In [1]:
```
#IMPORT REQUIRED LIBRARIES
import pandas as pd #data manipulation/analysis
import numpy as np #numerical operations
import matplotlib.pyplot as plt #plotting
import seaborn as sns #statisticals visualizations
from sklearn.model_selection import train_test_split #training/testing sets
from sklearn.ensemble import RandomForestRegressor  # Use regressor instead of classifier
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import joblib   #saving/loading trained models
```

In [2]:
```
#LOAD DATASETS
#these data sets contain features and target data we need
#for our project/model
weather_data = pd.read_csv("all_conditions.csv")
wildfire_data = pd.read_csv("California_Fire_Incidents_utf8.csv")
forest_fires = pd.read_csv("forestfires.csv")
```

In [3]:
```
#EXPLORATORY
#display first few rows
#examine basic info about the datasets

#Descriptive Stats
#weather dataset
print("Weather Data Summary: \n", weather_data.describe())
#wildfire dataset
print("\nWildfire Data Summary: \n", wildfire_data.describe())
#forest fire dataset
print("\nForest Fires Data Summary: \n", forest_fires.describe())
```

Weather Data Summary:

	Stn Id	ETo (in)	Precip (in)	Sol Rad (Ly/day)
count	128125.000000	128042.000000	128125.000000	128125.000000
mean	157.257686	0.157898	0.038263	458.957136
std	72.576703	0.086695	0.202978	198.349566
min	2.000000	0.000000	0.000000	0.000000
25%	99.000000	0.080000	0.000000	301.000000
50%	171.000000	0.150000	0.000000	471.000000
75%	219.000000	0.230000	0.000000	633.000000
max	262.000000	0.490000	13.610000	6618.000000

	Avg Vap Pres (mBars)	Max Air Temp (F)	Min Air Temp (F)
count	128125.000000	128122.000000	128124.000000
mean	11.285094	75.279722	48.495665
std	4.105851	14.742913	11.430017
min	0.000000	24.900000	-5.000000
25%	8.400000	64.000000	41.100000
50%	11.200000	74.300000	48.800000
75%	14.000000	86.100000	55.800000
max	39.700000	123.700000	93.400000

	Avg Air Temp (F)	Max Rel Hum (%)	Min Rel Hum (%)	Avg Rel Hum (%)
count	128120.000000	128125.000000	128125.000000	128112.000000
mean	61.185965	85.018100	40.166431	61.374438
std	12.386885	16.407883	20.678380	20.273168
min	13.000000	0.000000	0.000000	0.000000
25%	52.800000	78.000000	23.000000	46.000000
50%	60.100000	91.000000	37.000000	63.000000
75%	69.200000	97.000000	55.000000	78.000000
max	106.500000	100.000000	100.000000	100.000000

	Dew Point (F)	Avg Wind Speed (mph)	Wind Run (miles)
count	128112.000000	128125.000000	128125.000000
mean	45.939090	4.314043	103.532969
std	10.927562	2.041915	48.999902
min	-74.300000	0.700000	16.200000
25%	40.000000	3.000000	71.600000
50%	47.500000	3.800000	92.300000
75%	53.600000	5.100000	122.400000
max	82.200000	46.900000	1125.300000

	Avg Soil Temp (F)	Target
count	128105.000000	128125.000000
mean	62.799889	0.041787
std	10.672439	0.200104
min	31.500000	0.000000
25%	54.400000	0.000000
50%	62.900000	0.000000
75%	70.800000	0.000000
max	96.900000	1.000000

Wildfire Data Summary:

	AcresBurned	AirTankers	ArchiveYear	CrewsInvolved	Dozers
count	1633.000000	28.000000	1636.000000	171.000000	123.000000
mean	4589.443968	4.071429	2016.608802	11.561404	7.585366
std	27266.337722	6.399818	1.845340	14.455633	14.028616
min	0.000000	0.000000	2013.000000	0.000000	0.000000
25%	35.000000	2.000000	2015.000000	2.500000	1.000000
50%	100.000000	2.000000	2017.000000	6.000000	2.000000
75%	422.000000	4.000000	2018.000000	13.500000	5.000000
max	410203.000000	27.000000	2019.000000	82.000000	76.000000

	Engines	Fatalities	Helicopters	Injuries	Latitude

```
count  191.000000   21.000000    84.000000  120.000000  1636.000000
mean    23.565445    8.619048     5.357143    3.500000    37.203975
std     41.004424   18.529642     7.265437    3.806231   135.401380
min      0.000000    1.000000     0.000000    0.000000  -120.258000
25%      5.000000    1.000000     1.000000    1.000000    34.165891
50%     11.000000    3.000000     2.000000    3.000000    37.104065
75%     24.000000    6.000000     5.000000    4.000000    39.086808
max    256.000000   85.000000    29.000000   26.000000  5487.000000

         Longitude  PercentContained  PersonnelInvolved  StructuresDamaged  \
count  1636.000000            1633.0         204.000000          67.000000
mean   -108.082642             100.0         328.553922          67.970149
std      37.006927               0.0         521.138789         155.771975
min    -124.196290             100.0           0.000000           0.000000
25%    -121.768358             100.0          55.000000           1.000000
50%    -120.461560             100.0         151.500000           6.000000
75%    -117.474073             100.0         350.000000          49.500000
max     118.908200             100.0        3100.000000         783.000000

       StructuresDestroyed  StructuresEvacuated  StructuresThreatened  \
count           175.000000                  0.0             30.000000
mean            271.788571                  NaN            522.800000
std            1557.255963                  NaN            739.586856
min               0.000000                  NaN              0.000000
25%               1.000000                  NaN              0.000000
50%               7.000000                  NaN             45.000000
75%              41.500000                  NaN           1043.750000
max           18804.000000                  NaN           2600.000000

       WaterTenders
count    146.000000
mean       7.815068
std       12.719251
min        1.000000
25%        2.000000
50%        4.000000
75%        6.000000
max       79.000000

Forest Fires Data Summary:
                X           Y        FFMC         DMC          DC         ISI  \
count  517.000000  517.000000  517.000000  517.000000  517.000000  517.000000
mean     4.669246    4.299807   90.644681  110.872340  547.940039    9.021663
std      2.313778    1.229900    5.520111   64.046482  248.066192    4.559477
min      1.000000    2.000000   18.700000    1.100000    7.900000    0.000000
25%      3.000000    4.000000   90.200000   68.600000  437.700000    6.500000
50%      4.000000    4.000000   91.600000  108.300000  664.200000    8.400000
75%      7.000000    5.000000   92.900000  142.400000  713.900000   10.800000
max      9.000000    9.000000   96.200000  291.300000  860.600000   56.100000

             temp          RH        wind        rain         area
count  517.000000  517.000000  517.000000  517.000000   517.000000
mean    18.889168   44.288201    4.017602    0.021663    12.847292
std      5.806625   16.317469    1.791653    0.295959    63.655818
min      2.200000   15.000000    0.400000    0.000000     0.000000
25%     15.500000   33.000000    2.700000    0.000000     0.000000
50%     19.300000   42.000000    4.000000    0.000000     0.520000
75%     22.800000   53.000000    4.900000    0.000000     6.570000
max     33.300000  100.000000    9.400000    6.400000  1090.840000
```

In [4]:
```
#display first few lines of the datasets
#weather dataset
print("Weather Data Summary: \n", weather_data.head())
```

```
#wildfire dataset
print("\nWildfire Data Summary: \n", wildfire_data.head())
#forest fire dataset
print("\nForest Fires Data Summary: \n", forest_fires.head())
```

Weather Data Summary:

	Stn Id	Stn Name	CIMIS Region	Date	ETo (in)	Precip (in)
0	2	FivePoints	San Joaquin Valley	1/1/2018	0.06	0.00
1	2	FivePoints	San Joaquin Valley	1/2/2018	0.04	0.00
2	2	FivePoints	San Joaquin Valley	1/3/2018	0.04	0.00
3	2	FivePoints	San Joaquin Valley	1/4/2018	0.07	0.01
4	2	FivePoints	San Joaquin Valley	1/5/2018	0.07	0.00

	Sol Rad (Ly/day)	Avg Vap Pres (mBars)	Max Air Temp (F)	Min Air Temp (F)
0	219.0	7.3	63.4	35.3
1	127.0	7.4	59.8	37.7
2	125.0	8.4	61.1	37.3
3	219.0	11.6	69.2	48.7
4	239.0	12.7	73.8	47.5

	Avg Air Temp (F)	Max Rel Hum (%)	Min Rel Hum (%)	Avg Rel Hum (%)
0	47.8	82.0	46.0	65.0
1	47.2	80.0	52.0	67.0
2	49.9	79.0	49.0	68.0
3	56.8	94.0	52.0	74.0
4	59.8	94.0	49.0	72.0

	Dew Point (F)	Avg Wind Speed (mph)	Wind Run (miles)	Avg Soil Temp (F)
0	36.6	3.3	78.3	51.1
1	36.7	3.1	74.5	51.3
2	39.9	4.5	107.5	51.3
3	48.5	5.8	140.2	53.0
4	50.8	4.2	101.4	54.4

	Target
0	0
1	0
2	0
3	0
4	0

Wildfire Data Summary:

	AcresBurned	Active	AdminUnit
0	257314.0	False	Stanislaus National Forest/Yosemite National Park
1	30274.0	False	USFS Angeles National Forest/Los Angeles Count...
2	27531.0	False	CAL FIRE Riverside Unit / San Bernardino Natio...
3	27440.0	False	Tahoe National Forest
4	24251.0	False	Ventura County Fire/CAL FIRE

	AirTankers	ArchiveYear	CalFireIncident
0	NaN	2013	True
1	NaN	2013	True
2	NaN	2013	True
3	NaN	2013	False
4	NaN	2013	True

	CanonicalUrl
0	/incidents/2013/8/17/rim-fire/
1	/incidents/2013/5/30/powerhouse-fire/
2	/incidents/2013/7/15/mountain-fire/
3	/incidents/2013/8/10/american-fire/
4	/incidents/2013/5/2/springs-fire/

	ConditionStatement	ControlStatement
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN

```
4  Acreage has been reduced based upon more accur...              NaN

      Counties  ...                                    SearchKeywords  \
0     Tuolumne  ...  Rim Fire, Stanislaus National Forest, Yosemite...
1  Los Angeles  ...  Powerhouse Fire, May 2013, June 2013, Angeles ...
2    Riverside  ...  Mountain Fire, July 2013, Highway 243, Highway...
3       Placer  ...  American Fire, August 2013, Deadwood Ridge, Fo...
4      Ventura  ...  Springs Fire, May 2013, Highway 101, Camarillo...

                Started     Status  StructuresDamaged StructuresDestroyed  \
0  2013-08-17T15:25:00Z  Finalized               NaN                 NaN
1  2013-05-30T15:28:00Z  Finalized               NaN                 NaN
2  2013-07-15T13:43:00Z  Finalized               NaN                 NaN
3  2013-08-10T16:30:00Z  Finalized               NaN                 NaN
4  2013-05-02T07:01:00Z  Finalized               6.0                10.0

   StructuresEvacuated  StructuresThreatened  \
0                  NaN                   NaN
1                  NaN                   NaN
2                  NaN                   NaN
3                  NaN                   NaN
4                  NaN                   NaN

                               UniqueId               Updated  WaterTenders
0  5fb18d4d-213f-4d83-a179-daaf11939e78  2013-09-06T18:30:00Z           NaN
1  bf37805e-1cc2-4208-9972-753e47874c87  2013-06-08T18:30:00Z           NaN
2  a3149fec-4d48-427c-8b2c-59e8b79d59db  2013-07-30T18:00:00Z           NaN
3  8213f5c7-34fa-403b-a4bc-da2ace6e6625  2013-08-30T08:00:00Z           NaN
4  46731fb8-3350-4920-bdf7-910ac0eb715c  2013-05-11T06:30:00Z          11.0

[5 rows x 40 columns]

Forest Fires Data Summary:
    X  Y month  day  FFMC   DMC     DC  ISI  temp  RH  wind  rain  area
0  7  5   mar  fri  86.2  26.2   94.3  5.1   8.2  51   6.7   0.0   0.0
1  7  4   oct  tue  90.6  35.4  669.1  6.7  18.0  33   0.9   0.0   0.0
2  7  4   oct  sat  90.6  43.7  686.9  6.7  14.6  33   1.3   0.0   0.0
3  8  6   mar  fri  91.7  33.3   77.5  9.0   8.3  97   4.0   0.2   0.0
4  8  6   mar  sun  89.3  51.3  102.2  9.6  11.4  99   1.8   0.0   0.0
```

In [5]:
```
#check for missing values
#If critical data like temperature
#or fire occurrence is missing, it can impact predictions.
#We decide here how to handle them—whether to drop, fill, or interpolate
print("\nMissing Values in Weather Data:\n", weather_data.isnull().sum())
print("\nMissing Values in Wildfire Data:\n", wildfire_data.isnull().sum())
print("\nMissing Values in Forest Fires Data:\n", forest_fires.isnull().sum())
```

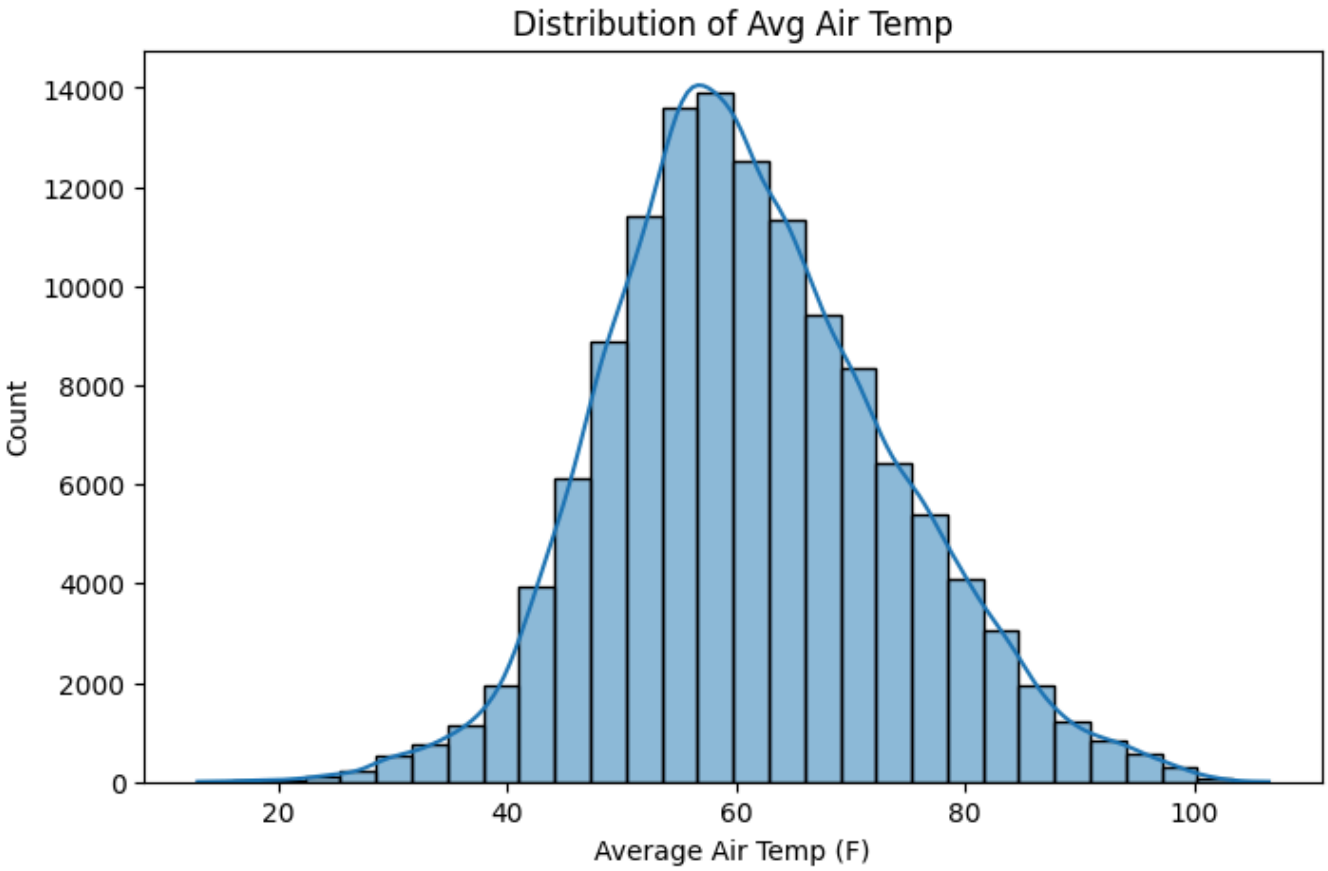
```
Missing Values in Weather Data:
 Stn Id                   0
Stn Name                  0
CIMIS Region              0
Date                      0
ETo (in)                 83
Precip (in)               0
Sol Rad (Ly/day)          0
Avg Vap Pres (mBars)      0
Max Air Temp (F)          3
Min Air Temp (F)          1
Avg Air Temp (F)          5
Max Rel Hum (%)           0
Min Rel Hum (%)           0
Avg Rel Hum (%)          13
Dew Point (F)            13
Avg Wind Speed (mph)      0
Wind Run (miles)          0
Avg Soil Temp (F)        20
Target                    0
dtype: int64

Missing Values in Wildfire Data:
 AcresBurned                3
Active                      0
AdminUnit                   0
AirTankers               1608
ArchiveYear                 0
CalFireIncident             0
CanonicalUrl                0
ConditionStatement       1352
ControlStatement         1531
Counties                    0
CountyIds                   0
CrewsInvolved            1465
Dozers                   1513
Engines                  1445
Extinguished               59
Fatalities               1615
Featured                    0
Final                       0
FuelType                 1624
Helicopters              1552
Injuries                 1516
Latitude                    0
Location                    0
Longitude                   0
MajorIncident               0
Name                        0
PercentContained            3
PersonnelInvolved        1432
Public                      0
SearchDescription          17
SearchKeywords            203
Started                     0
Status                      0
StructuresDamaged        1569
StructuresDestroyed      1461
StructuresEvacuated      1636
StructuresThreatened     1606
UniqueId                    0
Updated                     0
WaterTenders             1490
```

```
dtype: int64

Missing Values in Forest Fires Data:
 X        0
Y        0
month    0
day      0
FFMC     0
DMC      0
DC       0
ISI      0
temp     0
RH       0
wind     0
rain     0
area     0
dtype: int64
```

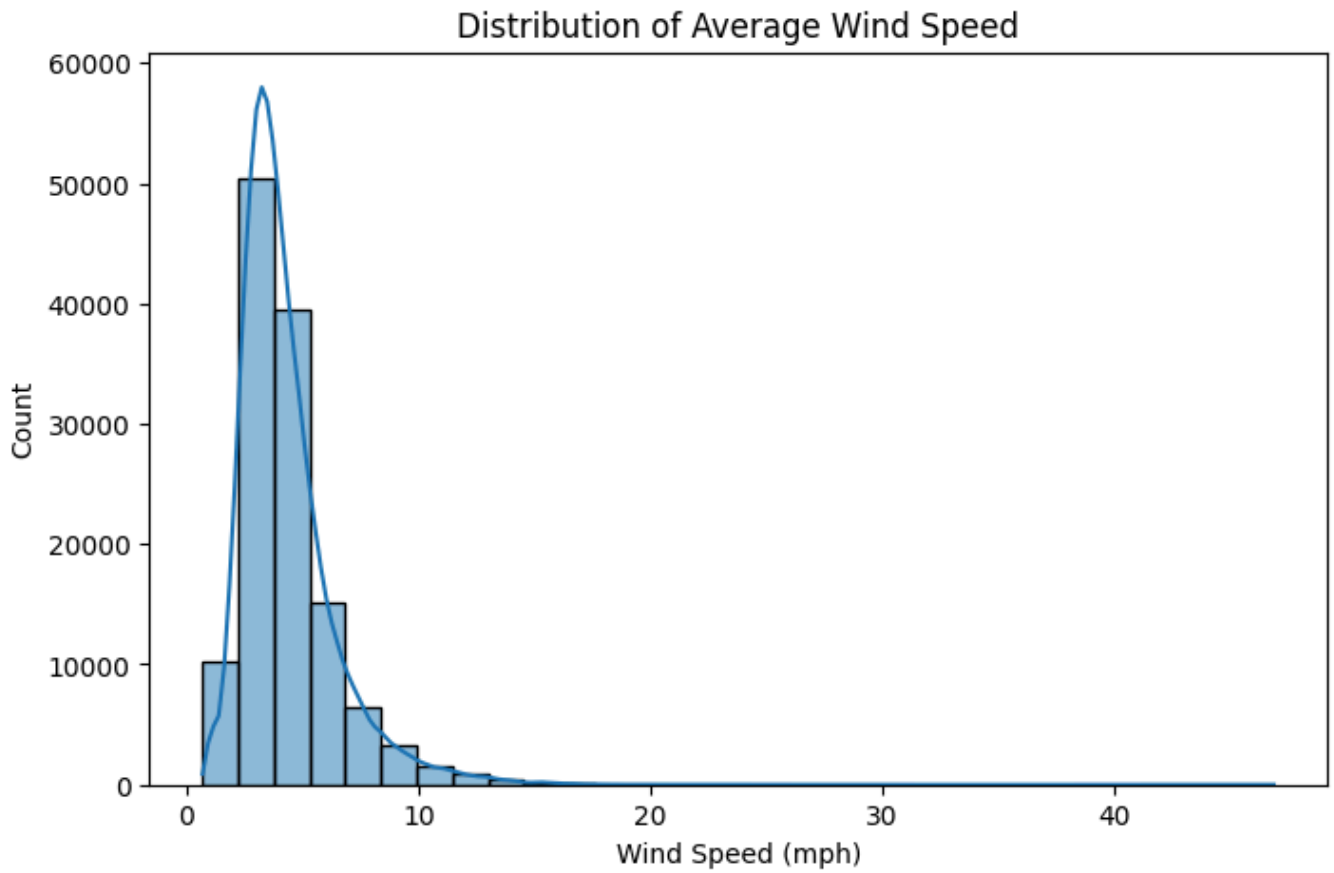
In [6]:
```python
#Handle missing values (example: fill with mean or drop)
weather_data.fillna(weather_data.select_dtypes(include=[np.number]).mean(), inplace=True)
wildfire_data.dropna(subset=['AcresBurned', 'Started'], inplace=True)  # Drop if critical colu
forest_fires.fillna(0, inplace=True)

# Explanation:
# - For weather_data, we fill missing numerical values with their mean to avoid data loss whi
# - For wildfire_data, we drop rows missing 'AcresBurned' or 'Started' as they are critical fo
# - For forest_fires, we fill any missing values with 0 since this dataset has less complexity
```

In [7]:
```python
#CONVERT DATES TO DATETIME FORMAT
#convert data columns to datetime format for proper
#alignment and merging of datasets later

#convert weather dates
#convert fire start dates
weather_data['Date'] = pd.to_datetime(weather_data['Date'], errors='coerce')  # Coerce invalid
wildfire_data['Started'] = pd.to_datetime(wildfire_data['Started'], errors='coerce')

#since dataset doesn't provide year info, we create synthetic dates based on the month
#assuming single year (create synthetic dates for the forestfires dataset)
forest_fires['synthetic_date'] = pd.to_datetime(
    forest_fires['month'] + '-01-2023', format='%b-%d-%Y'
) #create synthetic dates based on month
```

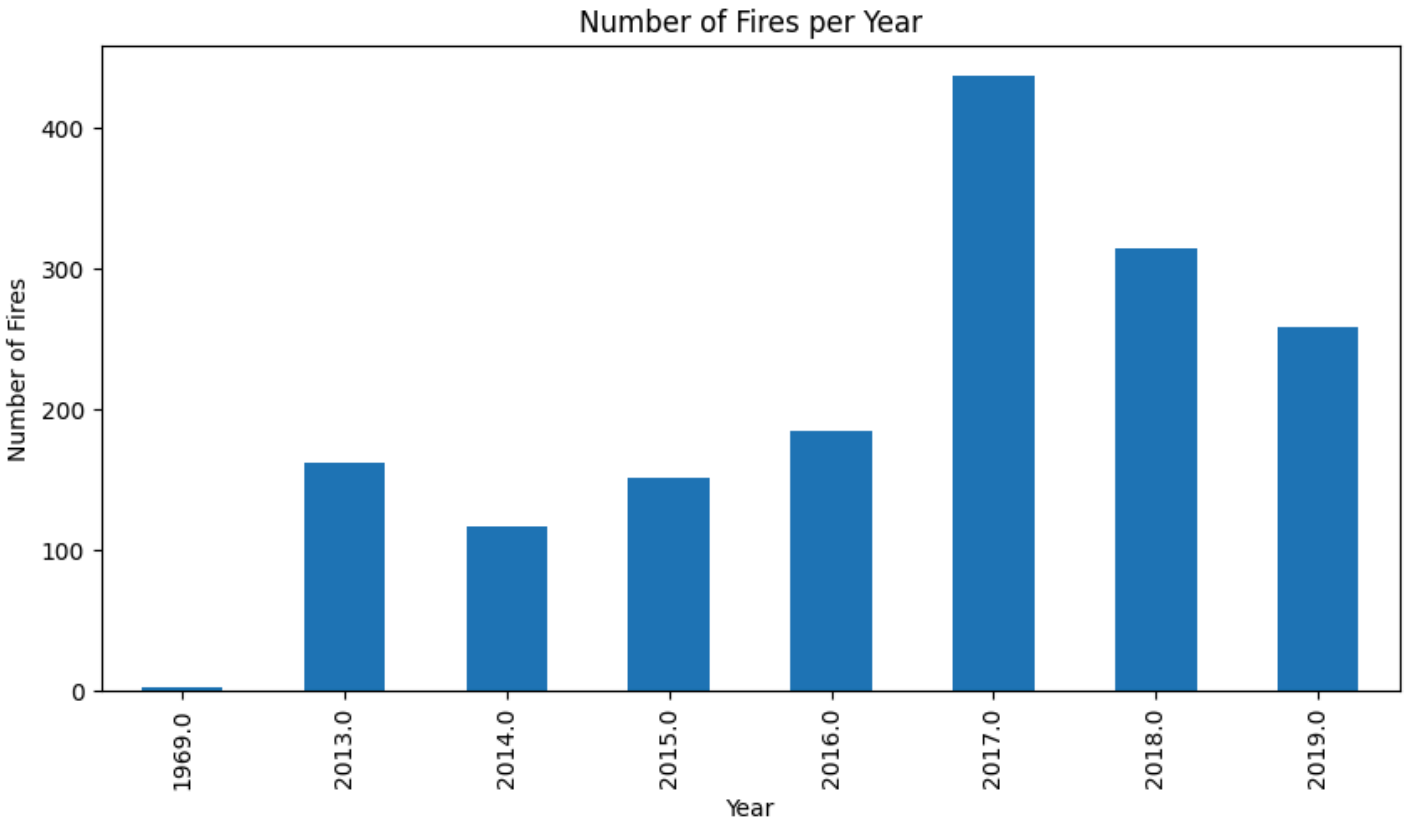
In [8]:
```python
#EXPLORATORY ANALYSIS WITH PLOTS
#visualize key features to understand the distribution
#and patterns in the data

plt.figure(figsize=(8,5)) #set figure size
#disbribution of temperature
sns.histplot(weather_data['Avg Air Temp (F)'], bins=30, kde=True)
plt.title('Distribution of Avg Air Temp')
plt.xlabel('Average Air Temp (F)')
plt.show()
```



In [9]:
```
plt.figure(figsize=(8,5)) #set figure size
#Distribution of wind speed
sns.histplot(weather_data['Avg Wind Speed (mph)'], bins=30, kde=True)
plt.title('Distribution of Average Wind Speed')
plt.xlabel('Wind Speed (mph)')
plt.show()
```



In [10]:
```
plt.figure(figsize=(10,5)) #set figure size
#fires per year
wildfire_data['Started'].groupby(wildfire_data['Started'].dt.year).count().plot(kind='bar')
plt.title('Number of Fires per Year')
plt.xlabel('Year')
plt.ylabel('Number of Fires')
plt.show()
```



In [11]:
```
#AGGREGATE WILDFIRE INCIDENTS BY DATE AND
#MERGE WITH WEATHER DATA
#aggregating the wildfire data by date so we can create features
#representing the total number of fires and acres burned per day

fire_daily = wildfire_data.groupby(wildfire_data['Started'].dt.date).agg(
    total_acres_burned=('AcresBurned', 'sum'), #total acres burned per day
    num_fires=('Started', 'count') #number of fires per day
).reset_index()

fire_daily.rename(columns={'Started':'Date'}, inplace=True) #rename for merging
fire_daily['Date'] = pd.to_datetime(fire_daily['Date']) #ensure date column is datetime
```

In [12]:
```
# Merge weather and wildfire data on 'Date'
# "Left-join" to retain all weather data and attach corresponding fire data (if available)
merged_data = pd.merge(weather_data, fire_daily, on='Date', how='left')

# Fill missing fire data with 0 (days without fires)
# Using the recommended syntax to avoid chained assignment warnings
merged_data = merged_data.fillna({
    'total_acres_burned': 0,  # No fire means 0 acres burned
    'num_fires': 0  # No fire incidents
})
```

In [13]:
```
#AGGREGATE FOREST FIRES BY MONTH AND
#MERGE WITH MAIN DATASET
#summarize the forest fire data by month to
#generate features such as avg temp, wind, and fire area burned

forest_fires_agg = forest_fires.groupby('month').agg(
    avg_temp=('temp', 'mean'), #avg temp per month
    avg_wind=('wind', 'mean'), #avg wind speed per month
    avg_humidity=('RH', 'mean'), #avg humidity per month
    avg_fire_area=('area', 'mean') #avg fire area burned per month
).reset_index()
```

In [14]:
```
#merge the aggregated forest fire features on month
#this will allow us to introduce general monthly fire-related
#variables into our main dataset

merged_data = pd.merge(
    merged_data,
    forest_fires_agg,
    left_on=merged_data['Date'].dt.month.map(
        {
            1: 'jan', 2: 'feb', 3: 'mar', 4: 'apr',
            5: 'may', 6: 'jun', 7: 'jul', 8: 'aug',
            9: 'sep', 10: 'oct', 11: 'nov', 12: 'dec'}),
    right_on='month',
    how='left'
)
merged_data.fillna(0, inplace=True) #fill missing values with 0
```

In [15]:
```
#FEATURE ENGINEERING
#create lagged features and seasonal indicators to capture temporal
#dependencies in the data

#lagged temperature and wind speed (from previous day)
merged_data['lag_1_temp']=merged_data['Avg Air Temp (F)'].shift(1)
merged_data['lag_1_wind']=merged_data['Avg Wind Speed (mph)'].shift(1)
```

```
#create a seasonal feature based on the month
#to help understand seasonal patterns (such as dry vs wet seasons)
merged_data['season']=merged_data['Date'].dt.month.map(
    lambda x:   'Winter' if x in [12, 1, 2] else
                'Spring' if x in [3, 4, 5] else
                'Summer' if x in [6, 7, 8] else
                'Fall'
)
merged_data.fillna(0, inplace=True) #fill NaNs created by lagging with 0
```

In [16]:
```
# TRAIN-TEST SPLIT FOR REGRESSION

# Drop unnecessary columns to prepare the feature set
X = merged_data.drop(columns=['Date', 'total_acres_burned', 'num_fires'])  # Features

#--------
# "One-hot" encode categorical variables
# because Random Forest Regressor only takes numerics
# Convert categorical features (like location or season) into binary columns
X = pd.get_dummies(X, drop_first=True)   # Automatically handles all object or category columr
#--------

# Target variable for regression: total acres burned
y = merged_data['total_acres_burned']

# Split the data (80% training, 20% testing)
# random_state=42 ensures reproducibility
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [17]:
```
#TRAIN A RANDOM FOREST REGRESSOR
# Train a Random Forest model, suitable for non-linear data and interpretable via feature impo

# Initialize the Random Forest Regressor
rf = RandomForestRegressor(n_estimators=100, random_state=42)  # 100 decision trees (default)

# Train the regression model using the training data
rf.fit(X_train, y_train)
```

Out[17]:
```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```

In [18]:
```
#EVALUATE THE REGRESSION MODEL

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

# Evaluate the model using regression metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Display regression evaluation metrics
print("Regression Evaluation Metrics:")
print("Mean Absolute Error (MAE):", mae)  # Average absolute error in burned area prediction
print("Mean Squared Error (MSE):", mse)  # Penalizes larger errors more heavily
print("R² Score (Explained Variance):", r2)  # How much of the variance in the data is explair
```

```
Regression Evaluation Metrics:
Mean Absolute Error (MAE): 7174.584359024389
Mean Squared Error (MSE): 4010934836.796129
R² Score (Explained Variance): -0.11373998249509709
```

In [19]:
```python
#ATTEMPT TO IMPROVE THE MODEL

# Log-transform the target variable to handle skewness and stabilize the variance
y_train_log = np.log1p(y_train)  # log(1 + burned area) to avoid log(0) issues
y_test_log = np.log1p(y_test)

# Train a Gradient Boosting Regressor, which can handle complex relationships better
improved_model = GradientBoostingRegressor(
    n_estimators=200,  # More trees for better performance
    learning_rate=0.1,  # Standard learning rate
    max_depth=5,  # Reasonable depth to avoid overfitting
    random_state=42
)

# Train the model on log-transformed target
improved_model.fit(X_train, y_train_log)
```

Out[19]:

GradientBoostingRegressor

```
GradientBoostingRegressor(max_depth=5, n_estimators=200, random_state=42)
```

In [20]:
```python
# Make predictions on the test set
y_pred_log = improved_model.predict(X_test)

# Transform predictions back to the original scale (inverse of log1p)
y_pred = np.expm1(y_pred_log)

# Evaluate the improved model using regression metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Display evaluation metrics and what they mean
print("Improved Model Regression Evaluation Metrics:")
print("Mean Absolute Error (MAE):", mae,
      "- On average, the model's predictions are off by this many acres.")
print("Mean Squared Error (MSE):", mse,
      "- The average of the squared errors, penalizing larger errors more heavily.")
print("R² Score (Explained Variance):", r2,
      "- The proportion of the variance in the actual burned area explained by the model.")
```

```
Improved Model Regression Evaluation Metrics:
Mean Absolute Error (MAE): 3627.8116012859405 - On average, the model's predictions are off by
this many acres.
Mean Squared Error (MSE): 3613915929.690722 - The average of the squared errors, penalizing la
rger errors more heavily.
R² Score (Explained Variance): -0.0034973960054591746 - The proportion of the variance in the
actual burned area explained by the model.
```

In [21]:
```python
#SAVE MODEL TO A FILE

# Save the trained model to a file
joblib.dump(improved_model, 'wildfire_burned_area_model.pkl')
```

Out[21]:
```
['wildfire_burned_area_model.pkl']
```

In [22]:
```
#SAVE PREPROCESSED DATASET FOR FUTURE USE/REVISIT

merged_data.to_csv('final_preprocessed_wildfire_data.csv', index=False)
```

In [31]:
```
#TEST CELL -- testing the model

import joblib
import pandas as pd
import numpy as np

# Save the list of features used for training
joblib.dump(X_train.columns, 'expected_features.pkl')
print("Expected features saved.")

# Load the saved expected feature list and the trained model
expected_features = joblib.load('expected_features.pkl')
loaded_model = joblib.load('wildfire_burned_area_model.pkl')

# Example user-provided input (partial)
new_data = pd.DataFrame({
    'Avg Air Temp (F)': [80],
    'Avg Wind Speed (mph)': [10],
    'Precip (in)': [0.1],
    'log_wind_speed': [np.log1p(10)],  # Log transformation
    'dry_season': [1],  # 1 for dry season, 0 for not
    'season_Spring': [0]  # Assume not Spring for this test
})

# Reindex to match the expected feature set, filling any missing features with 0
new_data = new_data.reindex(columns=expected_features, fill_value=0)

# Make predictions using the loaded model
predicted_log_burned_area = loaded_model.predict(new_data)
predicted_burned_area = np.expm1(predicted_log_burned_area)  # Convert back from log scale

# Display prediction
print("\nPredicted Burned Area of Wildfire (in acres):", predicted_burned_area[0])
```

```
Expected features saved.

Predicted Burned Area of Wildfire (in acres): 18.56230569896174
```

In [ ]: