Assignment 1

Description of the forecasting problem

Weather data for the city of Szeged in Hungary, from 2006 to 2016, is available. This forecasting problem deals with predicting the **Apparent Temperature** value given a value of **Humidity** on that day.

Y = Apparent Temperature
X = Humidity

Where X is the independent variable/attribute, also known as predictor variable. Based on the values of X, we will predict the values of Y which is our variable of interest.

Description of the available data

Column name	Type	Description
Formatted Date	DateTime	Date and Time of the day
Summary	String	Short Summary of the day
Precip type	String	Type of precipitation
Temperature	Numeric	Actual Temperature
Apparent Temperature	Numeric	Temperature perceived by humans
Humidity	Numeric	Value of humidity
Wind Speed	Numeric	Speed of Wind
Wind Bearing	Numeric	Direction of the Wind
Visibility	Numeric	Distance at which an object or light can be clearly discerned
Loud Cover	Numeric	Total cover
Pressure	Numeric	Value of atmospheric pressure
Daily Summary	String	Overall summary for the day

Attributes used:

Y (Variable of interest) = Apparent Temperature
X = Humidity

Link to data source

Short overview of the selected algorithms

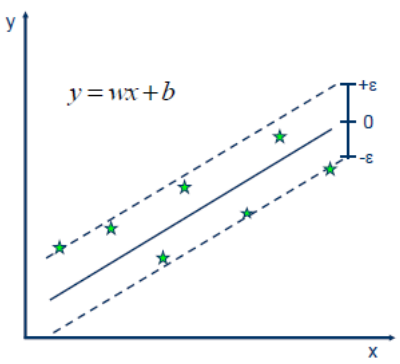
Linear Regression

Regression – an approach for modeling the relationship between a dependent variable and independent variables
Linear regression – linear relationship between a dependent variable and independent variables

In simple linear regression, we predict scores on one variable from the scores on a second variable. The variable we are predicting is called the *criterion variable* and is referred to as Y. The variable we are basing our predictions on is called the *predictor variable* and is referred to as X. When there is only one predictor variable, the prediction method is called *simple regression*. If there are more than one predictor variables, the prediction is called *multivariate linear regression.*

Support Vector Regression

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.



- Solution:

$$\min \frac{1}{2}\|w\|^2$$

- Constraints:

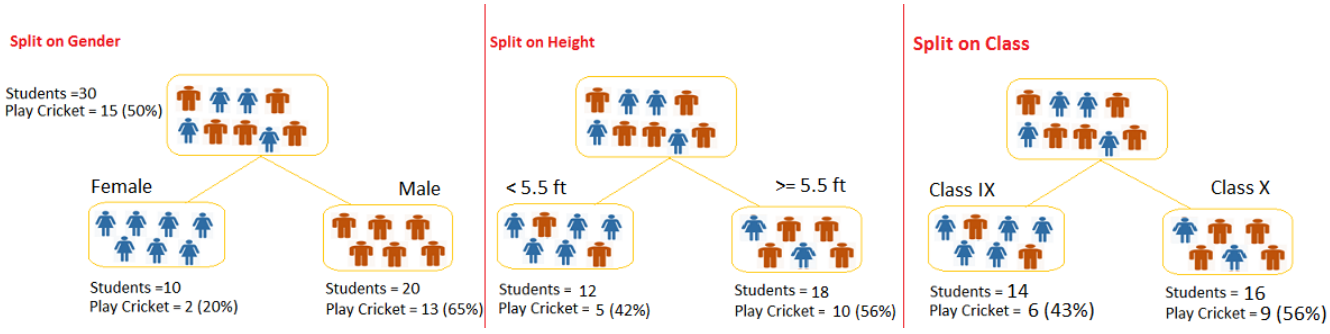
$$y_i - wx_i - b \le \varepsilon$$
$$wx_i + b - y_i \le \varepsilon$$

Decision Tree

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.

Let's say we have a sample of 30 students with three variables Gender (Boy/ Girl), Class(IX/ X) and Height (5 to 6 ft). 15 out of these 30 play cricket in leisure time. Now, I want to create a model to predict who will play cricket during leisure period? In this problem, we need to segregate students who play cricket in their leisure time based on highly significant input variable among all three.

This is where decision tree helps, it will segregate the students based on all values of three variable and identify the variable, which creates the best homogeneous sets of students (which are heterogeneous to each other). In the snapshot below, you can see that variable Gender is able to identify best homogeneous sets compared to the other two variables.



Types of Decision Tree :

Types of decision tree is based on the type of target variable we have. It can be of two types:

1. **Binary Variable Decision Tree:** Decision Tree which has binary target variable then it called as Binary Variable Decision Tree. Example:- In above scenario of student problem, where the target variable was “Student will play cricket or not” i.e. YES or NO.
2. **Continuous Variable Decision Tree:** Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

Specifics about how algorithms were applied and the evaluation procedure

I am interested in finding the relationship between **Apparent Temperature** and **Humidity.** After closely observing the data, it becomes clear that there is an inverse relationship between both. To mathematically prove it, I calculate the co-relation value in MS Excel and it comes out to be negative i.e. -0.73618961, which indicates a strong inversely proportional relationship.

CORREL (<Apparent Temp Values>, <Humidity values) = -0.73618961

Note: correlation value approaching 1 indicates strong directly proportional dependency whereas a correlation value approaching -1 indicates a strong inversely proportional dependency.

Since a casual relationship between two parameters (Apparent Temperature and Humidity) needs to be calculated, the Linear Regression Algorithm should most likely be the best fit for this kind of problem statement. However, algorithms like Support Vector Regression and Decision trees are also used.

The dataset is divided into **training** and **test** data in ratio 4:1 respectively. The training data is used to train the model and the test data is used to check the accuracy of the model.

Evaluation Procedure: Mean Squared Error

The mean squared error tells us how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It’s called the **mean** squared error as we are finding the average of a set of errors.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Where, n = number of data points

Y_i = Actual value for data point i

$Y_i^{\sim}$ =Predicted value for data point i

The smaller the mean squared error, the closer you are to finding the line of best fit. It means the algorithm which gives us the smallest value of MSE, will be our algorithm of choice.

Accuracy comparison

Algorithm	MSE
Linear Regression	10.899382981520409
Support Vector Regression	11.979722416103748
Decision Tree Analysis	12.157561528442072

Since *Mean Square Error* is least for Linear regression, it implies that Linear regression, for this particular problem statement and dataset, is a better algorithm than Support vector regression and Decision tree to model the data at hand and make more accurate predictions.

Code in Python

Linear Regression

```
1.  # Importing the libraries
2.  import numpy as np
3.  import matplotlib.pyplot as plt
4.  import pandas as pd
5.
6.  #Reading the .CSV data file
7.  dataset = pd.read_csv('//Users//muhbandtekamshuru//Downloads//SVR//HungaryWeather500.cs
    v')
8.  """dataset.head()"""
9.  X = dataset.iloc[:, 5].values
10. y = dataset.iloc[:, 4].values
11.
12. y=y.reshape(-1,1)
13. X=X.reshape(-1,1)
14.
15. # Dividing Data into Training and Test
16. from sklearn.cross_validation import train_test_split
17. X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
    = 0)
18.
19. from sklearn.linear_model import LinearRegression
20. regressor = LinearRegression()
21. regressor.fit(X_train, y_train)
22.
23. # Predicting the Test set results
```

```
y_pred = regressor.predict(X_test)

import statsmodels.formula.api as smf
a=regressor.score

dataset.columns=['Formatted Date', 'Summary', 'Precip Type', 'Temperature (C)',
       'ApparentTemperature', 'Humidity', 'Wind Speed (km/h)',
       'Wind Bearing (degrees)', 'Visibility (km)', 'Loud Cover',
       'Pressure (millibars)', 'Daily Summary']

import statsmodels.formula.api as sm

model = sm.ols(formula='ApparentTemperature ~ Humidity', data=dataset)
fitted1 = model.fit()
fitted1.summary()

y_pred=y_pred.reshape(-1,1)
a=sum(np.square(y_test-y_pred))

from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred)
```

MSE: 10.899382981520409

Support Vector Regression

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

dataset = pd.read_csv('//Users//muhbandtekamshuru//Downloads//SVR//HungaryWeather500.csv')

X = dataset.iloc[:, 5].values
y = dataset.iloc[:, 4].values

X=X.reshape(-1,1)
y=y.reshape(-1,1)

from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')

regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)
# Predicting a new result

# Visualising the SVR results
plt.scatter(X, y, color = 'red')
plt.plot(X, regressor.predict(X), color = 'blue')
plt.title('Truth or Bluff (SVR)')
```

```
plt.xlabel('Humidity')
plt.ylabel('Apparent Temperature')
plt.show()

# Visualising the SVR results (for higher resolution and smoother curve)
X_grid = np.arange(min(X), max(X), 0.01) # choice of 0.01 instead of 0.1 step because t
    he data is feature scaled
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
plt.title('Truth or Bluff (SVR)')
plt.xlabel('Humidity')
plt.ylabel('Apparent Temperature')
plt.show()

y_pred=y_pred.reshape(-1,1)

import numpy as np
a=sum(np.square(y_test-y_pred))

from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred)
```

MSE: 11.979722416103748

Decision Tree

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset = pd.read_csv('//Users//muhbandtekamshuru//Downloads//SVR//HungaryWeather500.cs
    v')
X=X.reshape(-1,1)

y=y.reshape(-1,1)

from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
    = 0)

# Fitting Decision Tree Regression to the dataset
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()

regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)
# Predicting a new result
regressor.score

y_pred=y_pred.reshape(-1,1)

a=np.square(y_pred-y_test)
```

```
28.
29. from sklearn.metrics import mean_squared_error
30. mean_squared_error(y_test, y_pred)
```

MSE: 12.157561528442072