## VALIDATION RELATIONSHIP BETWEEN THE ACCURACY AND THE PARAMETER(K VALUE) :

Importing the Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Loading the dataset

In [2]:
```
data = pd.read_csv('cleaned_iris_dataset.csv')
```

Displaying properties of dataset for further processing

In [3]:
```
data.info()
data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 93 entries, 0 to 92
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  93 non-null     float64
 1   sepal_width   93 non-null     float64
 2   petal_length  93 non-null     float64
 3   petal_width   93 non-null     float64
 4   species       93 non-null     object 
dtypes: float64(4), object(1)
memory usage: 3.8+ KB
```

Out[3]:

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| **0** | 5.045070 | 2.508203 | 3.018024 | 1.164924 | Iris-versicolor |
| **1** | 6.325517 | 2.498496 | 4.542052 | 1.413651 | Iris-versicolor |
| **2** | 5.257497 | 3.673501 | 1.470660 | 0.395348 | Iris-setosa |
| **3** | 6.675168 | 3.201700 | 5.785461 | 2.362764 | Iris-virginica |
| **4** | 5.595237 | 2.678166 | 4.077750 | 1.369266 | Iris-versicolor |

In [4]:
```
data.tail()
```

Out[4]:

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| **88** | 4.874848 | 3.217348 | 1.592887 | 0.123588 | Iris-setosa |
| **89** | 5.564197 | 2.771731 | 3.483588 | 1.074754 | Iris-versicolor |
| **90** | 5.548047 | 3.673501 | 1.453466 | 0.214527 | Iris-setosa |
| **91** | 5.510482 | 2.652867 | 4.276817 | 1.298032 | Iris-versicolor |
| **92** | 4.538713 | 3.056142 | 1.545136 | 0.241424 | Iris-setosa |

In [5]:
```
data.describe()
```

Out[5]:

| | sepal_length | sepal_width | petal_length | petal_width |
|---|---|---|---|---|
| **count** | 93.000000 | 93.000000 | 93.000000 | 93.000000 |
| **mean** | 5.867894 | 3.054063 | 3.808118 | 1.236858 |
| **std** | 0.892271 | 0.358692 | 1.811399 | 0.770872 |
| **min** | 4.344007 | 2.498496 | 1.033031 | 0.020731 |
| **25%** | 5.152435 | 2.794790 | 1.541564 | 0.343669 |
| **50%** | 5.636744 | 3.049459 | 4.192791 | 1.369266 |
| **75%** | 6.478961 | 3.239682 | 5.098860 | 1.837925 |
| **max** | 7.795561 | 3.673501 | 6.768611 | 2.603123 |

In [6]:
```
data.shape
```

Out[6]:
```
(93, 5)
```

Training the data classification model:

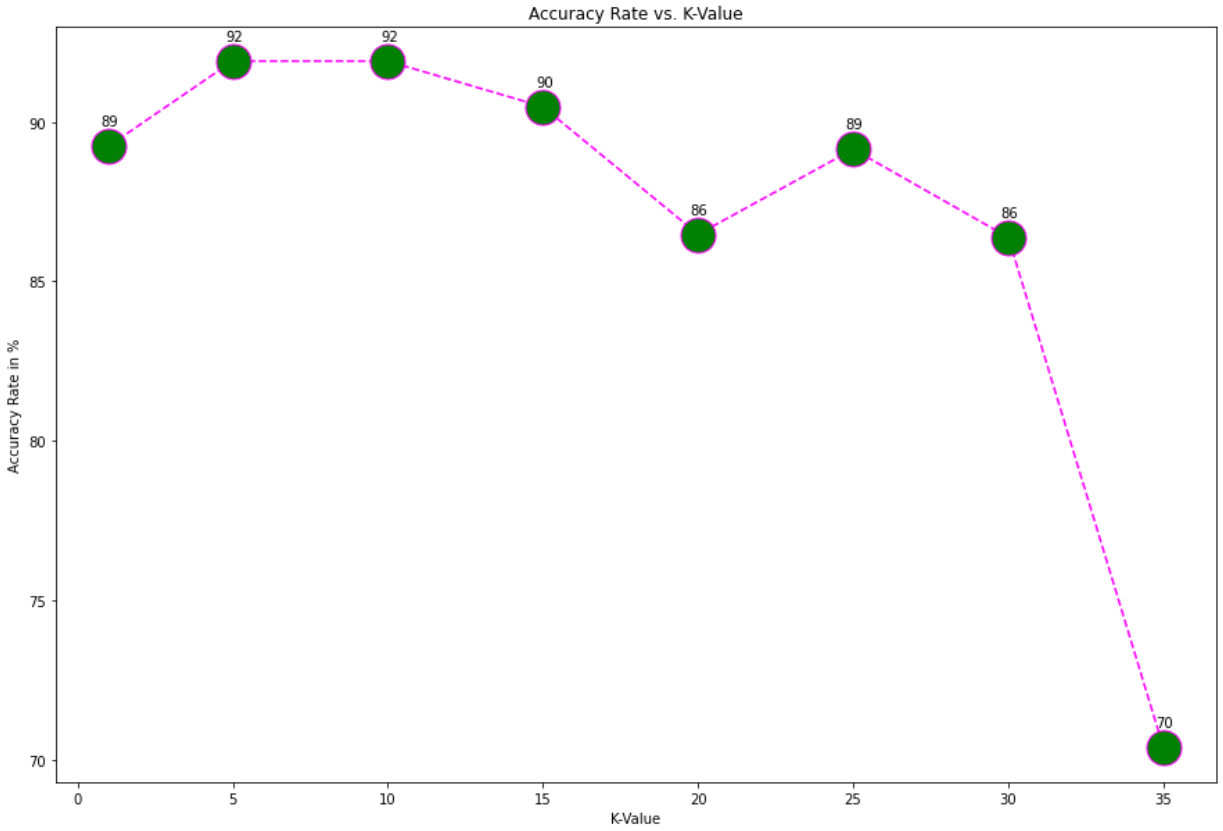
In [7]:
```
features_columns = ['sepal_length','sepal_width','petal_length','petal_width']
X = data[features_columns].values
Y = data['species'].values
```

In [8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random
```

**Calculating the best fit k value by testing the trained model against the validation set and Plotting the graph for accuracy vs K-values :**

In [9]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score,KFold

kf=KFold(n_splits=5,random_state=275,shuffle=True)

validate_score = []

listk=[1, 5, 10, 15, 20, 25, 30, 35]
for i in listk:

    knn = KNeighborsClassifier(n_neighbors=i)
    score=cross_val_score(knn, X_train, Y_train,cv=kf)
    validate_score.append(score.mean()*100)

plt.figure(figsize=(15,10))
plt.plot(listk,validate_score,color='magenta', linestyle='dashed', marker='o',
         markerfacecolor='green', markersize=25)
for x,y in zip(listk, validate_score):
    label = "{:.0f}".format(y)
    plt.annotate(label,(x,y),textcoords="offset points",xytext=(1,15),ha='center'

plt.title('Accuracy Rate vs. K-Value')
plt.xlabel('K-Value')
plt.ylabel('Accuracy Rate in %')
```

Out[9]:
```
Text(0, 0.5, 'Accuracy Rate in %')
```



From the above plot that depicts the validation relationship of accuracy rate aganist different values of K, it can be inferred that,

The best values for k can be considered are **5** and **10** as the data classification model has achieved max accuracy rate of **"92%"** when compared aganist the accuracy rates of the list of other given k values.

We are considering the value of **k=5** as the best parameter for our classifier model as this would less error rate compared to k=10 when developing the classifer model.