_error: 0.6862
Epoch 49/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.9144 - mean_squared_error: 0.9144 - mean_absolute_error: 0.6857
Epoch 50/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9117 - mean_squared_error: 0.9117 - mean_absolute_error: 0.6850
Epoch 51/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9110 - mean_squared_error: 0.9110 - mean_absolute_error: 0.6844
Epoch 52/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9095 - mean_squared_error: 0.9095 - mean_absolute_error: 0.6842
Epoch 53/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9094 - mean_squared_error: 0.9094 - mean_absolute_error: 0.6832
Epoch 54/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9073 - mean_squared_error: 0.9073 - mean_absolute_error: 0.6832
Epoch 55/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9072 - mean_squared_error: 0.9072 - mean_absolute_error: 0.6827
Epoch 56/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9075 - mean_squared_error: 0.9075 - mean_absolute_error: 0.6836
Epoch 57/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9049 - mean_squared_error: 0.9049 - mean_absolute_error: 0.6820
Epoch 58/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9045 - mean_squared_error: 0.9045 - mean_absolute_error: 0.6812
Epoch 59/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9039 - mean_squared_error: 0.9039 - mean_absolute_error: 0.6808
Epoch 60/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9038 - mean_squared_error: 0.9038 - mean_absolute_error: 0.6812
Epoch 61/300
```

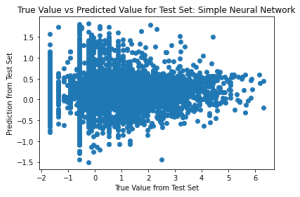
```
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9022 - mean_squared_error: 0.9022 - mean_absolute_error: 0.6802
Epoch 62/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9021 - mean_squared_error: 0.9021 - mean_absolute_error: 0.6797
Epoch 63/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9029 - mean_squared_error: 0.9029 - mean_absolute_error: 0.6809
Epoch 64/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9023 - mean_squared_error: 0.9023 - mean_absolute_error: 0.6792
Epoch 65/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9014 - mean_squared_error: 0.9014 - mean_absolute_error: 0.6799
Epoch 66/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9004 - mean_squared_error: 0.9004 - mean_absolute_error: 0.6788
Epoch 67/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8998 - mean_squared_error: 0.8998 - mean_absolute_error: 0.6793
Epoch 68/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.9001 - mean_squared_error: 0.9001 - mean_absolute_error: 0.6782
Epoch 69/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8967 - mean_squared_error: 0.8967 - mean_absolute_error: 0.6783
Epoch 70/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8981 - mean_squared_error: 0.8981 - mean_absolute_error: 0.6781
Epoch 71/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8969 - mean_squared_error: 0.8969 - mean_absolute_error: 0.6773
Epoch 72/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8962 - mean_squared_error: 0.8962 - mean_absolute_error: 0.6778
Epoch 73/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8967 - mean_squared_error: 0.8967 - mean_absolute_error: 0.6771
Epoch 74/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8967 - mean_squared_error: 0.8967 - mean_absolute_error: 0.6779
Epoch 75/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.8954 - mean_squared_error: 0.8954 - mean_absolute_error: 0.6768
Epoch 76/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.8928 - mean_squared_error: 0.8928 - mean_absolute_error: 0.6771
Epoch 77/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8917 - mean_squared_error: 0.8917 - mean_absolute_error: 0.6752
Epoch 78/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.8938 - mean_squared_error: 0.8938 - mean_absolute_error: 0.6767
Epoch 79/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8941 - mean_squared_error: 0.8941 - mean_absolute_error: 0.6757
Epoch 80/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8930 - mean_squared_error: 0.8930 - mean_absolute_error: 0.6760
Epoch 81/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8937 - mean_squared_error: 0.8937 - mean_absolute_error: 0.6765
Epoch 82/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.8926 - mean_squared_error: 0.8926 - mean_absolute_error: 0.6758
Epoch 83/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.8903 - mean_squared_error: 0.8903 - mean_absolute_error: 0.6738
Epoch 84/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.8921 - mean_squared_error: 0.8921 - mean_absolute_error: 0.6755
Epoch 85/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.8887 - mean_squared_error: 0.8887 - mean_absolute_error: 0.6739
Epoch 86/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.8912 - mean_squared_error: 0.8912 - mean_absolute_error: 0.6750
Epoch 87/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.8888 - mean_squared_error: 0.8888 - mean_absolute_error: 0.6741
Epoch 88/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8874 - mean_squared_error: 0.8874 - mean_absolute_error: 0.6736
Epoch 89/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8896 - mean_squared_error: 0.8896 - mean_absolute_error: 0.6745
Epoch 90/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8871 - mean_squared_error: 0.8871 - mean_absolute_error: 0.6732
Epoch 91/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8855 - mean_squared_error: 0.8855 - mean_absolute_error: 0.6722
Epoch 92/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8865 - mean_squared_error: 0.8865 - mean_absolute_error: 0.6736
Epoch 93/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8863 - mean_squared_error: 0.8863 - mean_absolute_error: 0.6724
Epoch 94/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8850 - mean_squared_error: 0.8850 - mean_absolute_error: 0.6709
Epoch 95/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8828 - mean_squared_error: 0.8828 - mean_absolute_error: 0.6714
Epoch 96/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8837 - mean_squared_error: 0.8837 - mean_absolute_error: 0.6716
Epoch 97/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8846 - mean_squared_error: 0.8846 - mean_absolute_error: 0.6710
Epoch 98/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8872 - mean_squared_error: 0.8872 - mean_absolute_error: 0.6725
Epoch 99/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.8827 - mean_squared_error: 0.8827 - mean_absolute_error: 0.6702
Epoch 100/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8817 - mean_squared_error: 0.8817 - mean_absolute_error: 0.6704
Epoch 101/300
2178/2178 [==============================] - 7s 3ms/step - loss: 0.8816 - mean_squared_error: 0.8816 - mean_absolute_error: 0.6707
Epoch 102/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8835 - mean_squared_error: 0.8835 - mean_absolute_error: 0.6711
Epoch 103/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8813 - mean_squared_error: 0.8813 - mean_absolute_error: 0.6697
Epoch 104/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8798 - mean_squared_error: 0.8798 - mean_absolute_error: 0.6692
Epoch 105/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8797 - mean_squared_error: 0.8797 - mean_absolute_error: 0.6697
Epoch 106/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8782 - mean_squared_error: 0.8782 - mean_absolute_error: 0.6685
Epoch 107/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8785 - mean_squared_error: 0.8785 - mean_absolute_error: 0.6690
Epoch 108/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8802 - mean_squared_error: 0.8802 - mean_absolute_error: 0.6692
Epoch 109/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8782 - mean_squared_error: 0.8782 - mean_absolute_error: 0.6691
Epoch 110/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8788 - mean_squared_error: 0.8788 - mean_absolute_error: 0.6695
Epoch 111/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8793 - mean_squared_error: 0.8793 - mean_absolute_error: 0.6690
Epoch 112/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8787 - mean_squared_error: 0.8787 - mean_absolute_error: 0.6688
Epoch 113/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8775 - mean_squared_error: 0.8775 - mean_absolute_error: 0.6686
Epoch 114/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8781 - mean_squared_error: 0.8781 - mean_absolute_error: 0.6686
Epoch 115/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8789 - mean_squared_error: 0.8789 - mean_absolute_error: 0.6683
Epoch 116/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8762 - mean_squared_error: 0.8762 - mean_absolute_error: 0.6680
Epoch 117/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8756 - mean_squared_error: 0.8756 - mean_absolute_error: 0.6670
Epoch 118/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8754 - mean_squared_error: 0.8754 - mean_absolute_error: 0.6669
Epoch 119/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8769 - mean_squared_error: 0.8769 - mean_absolute_error: 0.6682
Epoch 120/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8749 - mean_squared_error: 0.8749 - mean_absolute_error: 0.6671
Epoch 121/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8742 - mean_squared_error: 0.8742 - mean_absolute_error: 0.6664
Epoch 122/300
```

```
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8733 - mean_squared_error: 0.8733 - mean_absolute_error: 0.6672
Epoch 123/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8758 - mean_squared_error: 0.8758 - mean_absolute_error: 0.6674
Epoch 124/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8721 - mean_squared_error: 0.8721 - mean_absolute_error: 0.6659
Epoch 125/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8733 - mean_squared_error: 0.8733 - mean_absolute_error: 0.6665
Epoch 126/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8705 - mean_squared_error: 0.8705 - mean_absolute_error: 0.6654
Epoch 127/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8702 - mean_squared_error: 0.8702 - mean_absolute_error: 0.6655
Epoch 128/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8707 - mean_squared_error: 0.8707 - mean_absolute_error: 0.6648
Epoch 129/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8720 - mean_squared_error: 0.8720 - mean_absolute_error: 0.6662
Epoch 130/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8717 - mean_squared_error: 0.8717 - mean_absolute_error: 0.6658
Epoch 131/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8690 - mean_squared_error: 0.8690 - mean_absolute_error: 0.6640
Epoch 132/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8717 - mean_squared_error: 0.8717 - mean_absolute_error: 0.6662
Epoch 133/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8707 - mean_squared_error: 0.8707 - mean_absolute_error: 0.6648
Epoch 134/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8703 - mean_squared_error: 0.8703 - mean_absolute_error: 0.6656
Epoch 135/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8693 - mean_squared_error: 0.8693 - mean_absolute_error: 0.6649
Epoch 136/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8681 - mean_squared_error: 0.8681 - mean_absolute_error: 0.6639
Epoch 137/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8691 - mean_squared_error: 0.8691 - mean_absolute_error: 0.6641
Epoch 138/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8661 - mean_squared_error: 0.8661 - mean_absolute_error: 0.6634
Epoch 139/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8697 - mean_squared_error: 0.8697 - mean_absolute_error: 0.6638
Epoch 140/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8661 - mean_squared_error: 0.8661 - mean_absolute_error: 0.6640
Epoch 141/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8681 - mean_squared_error: 0.8681 - mean_absolute_error: 0.6648
Epoch 142/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8668 - mean_squared_error: 0.8668 - mean_absolute_error: 0.6633
Epoch 143/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8667 - mean_squared_error: 0.8667 - mean_absolute_error: 0.6637
Epoch 144/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8662 - mean_squared_error: 0.8662 - mean_absolute_error: 0.6634
Epoch 145/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8665 - mean_squared_error: 0.8665 - mean_absolute_error: 0.6629
Epoch 146/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8639 - mean_squared_error: 0.8639 - mean_absolute_error: 0.6628
Epoch 147/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8634 - mean_squared_error: 0.8634 - mean_absolute_error: 0.6621
Epoch 148/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8670 - mean_squared_error: 0.8670 - mean_absolute_error: 0.6635
Epoch 149/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8641 - mean_squared_error: 0.8641 - mean_absolute_error: 0.6615
Epoch 150/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8641 - mean_squared_error: 0.8641 - mean_absolute_error: 0.6623
Epoch 151/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8642 - mean_squared_error: 0.8642 - mean_absolute_error: 0.6623
Epoch 152/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8630 - mean_squared_error: 0.8630 - mean_absolute_error: 0.6618
Epoch 153/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8641 - mean_squared_error: 0.8641 - mean_absolute_error: 0.6615
Epoch 154/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8625 - mean_squared_error: 0.8625 - mean_absolute_error: 0.6612
Epoch 155/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8643 - mean_squared_error: 0.8643 - mean_absolute_error: 0.6622
Epoch 156/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8647 - mean_squared_error: 0.8647 - mean_absolute_error: 0.6619
Epoch 157/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8625 - mean_squared_error: 0.8625 - mean_absolute_error: 0.6623
Epoch 158/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8635 - mean_squared_error: 0.8635 - mean_absolute_error: 0.6614
Epoch 159/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8626 - mean_squared_error: 0.8626 - mean_absolute_error: 0.6612
Epoch 160/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8628 - mean_squared_error: 0.8628 - mean_absolute_error: 0.6620
Epoch 161/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8586 - mean_squared_error: 0.8586 - mean_absolute_error: 0.6598
Epoch 162/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8603 - mean_squared_error: 0.8603 - mean_absolute_error: 0.6608
Epoch 163/300
2178/2178 [==============================] - 8s 3ms/step - loss: 0.8579 - mean_squared_error: 0.8579 - mean_absolute_error: 0.6600
Epoch 164/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8604 - mean_squared_error: 0.8604 - mean_absolute_error: 0.6610
Epoch 165/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8587 - mean_squared_error: 0.8587 - mean_absolute_error: 0.6596
Epoch 166/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8587 - mean_squared_error: 0.8587 - mean_absolute_error: 0.6600
Epoch 167/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8576 - mean_squared_error: 0.8576 - mean_absolute_error: 0.6586
Epoch 168/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8594 - mean_squared_error: 0.8594 - mean_absolute_error: 0.6592
Epoch 169/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8580 - mean_squared_error: 0.8580 - mean_absolute_error: 0.6594
Epoch 170/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8586 - mean_squared_error: 0.8586 - mean_absolute_error: 0.6596
Epoch 171/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8575 - mean_squared_error: 0.8575 - mean_absolute_error: 0.6588
Epoch 172/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8575 - mean_squared_error: 0.8575 - mean_absolute_error: 0.6598
Epoch 173/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8575 - mean_squared_error: 0.8575 - mean_absolute_error: 0.6587
Epoch 174/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8580 - mean_squared_error: 0.8580 - mean_absolute_error: 0.6595
Epoch 175/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8579 - mean_squared_error: 0.8579 - mean_absolute_error: 0.6585
Epoch 176/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8570 - mean_squared_error: 0.8570 - mean_absolute_error: 0.6586
Epoch 177/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8571 - mean_squared_error: 0.8571 - mean_absolute_error: 0.6591
Epoch 178/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8558 - mean_squared_error: 0.8558 - mean_absolute_error: 0.6581
Epoch 179/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8541 - mean_squared_error: 0.8541 - mean_absolute_error: 0.6575
Epoch 180/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8551 - mean_squared_error: 0.8551 - mean_absolute_error: 0.6575
Epoch 181/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8559 - mean_squared_error: 0.8559 - mean_absolute_error: 0.6593
Epoch 182/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8537 - mean_squared_error: 0.8537 - mean_absolute_error: 0.6568
Epoch 183/300
```

```
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8525 - mean_squared_error: 0.8525 - mean_absolute_error: 0.6571
Epoch 184/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8534 - mean_squared_error: 0.8534 - mean_absolute_error: 0.6567
Epoch 185/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8539 - mean_squared_error: 0.8539 - mean_absolute_error: 0.6580
Epoch 186/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8555 - mean_squared_error: 0.8555 - mean_absolute_error: 0.6579
Epoch 187/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8504 - mean_squared_error: 0.8504 - mean_absolute_error: 0.6556
Epoch 188/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8562 - mean_squared_error: 0.8562 - mean_absolute_error: 0.6584
Epoch 189/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8546 - mean_squared_error: 0.8546 - mean_absolute_error: 0.6574
Epoch 190/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8516 - mean_squared_error: 0.8516 - mean_absolute_error: 0.6566
Epoch 191/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8532 - mean_squared_error: 0.8532 - mean_absolute_error: 0.6574
Epoch 192/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8536 - mean_squared_error: 0.8536 - mean_absolute_error: 0.6569
Epoch 193/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8504 - mean_squared_error: 0.8504 - mean_absolute_error: 0.6554
Epoch 194/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8503 - mean_squared_error: 0.8503 - mean_absolute_error: 0.6560
Epoch 195/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8529 - mean_squared_error: 0.8529 - mean_absolute_error: 0.6580
Epoch 196/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8522 - mean_squared_error: 0.8522 - mean_absolute_error: 0.6562
Epoch 197/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8503 - mean_squared_error: 0.8503 - mean_absolute_error: 0.6560
Epoch 198/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8524 - mean_squared_error: 0.8524 - mean_absolute_error: 0.6566
Epoch 199/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8480 - mean_squared_error: 0.8480 - mean_absolute_error: 0.6555
Epoch 200/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8498 - mean_squared_error: 0.8498 - mean_absolute_error: 0.6552
Epoch 201/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8503 - mean_squared_error: 0.8503 - mean_absolute_error: 0.6553
Epoch 202/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8488 - mean_squared_error: 0.8488 - mean_absolute_error: 0.6548
Epoch 203/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8494 - mean_squared_error: 0.8494 - mean_absolute_error: 0.6558
Epoch 204/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8516 - mean_squared_error: 0.8516 - mean_absolute_error: 0.6568
Epoch 205/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8479 - mean_squared_error: 0.8479 - mean_absolute_error: 0.6542
Epoch 206/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8478 - mean_squared_error: 0.8478 - mean_absolute_error: 0.6547
Epoch 207/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8486 - mean_squared_error: 0.8486 - mean_absolute_error: 0.6548
Epoch 208/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8490 - mean_squared_error: 0.8490 - mean_absolute_error: 0.6543
Epoch 209/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8438 - mean_squared_error: 0.8438 - mean_absolute_error: 0.6532
Epoch 210/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8473 - mean_squared_error: 0.8473 - mean_absolute_error: 0.6544
Epoch 211/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8471 - mean_squared_error: 0.8471 - mean_absolute_error: 0.6530
Epoch 212/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8467 - mean_squared_error: 0.8467 - mean_absolute_error: 0.6533
Epoch 213/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8468 - mean_squared_error: 0.8468 - mean_absolute_error: 0.6534
Epoch 214/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8456 - mean_squared_error: 0.8456 - mean_absolute_error: 0.6537
Epoch 215/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8465 - mean_squared_error: 0.8465 - mean_absolute_error: 0.6543
Epoch 216/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8459 - mean_squared_error: 0.8459 - mean_absolute_error: 0.6540
Epoch 217/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8455 - mean_squared_error: 0.8455 - mean_absolute_error: 0.6542
Epoch 218/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8466 - mean_squared_error: 0.8466 - mean_absolute_error: 0.6527
Epoch 219/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8445 - mean_squared_error: 0.8445 - mean_absolute_error: 0.6522
Epoch 220/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8449 - mean_squared_error: 0.8449 - mean_absolute_error: 0.6529
Epoch 221/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8453 - mean_squared_error: 0.8453 - mean_absolute_error: 0.6531
Epoch 222/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8429 - mean_squared_error: 0.8429 - mean_absolute_error: 0.6518
Epoch 223/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8428 - mean_squared_error: 0.8428 - mean_absolute_error: 0.6512
Epoch 224/300
2178/2178 [==============================] - 10s 5ms/step - loss: 0.8433 - mean_squared_error: 0.8433 - mean_absolute_error: 0.6515
Epoch 225/300
2178/2178 [==============================] - 10s 5ms/step - loss: 0.8434 - mean_squared_error: 0.8434 - mean_absolute_error: 0.6522
Epoch 226/300
2178/2178 [==============================] - 10s 5ms/step - loss: 0.8432 - mean_squared_error: 0.8432 - mean_absolute_error: 0.6523
Epoch 227/300
2178/2178 [==============================] - 10s 5ms/step - loss: 0.8433 - mean_squared_error: 0.8433 - mean_absolute_error: 0.6515
Epoch 228/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8434 - mean_squared_error: 0.8434 - mean_absolute_error: 0.6532
Epoch 229/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8412 - mean_squared_error: 0.8412 - mean_absolute_error: 0.6515
Epoch 230/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8408 - mean_squared_error: 0.8408 - mean_absolute_error: 0.6509
Epoch 231/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8437 - mean_squared_error: 0.8437 - mean_absolute_error: 0.6531
Epoch 232/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8400 - mean_squared_error: 0.8400 - mean_absolute_error: 0.6506
Epoch 233/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8414 - mean_squared_error: 0.8414 - mean_absolute_error: 0.6512
Epoch 234/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8439 - mean_squared_error: 0.8439 - mean_absolute_error: 0.6521
Epoch 235/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8398 - mean_squared_error: 0.8398 - mean_absolute_error: 0.6502
Epoch 236/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8403 - mean_squared_error: 0.8403 - mean_absolute_error: 0.6508
Epoch 237/300
2178/2178 [==============================] - 10s 5ms/step - loss: 0.8406 - mean_squared_error: 0.8406 - mean_absolute_error: 0.6510
Epoch 238/300
2178/2178 [==============================] - 11s 5ms/step - loss: 0.8419 - mean_squared_error: 0.8419 - mean_absolute_error: 0.6516
Epoch 239/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8411 - mean_squared_error: 0.8411 - mean_absolute_error: 0.6510
Epoch 240/300
2178/2178 [==============================] - 10s 5ms/step - loss: 0.8406 - mean_squared_error: 0.8406 - mean_absolute_error: 0.6503
Epoch 241/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8390 - mean_squared_error: 0.8390 - mean_absolute_error: 0.6495
Epoch 242/300
2178/2178 [==============================] - 10s 4ms/step - loss: 0.8386 - mean_squared_error: 0.8386 - mean_absolute_error: 0.6497
Epoch 243/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8415 - mean_squared_error: 0.8415 - mean_absolute_error: 0.6513
Epoch 244/300
```

```
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8413 - mean_squared_error: 0.8413 - mean_absolute_error: 0.6512
Epoch 245/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8398 - mean_squared_error: 0.8398 - mean_absolute_error: 0.6502
Epoch 246/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8392 - mean_squared_error: 0.8392 - mean_absolute_error: 0.6502
Epoch 247/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8397 - mean_squared_error: 0.8397 - mean_absolute_error: 0.6506
Epoch 248/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8378 - mean_squared_error: 0.8378 - mean_absolute_error: 0.6495
Epoch 249/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8369 - mean_squared_error: 0.8369 - mean_absolute_error: 0.6497
Epoch 250/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8375 - mean_squared_error: 0.8375 - mean_absolute_error: 0.6489
Epoch 251/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8392 - mean_squared_error: 0.8392 - mean_absolute_error: 0.6506
Epoch 252/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8369 - mean_squared_error: 0.8369 - mean_absolute_error: 0.6490
Epoch 253/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8378 - mean_squared_error: 0.8378 - mean_absolute_error: 0.6496
Epoch 254/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8358 - mean_squared_error: 0.8358 - mean_absolute_error: 0.6486
Epoch 255/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8368 - mean_squared_error: 0.8368 - mean_absolute_error: 0.6484
Epoch 256/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8389 - mean_squared_error: 0.8389 - mean_absolute_error: 0.6506
Epoch 257/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8379 - mean_squared_error: 0.8379 - mean_absolute_error: 0.6494
Epoch 258/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8374 - mean_squared_error: 0.8374 - mean_absolute_error: 0.6491
Epoch 259/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8357 - mean_squared_error: 0.8357 - mean_absolute_error: 0.6480
Epoch 260/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8357 - mean_squared_error: 0.8357 - mean_absolute_error: 0.6488
Epoch 261/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8339 - mean_squared_error: 0.8339 - mean_absolute_error: 0.6474
Epoch 262/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8356 - mean_squared_error: 0.8356 - mean_absolute_error: 0.6480
Epoch 263/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8331 - mean_squared_error: 0.8331 - mean_absolute_error: 0.6467
Epoch 264/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8353 - mean_squared_error: 0.8353 - mean_absolute_error: 0.6480
Epoch 265/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8365 - mean_squared_error: 0.8365 - mean_absolute_error: 0.6489
Epoch 266/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8337 - mean_squared_error: 0.8337 - mean_absolute_error: 0.6480
Epoch 267/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8322 - mean_squared_error: 0.8322 - mean_absolute_error: 0.6466
Epoch 268/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8336 - mean_squared_error: 0.8336 - mean_absolute_error: 0.6467
Epoch 269/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8359 - mean_squared_error: 0.8359 - mean_absolute_error: 0.6487
Epoch 270/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8323 - mean_squared_error: 0.8323 - mean_absolute_error: 0.6474
Epoch 271/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8347 - mean_squared_error: 0.8347 - mean_absolute_error: 0.6477
Epoch 272/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8333 - mean_squared_error: 0.8333 - mean_absolute_error: 0.6472
Epoch 273/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8332 - mean_squared_error: 0.8332 - mean_absolute_error: 0.6477
Epoch 274/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8347 - mean_squared_error: 0.8347 - mean_absolute_error: 0.6476
Epoch 275/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8354 - mean_squared_error: 0.8354 - mean_absolute_error: 0.6484
Epoch 276/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8306 - mean_squared_error: 0.8306 - mean_absolute_error: 0.6456
Epoch 277/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8325 - mean_squared_error: 0.8325 - mean_absolute_error: 0.6473
Epoch 278/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8323 - mean_squared_error: 0.8323 - mean_absolute_error: 0.6463
Epoch 279/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8291 - mean_squared_error: 0.8291 - mean_absolute_error: 0.6460
Epoch 280/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8335 - mean_squared_error: 0.8335 - mean_absolute_error: 0.6476
Epoch 281/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8327 - mean_squared_error: 0.8327 - mean_absolute_error: 0.6466
Epoch 282/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8295 - mean_squared_error: 0.8295 - mean_absolute_error: 0.6455
Epoch 283/300
2178/2178 [==============================] - 8s 4ms/step - loss: 0.8303 - mean_squared_error: 0.8303 - mean_absolute_error: 0.6456
Epoch 284/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8280 - mean_squared_error: 0.8280 - mean_absolute_error: 0.6446
Epoch 285/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8301 - mean_squared_error: 0.8301 - mean_absolute_error: 0.6462
Epoch 286/300
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8278 - mean_squared_error: 0.8278 - mean_absolute_error: 0.6443
```

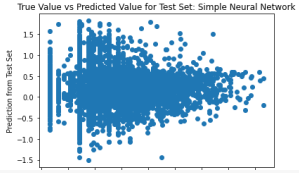
```
1
```




```
2178/2178 [==============================] - 9s 4ms/step - loss: 0.8281 - mean_squared_error: 0.8281 - mean_absolute_error: 0.6454
```

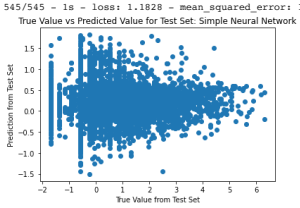
```
# 60 epochs, 231, 50, 50, 50 ,50
model.evaluate(X_test, y_test, verbose=2)

test_predictions = model.predict(X_test)

fig = plt.figure()
ax1 = fig.add_subplot()
ax1.set_xlabel("True Value from Test Set")
ax1.set_ylabel('Prediction from Test Set')
ax1.set_title('True Value vs Predicted Value for Test Set: Simple Neural Network')
ax1.scatter(y_test, test_predictions)
plt.show()
```

```
545/545 - 1s - loss: 1.1828 - mean_squared_error: 1.1828 - mean_absolute_error: 0.8530 - 816ms/epoch - 1ms/step
```



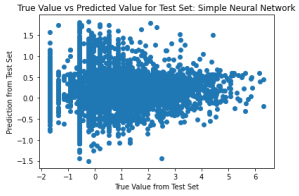

```
model.evaluate(X_test, y_test, verbose=2)

test_predictions = model.predict(X_test)

fig = plt.figure()
ax1 = fig.add_subplot()
ax1.set_xlabel("True Value from Test Set")
ax1.set_ylabel('Prediction from Test Set')
ax1.set_title('True Value vs Predicted Value for Test Set: Simple Neural Network')
ax1.scatter(y_test, test_predictions)
plt.show()
```

```
545/545 - 1s - loss: 1.1828 - mean_squared_error: 1.1828 - mean_absolute_error: 0.8530 - 824ms/epoch - 2ms/step
```




```
# model = tf.keras.models.Sequential([Dense(231, activation="relu"),
#                                     Dropout(0.2),
#                                     Dense(50, activation="relu"),
#                                     Dense(25, activation="relu"),
#                                     Dense(1)])

model.evaluate(X_test, y_test, verbose=2)

test_predictions = model.predict(X_test)

fig = plt.figure()
ax1 = fig.add_subplot()
ax1.set_xlabel("True Value from Test Set")
ax1.set_ylabel('Prediction from Test Set')
ax1.set_title('True Value vs Predicted Value for Test Set: Simple Neural Network')
ax1.scatter(y_test, test_predictions)
plt.show()
```

```
545/545 - 1s - loss: 1.1828 - mean_squared_error: 1.1828 - mean_absolute_error: 0.8530 - 881ms/epoch - 2ms/step
```




```
# model = tf.keras.models.Sequential([Dense(231, activation="relu"),
#                                     Dropout(0.2),
#                                     Dense(50, activation="relu"),
#                                     Dense(25, activation="relu"),
#                                     Dense(1)])

# loss_function = MeanSquaredError()
# model.compile(optimizer="adam", loss=loss_function, metrics=["mean_squared_error", "mean_absolute_error"])
# model.fit(X_train, y_train, epochs = 1000)

model.evaluate(X_test, y_test, verbose=2)

test_predictions = model.predict(X_test)

fig = plt.figure()
ax1 = fig.add_subplot()
ax1.set_xlabel("True Value from Test Set")
ax1.set_ylabel('Prediction from Test Set')
ax1.set_title('True Value vs Predicted Value for Test Set: Simple Neural Network')
ax1.scatter(y_test, test_predictions)
plt.show()
```

```
545/545 - 1s - loss: 1.1828 - mean_squared_error: 1.1828 - mean_absolute_error: 0.8530 - 783ms/epoch - 1ms/step
```

Neural Network Classifier

First, we want to remove the status cause code from out dataframe, and convert it to a one-hot encoding.

```
# cleaned.info(verbose=True)
cause_codes = cleaned["STAT_CAUSE_CODE"]

features = primary_station_df.copy()

if "STAT_CAUSE_CODE" in features:
  del features["STAT_CAUSE_CODE"]
if "STAT_CAUSE_DESCR" in features:
  del features["STAT_CAUSE_DESCR"]

# encoder = OneHotEncoder()
# cause_codes = np.array(cause_codes)
# cause_codes = cause_codes.reshape(-1, 1)
# cause_codes_encoded = encoder.fit_transform(cause_codes).toarray() * 1.0
# print(cause_codes_encoded)
```

```
import tensorflow as tf
from tensorflow.keras.layers import *
from tensorflow.keras.losses import SparseCategoricalCrossentropy

def run_full_classification_neural_net(features, predictors, epochs = 10, model=None):
  X_train, X_test, y_train, y_test = train_test_split(np.array(X), Y, test_size=0.2, random_state=2020)
  # Classification into 12 different fire cause code classes.
  if model is None:
    model = tf.keras.models.Sequential([Dense(233, activation="relu"),
                                        Dense(25, activation="relu"),
                                        Dropout(0.2),
                                        Dense(25, activation="relu"),
                                        Dense(25, activation="relu"),
                                        Dense(14, activation="relu")])


  # Make a quick untrained prediction
  predictions = model(X_train[:1]).numpy()
  print(predictions)
  print(tf.nn.softmax(predictions).numpy()) # Probability of each class

  loss_function = SparseCategoricalCrossentropy(from_logits=True)
  model.compile(optimizer="adam", loss=loss_function, metrics=["accuracy"])
  print(model.summary())
  model.fit(X_train, y_train, epochs = epochs)
  print("EVALUATION:")
  print(model.evaluate(X_test, y_test))

```

```
# These are the variables to learn against
X = features
Y = cause_codes

run_full_classification_neural_net(X, Y, epochs = 200)
```

```
[[0.48183015 0.53273875 0.         1.0898234  0.14692128 0.
  0.         0.35303557 0.54752827 0.18044797 0.2152055  0.1799257
  0.13928226 0.00763759]]
[[0.08345924 0.08781805 0.05154876 0.15329307 0.05970702 0.05154876
  0.05154876 0.07337357 0.08912648 0.06174273 0.06392648 0.06171049
  0.05925265 0.05194398]]
Model: "sequential_14"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_65 (Dense)            (1, 233)                  54522

 dense_66 (Dense)            (1, 400)                  93600

 dropout_11 (Dropout)        (1, 400)                  0

 dense_67 (Dense)            (1, 400)                  160400

 dense_68 (Dense)            (1, 400)                  160400

 dense_69 (Dense)            (1, 14)                   5614

=================================================================
Total params: 474,536
Trainable params: 474,536
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/60
2178/2178 [==============================] - 22s 10ms/step - loss: 2.1051 - accuracy: 0.3800
Epoch 2/60
2178/2178 [==============================] - 21s 10ms/step - loss: 1.7774 - accuracy: 0.4077
Epoch 3/60
2178/2178 [==============================] - 21s 10ms/step - loss: 1.8005 - accuracy: 0.4172
Epoch 4/60
2178/2178 [==============================] - 21s 10ms/step - loss: 1.7298 - accuracy: 0.4244
Epoch 5/60
2178/2178 [==============================] - 20s 9ms/step - loss: 1.7021 - accuracy: 0.4312
Epoch 6/60
2178/2178 [==============================] - 21s 10ms/step - loss: 1.8003 - accuracy: 0.4348
Epoch 7/60
2178/2178 [==============================] - 22s 10ms/step - loss: 1.6759 - accuracy: 0.4395
Epoch 8/60
2178/2178 [==============================] - 22s 10ms/step - loss: 1.6603 - accuracy: 0.4434
Epoch 9/60
2178/2178 [==============================] - 21s 9ms/step - loss: 1.6528 - accuracy: 0.4443
Epoch 10/60
2178/2178 [==============================] - 22s 10ms/step - loss: 1.6422 - accuracy: 0.4470
```

```
Epoch 11/60
2178/2178 [==============================] - 21s 10ms/step - loss: 1.6381 - accuracy: 0.4508
Epoch 12/60
2178/2178 [==============================] - 21s 10ms/step - loss: 1.6242 - accuracy: 0.4548
Epoch 13/60
2178/2178 [==============================] - 20s 9ms/step - loss: 1.6264 - accuracy: 0.4560
Epoch 14/60
2178/2178 [==============================] - 20s 9ms/step - loss: 1.6141 - accuracy: 0.4574
Epoch 15/60
2178/2178 [==============================] - 19s 9ms/step - loss: 1.6028 - accuracy: 0.4595
```

Next, we will try to train and test ignoring fires that have unknown cause.

```
cause_codes = cleaned["STAT_CAUSE_CODE"]
features = primary_station_df.copy()

known_cause_index = (cause_codes != 9) & (cause_codes != 13)

known_cause_codes = cause_codes[known_cause_index]
known_features = features[known_cause_index]

run_full_classification_neural_net(known_features, known_cause_codes, epochs = 100)
```

```
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5643 - accuracy: 0.4726
Epoch 48/100
2178/2178 [==============================] - 5s 3ms/step - loss: 1.5654 - accuracy: 0.4697
Epoch 49/100
2178/2178 [==============================] - 5s 3ms/step - loss: 1.5604 - accuracy: 0.4725
Epoch 50/100
2178/2178 [==============================] - 5s 3ms/step - loss: 1.5594 - accuracy: 0.4736
Epoch 51/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5583 - accuracy: 0.4721
Epoch 52/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5768 - accuracy: 0.4739
Epoch 53/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5550 - accuracy: 0.4722
Epoch 54/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5539 - accuracy: 0.4741
Epoch 55/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5543 - accuracy: 0.4750
Epoch 56/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5515 - accuracy: 0.4753
Epoch 57/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5525 - accuracy: 0.4741
Epoch 58/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5505 - accuracy: 0.4755
Epoch 59/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5485 - accuracy: 0.4756
Epoch 60/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5464 - accuracy: 0.4766
Epoch 61/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5464 - accuracy: 0.4771
Epoch 62/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5483 - accuracy: 0.4763
Epoch 63/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5442 - accuracy: 0.4771
Epoch 64/100
2178/2178 [==============================] - 6s 3ms/step - loss: 1.5412 - accuracy: 0.4779
Epoch 65/100
2178/2178 [==============================] - 5s 3ms/step - loss: 1.5757 - accuracy: 0.4762
Epoch 66/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5442 - accuracy: 0.4785
Epoch 67/100
2178/2178 [==============================] - 5s 3ms/step - loss: 1.5395 - accuracy: 0.4782
Epoch 68/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5395 - accuracy: 0.4788
Epoch 69/100
2178/2178 [==============================] - 5s 3ms/step - loss: 1.5386 - accuracy: 0.4796
Epoch 70/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5341 - accuracy: 0.4807
Epoch 71/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5378 - accuracy: 0.4795
Epoch 72/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5345 - accuracy: 0.4799
Epoch 73/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5339 - accuracy: 0.4796
Epoch 74/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5393 - accuracy: 0.4796
Epoch 75/100
2178/2178 [==============================] - 5s 2ms/step - loss: 1.5310 - accuracy: 0.4789
Epoch 76/100
```

```
# trying a more complex architecture
model = tf.keras.models.Sequential([Dense(233, activation="relu"),
                                    Dense(233, activation="relu"),
                                    Dropout(0.1),
                                    Dense(233, activation="relu"),
                                    Dropout(0.2),
                                    Dense(14, activation="relu")])
run_full_classification_neural_net(known_features, known_cause_codes, epochs = 400, model=model)
```

```
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4925 - accuracy: 0.4967
Epoch 44/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.5050 - accuracy: 0.4910
Epoch 45/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.5036 - accuracy: 0.4923
Epoch 46/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4985 - accuracy: 0.4920
Epoch 47/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4946 - accuracy: 0.4937
Epoch 48/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4897 - accuracy: 0.4966
Epoch 49/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4856 - accuracy: 0.4970
Epoch 50/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4789 - accuracy: 0.4988
Epoch 51/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4899 - accuracy: 0.4942
Epoch 52/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4813 - accuracy: 0.4954
Epoch 53/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4756 - accuracy: 0.5033
Epoch 54/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4689 - accuracy: 0.4998
Epoch 55/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4669 - accuracy: 0.5036
Epoch 56/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4688 - accuracy: 0.5032
Epoch 57/400
```

```
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4708 - accuracy: 0.5013
Epoch 58/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4628 - accuracy: 0.5045
Epoch 59/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4661 - accuracy: 0.5036
Epoch 60/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4620 - accuracy: 0.5032
Epoch 61/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4615 - accuracy: 0.5052
Epoch 62/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4555 - accuracy: 0.5043
Epoch 63/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4522 - accuracy: 0.5085
Epoch 64/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4488 - accuracy: 0.5075
Epoch 65/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4522 - accuracy: 0.5065
Epoch 66/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4502 - accuracy: 0.5084
Epoch 67/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4756 - accuracy: 0.5024
Epoch 68/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4486 - accuracy: 0.5082
Epoch 69/400
2178/2178 [==============================] - 9s 4ms/step - loss: 1.4497 - accuracy: 0.5062
Epoch 70/400
2178/2178 [==============================] - 9s 4ms/step - loss: 1.4512 - accuracy: 0.5073
Epoch 71/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4506 - accuracy: 0.5057
Epoch 72/400
2178/2178 [==============================] - 8s 4ms/step - loss: 1.4480 - accuracy: 0.5053
```

```
# probability_model = tf.keras.Sequential([
#   model,
#   tf.keras.layers.Softmax()
# ])
```

```
# probability_model(X_test[:5])
```

```
<tf.Tensor: shape=(5, 14), dtype=float32, numpy=
array([[3.29549704e-03, 1.66717973e-02, 2.67233163e-01, 3.45799029e-02,
        4.54233326e-02, 6.08601756e-02, 3.29549704e-03, 9.02992263e-02,
        4.02569845e-02, 3.03870469e-01, 3.29549704e-03, 5.01898900e-02,
        3.29549704e-03, 7.74330422e-02],
       [6.72030510e-05, 9.49687421e-01, 8.00500251e-03, 3.49900965e-03,
        7.98075087e-03, 2.22626817e-03, 6.72030510e-05, 2.14828923e-03,
        1.18853315e-03, 2.42914725e-02, 6.72030510e-05, 6.72030510e-05,
        6.72030510e-05, 6.37105783e-04],
       [1.72957755e-03, 1.41575588e-02, 1.70311946e-02, 1.72957755e-03,
        5.11530787e-02, 8.17846507e-02, 1.72957755e-03, 7.08773196e-01,
        7.14560365e-03, 9.76177081e-02, 1.72957755e-03, 1.72957755e-03,
        1.72957755e-03, 1.19594922e-02],
       [7.89784535e-04, 5.61121881e-01, 4.12679575e-02, 3.99304330e-02,
        1.66325241e-01, 1.64578483e-02, 7.89784535e-04, 1.33630764e-02,
        1.30247315e-02, 1.40769288e-01, 7.89784535e-04, 7.89784535e-04,
        7.89784535e-04, 3.79060791e-03],
       [1.20075094e-03, 1.16253376e-01, 1.08798437e-01, 5.24045490e-02,
        2.79174715e-01, 3.31705734e-02, 1.20075094e-03, 7.25342333e-02,
        4.18527313e-02, 2.71965116e-01, 1.20075094e-03, 1.56444055e-03,
        1.20075094e-03, 1.74788609e-02]], dtype=float32)>
```