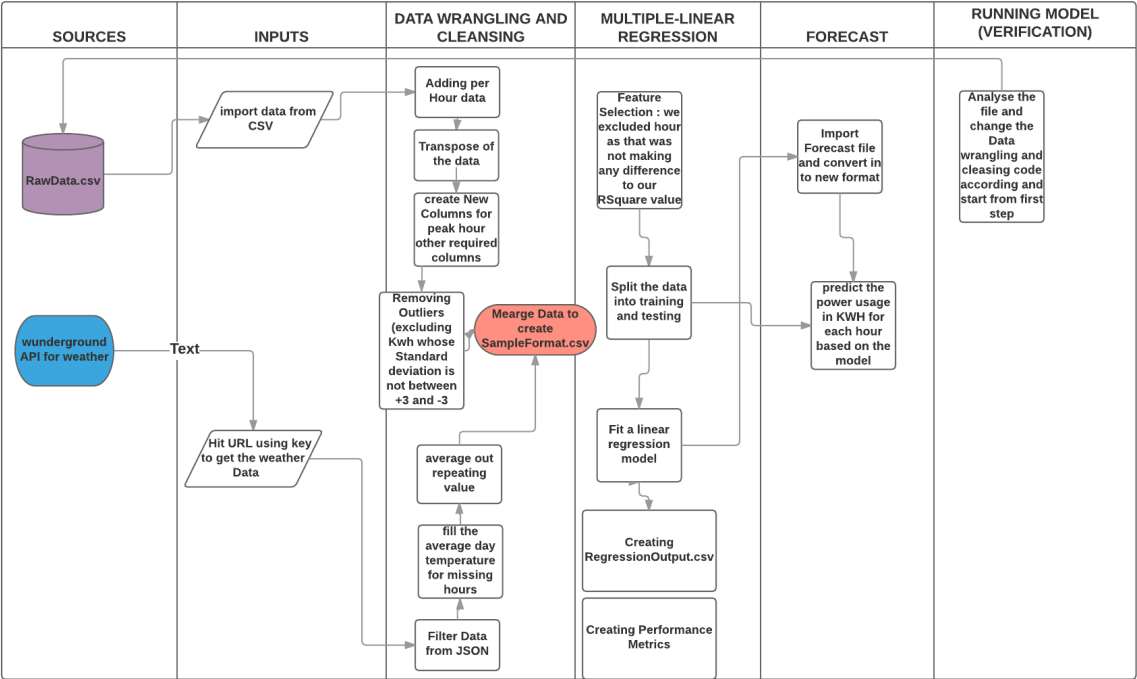


Case 1: Energy Forecasting

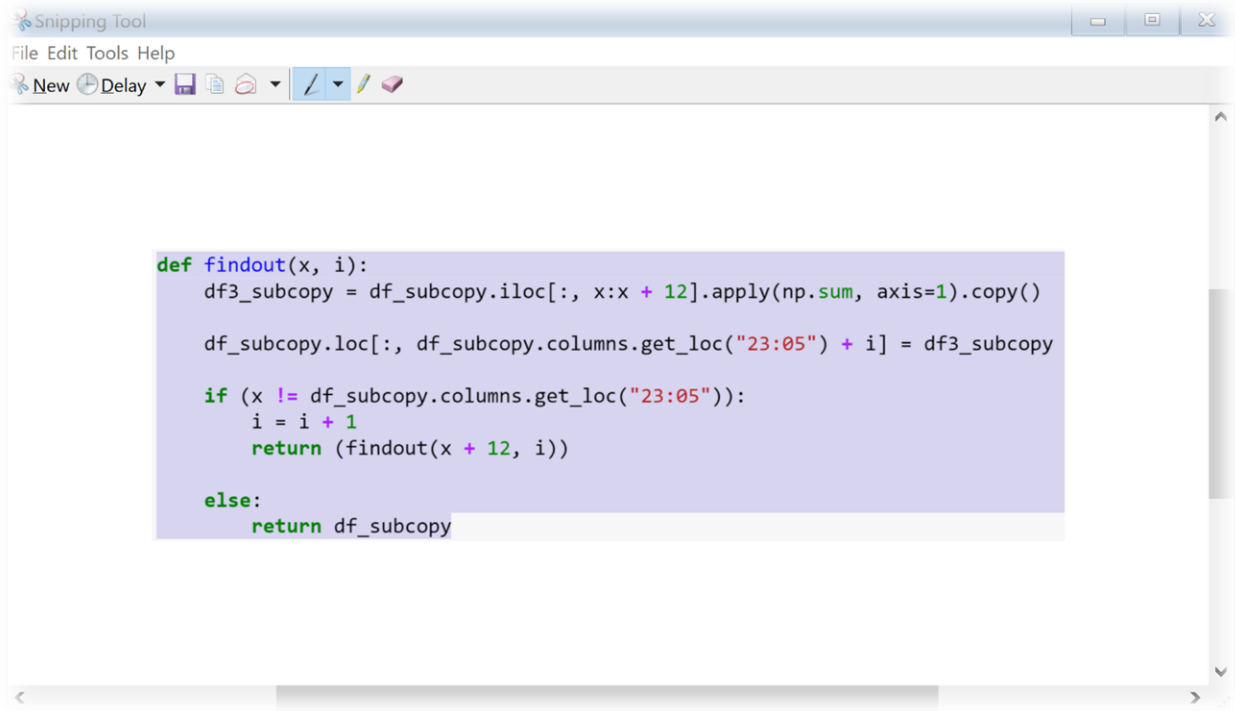
Flowchart :



Regression model design

Algorithm

- (a) To find out the sum of Energy consumed for each hour of the day



A screenshot of a Windows Snipping Tool window. The title bar reads "Snipping Tool". The menu bar includes "File", "Edit", "Tools", and "Help". The toolbar contains icons for "New", "Delay", and various snipping tools. The main area displays a Python function named `findout` with the following code:

```
def findout(x, i):  
    df3_subcopy = df_subcopy.iloc[:, x:x + 12].apply(np.sum, axis=1).copy()  
  
    df_subcopy.loc[:, df_subcopy.columns.get_loc("23:05") + i] = df3_subcopy  
  
    if (x != df_subcopy.columns.get_loc("23:05")):  
        i = i + 1  
        return (findout(x + 12, i))  
  
    else:  
        return df_subcopy
```

(b) Finding temperature for each hour of the day

```
Snipping Tool
File Edit Tools Help
New Delay

def get_temps():
    report = {}

    i = start_date
    diff = datetime.timedelta(days=1)
    while i <= end_date:
        date = i.strftime("%Y%m%d")
        url = "http://api.wunderground.com/api/b0398f0486550559/geolookup/history_" + date + "/q/MA/Boston.json"
        i += diff

        resp = requests.get(url)
        resp = json.loads(resp.text)
        times = resp["history"]["observations"]

        temps = {}
        for time in times:
            temps[int(time["date"])] = time["tempi"]

        for x in range(24):
            if x not in temps:
                temps[x] = resp["history"]["dailysummary"][0]["meantempi"]
        report[date] = temps
    return report

temps = get_temps()

def get_temp(row):
    date = row['newDate']
    date = date._date_repr.replace("-", "")

    if date in temps:
        hour = int(row["Hour"])
        if hour in temps[date]:
            return temps[date][hour]

df_subcopy['temp'] = df_subcopy.apply(get_temp, axis=1)
```

Regression Output

```
import numpy as np
import sklearn
from sklearn import datasets
import pandas as pd
energy=pd.read_csv('SampleFormat.csv')
Y= energy.ix[:, ['kWh']].copy()
X = energy.ix[:, ['Month', 'Day', 'DayOfWeek', 'Weekday', 'PeakHour', 'temp']].copy()
account = energy.ix[:, ['Account']].copy()
accountNum = account["Account"][0]
from sklearn import linear_model
lm=linear_model.LinearRegression()
lm.fit (X,Y)
dftegf = pd.DataFrame(list(zip(X.columns,lm.coef_[0])), columns = ['features','estimatedCoefficients'])
dftegy = pd.DataFrame([['AccountNumber', accountNum],['Intercept', lm.intercept_[0]]],
                      columns=['features','estimatedCoefficients'])
dfrtd = dftegf.append(dftegy)
dfrtd.to_csv('C:/Users/Abhijeet/Desktop/WT/RegressionOutputs.csv',index=False)
```

features	estimatedCoefficients
Month	1.838896111
Day	-0.299076775
DayOfWeek	-8.52654542
Weekday	32.92292457
PeakHour	37.20541443
temp	2.101704604
AccountNumber	999999999
Intercept	-66.96289963

Performance Metrics

```
import numpy as np
import sklearn
from sklearn import datasets
import pandas as pd
energy=pd.read_csv('SampleFormat.csv')
Y= energy.ix[:, ['kWh']].copy()
X = energy.ix[:, ['Month', 'Day', 'DayOfWeek', 'Weekday', 'PeakHour', 'temp']].copy()
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, random_state=0)

#Fit a Linear regression model
from sklearn import linear_model
lm=linear_model.LinearRegression()
lm.fit (X_train, y_train)
lm.coef_
train_pred = lm.predict(X_train)
test_pred = lm.predict(X_test)
from sklearn.metrics import *
mae = mean_absolute_error(y_test, test_pred)
mse = mean_squared_error(y_test, test_pred)
mase = median_absolute_error(y_test, test_pred)
account = energy.ix[:, ['Account']].copy()
accountNum = account["Account"][0]
dftegy = pd.DataFrame([[ 'AccountNumber', accountNum], [ 'MeanAE', lm.intercept_[0]],
                        [ 'MSE', mse], [ 'MedianSE', mase]],
                        columns=['features', 'estimatedCoefficients'])
dftegy.to_csv('C:/Users/Abhijeet/Desktop/WT/PerformanceMetrics.csv', index=False)
```

features	estimatedCoefficients
AccountNumber	26435791004
MeanAE	77.28279241
MSE	2624.051605
MedianSE	39.00026131

Comments/Observations on model evaluation

Upon analyzing t values and p-values and trying various permutations and combinations, we made the following observations:

(aWeekday, PeakHour and DayOfWeek contributed significantly/ had more weightage as the value of R-squared change significantly upon removing/adding these variables