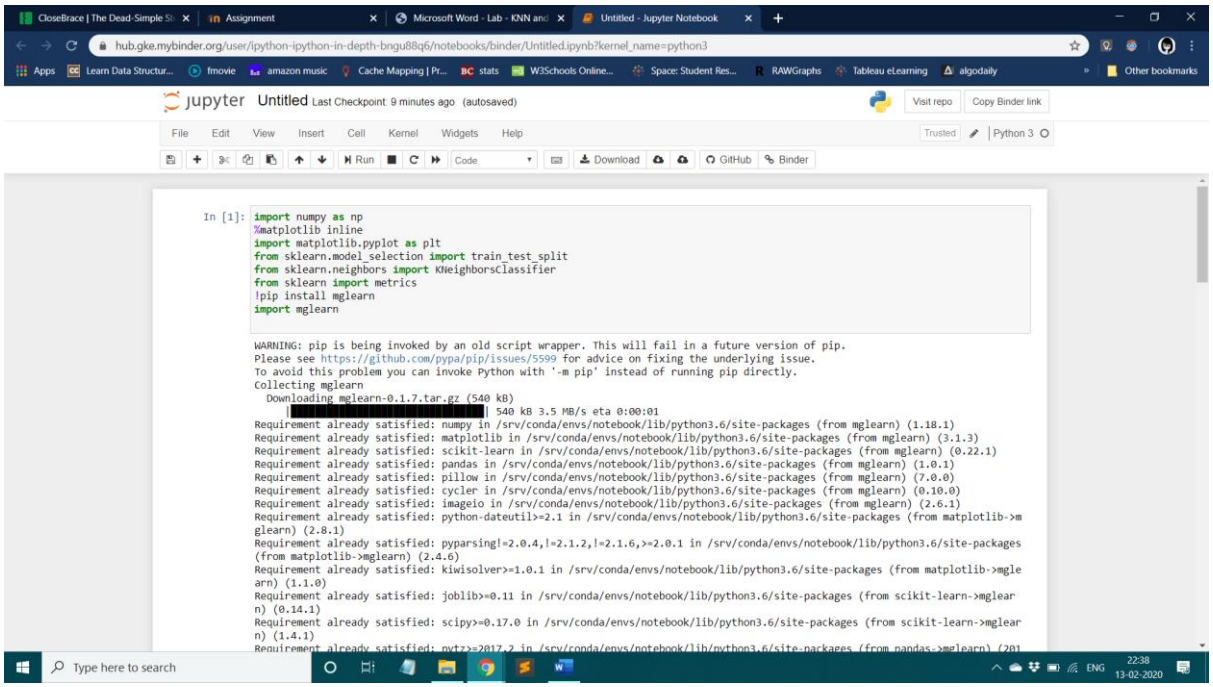
Lab 6

R Harini

18BCE1010

KNN Classifier



```
In [1]: import numpy as np
        %matplotlib inline
        import matplotlib.pyplot as plt
        from sklearn.model_selection import train_test_split
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn import metrics
        !pip install mglearn
        import mglearn
```

```
WARNING: pip is being invoked by an old script wrapper. This will fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip directly.
Collecting mglearn
  Downloading mglearn-0.1.7.tar.gz (540 kB)
     |████████████████████████████████| 540 kB 3.5 MB/s eta 0:00:01
Requirement already satisfied: numpy in /srv/conda/envs/notebook/lib/python3.6/site-packages (from mglearn) (1.18.1)
Requirement already satisfied: matplotlib in /srv/conda/envs/notebook/lib/python3.6/site-packages (from mglearn) (3.1.3)
Requirement already satisfied: scikit-learn in /srv/conda/envs/notebook/lib/python3.6/site-packages (from mglearn) (0.22.1)
Requirement already satisfied: pandas in /srv/conda/envs/notebook/lib/python3.6/site-packages (from mglearn) (1.0.1)
Requirement already satisfied: pillow in /srv/conda/envs/notebook/lib/python3.6/site-packages (from mglearn) (7.0.0)
Requirement already satisfied: cycler in /srv/conda/envs/notebook/lib/python3.6/site-packages (from mglearn) (0.10.0)
Requirement already satisfied: imageio in /srv/conda/envs/notebook/lib/python3.6/site-packages (from mglearn) (2.6.1)
Requirement already satisfied: python-dateutil>=2.1 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from matplotlib->m
glearn) (2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /srv/conda/envs/notebook/lib/python3.6/site-packages
(from matplotlib->mglearn) (2.4.6)
Requirement already satisfied: kiwisolver>=1.0.1 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from matplotlib->mgle
arn) (1.1.0)
Requirement already satisfied: joblib>=0.11 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from scikit-learn->mglear
n) (0.14.1)
Requirement already satisfied: scipy>=0.17.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from scikit-learn->mglear
n) (1.4.1)
Requirement already satisfied: pytz>=2017.2 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from pandas->mglearn) (201
```

jupyter Untitled Last Checkpoint: 9 minutes ago (autosaved)

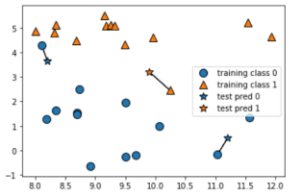
In [2]:
```
mglearn
```

Out[2]:
```
<module 'mglearn' from '/srv/conda/envs/notebook/lib/python3.6/site-packages/mglearn/__init__.py'>
```

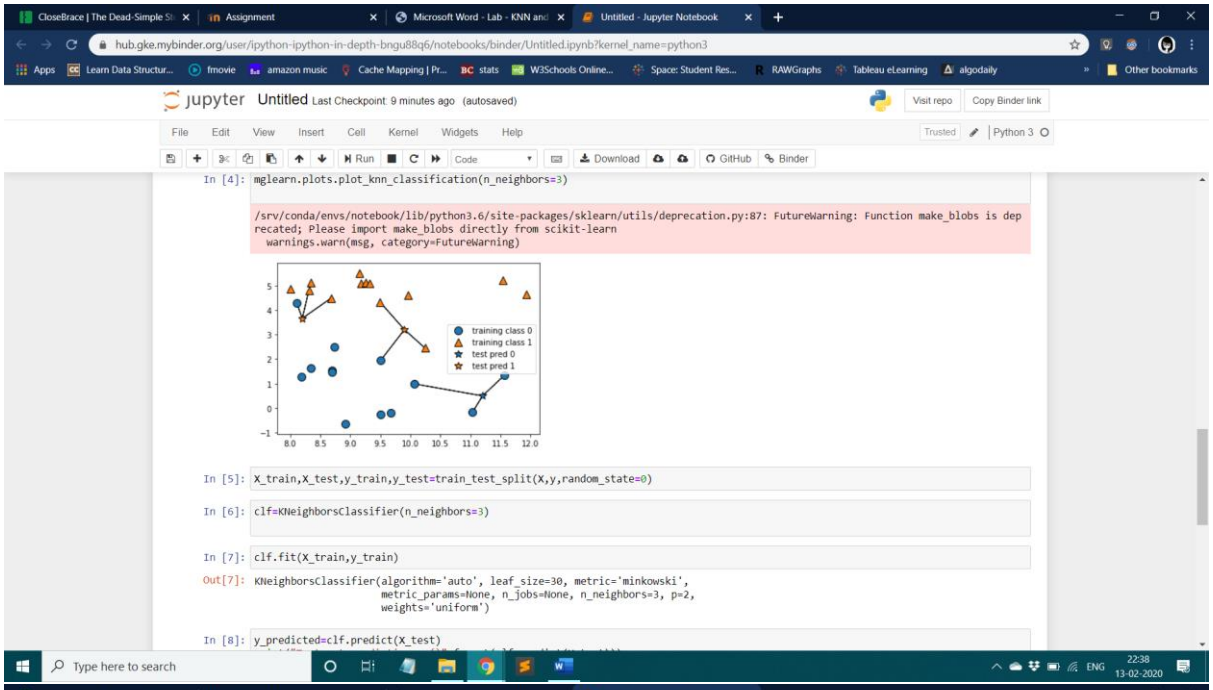
In [3]:
```
X,y = mglearn.datasets.make_forge()
print(X.shape)
print(y.shape)
mglearn.plots.plot_knn_classification(n_neighbors=1)
```

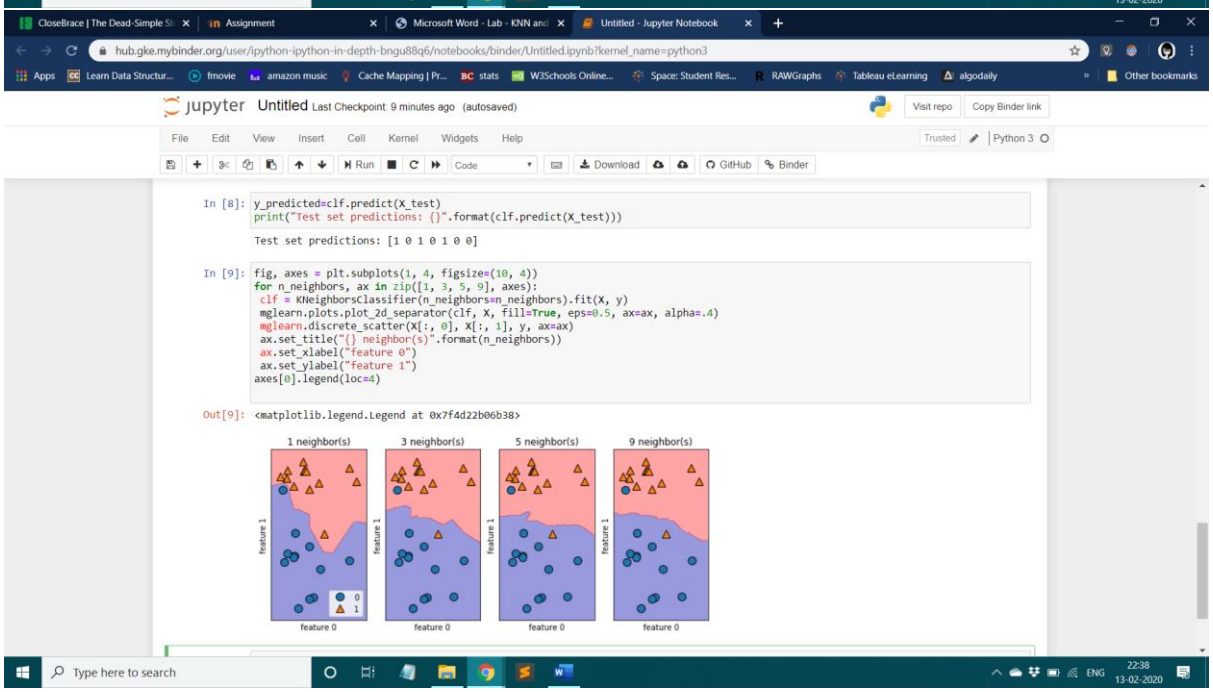
```
/srv/conda/envs/notebook/lib/python3.6/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function make_blobs is dep
recated; Please import make_blobs directly from scikit-learn
  warnings.warn(msg, category=FutureWarning)

(26, 2)
(26,)
```



In [4]:
```
mglearn.plots.plot_knn_classification(n_neighbors=3)
```





df=pd.read_csv('cancer.csv')

df

df= df.drop(columns=['id','Unnamed: 32'])

df.head()

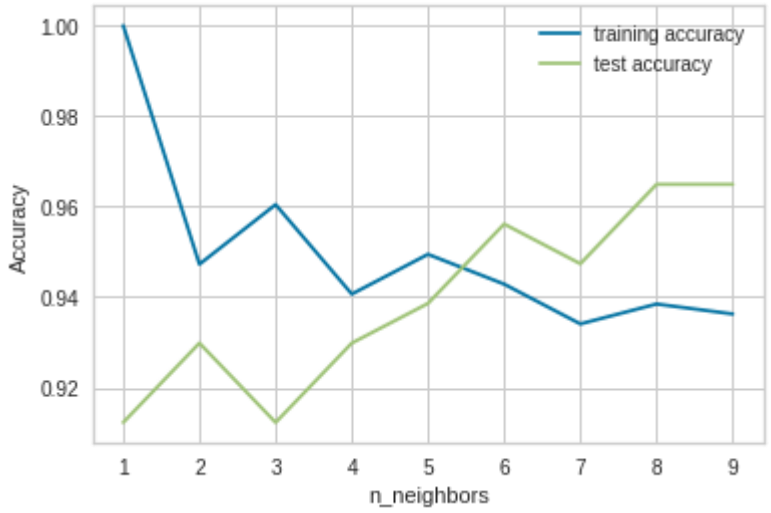
from sklearn.model_selection import train_test_split

array=df.values

df.head()

X=array[:,1:33]

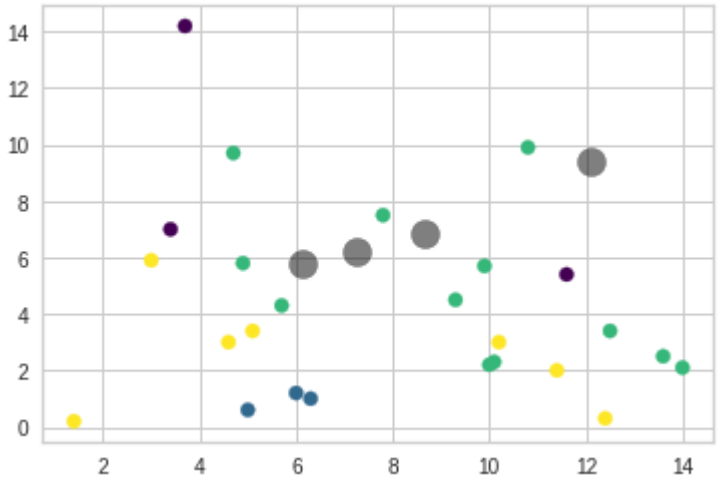
```
y=array[:,[0]]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
X_train.shape,X_test.shape
clf=KNeighborsClassifier(n_neighbors=3)
clf.fit(X_train,y_train)
acc_train = clf.score(X_train, y_train)
print('Train set accuracy: ', acc_train)
acc_train = clf.score(X_test, y_test)
print('Train set accuracy: ', acc_train)
training_accuracy =[]
testing_accuracy = []
neighbors_settings = range(1, 10)
for n_neighbors in neighbors_settings:
    clf = KNeighborsClassifier(n_neighbors=n_neighbors)
    clf.fit(X_train, y_train)
    training_accuracy.append(clf.score(X_train, y_train))
    testing_accuracy.append(clf.score(X_test, y_test))
plt.plot(neighbors_settings, training_accuracy, label="training accuracy")
plt.plot(neighbors_settings, testing_accuracy, label="test accuracy")
plt.ylabel("Accuracy")
plt.xlabel("n_neighbors")
plt.legend()
plt.show()
```



KMeans Clustering

```
from sklearn.cluster import KMeans
import numpy as np
df=pd.read_csv('protein.csv')
df
X = df.iloc[:,1:].values
kmeans = KMeans(n_clusters=4)
y_kmeans = kmeans.fit_predict(X)
print(y_kmeans5)
kmeans.cluster_centers_
plt.scatter(X[:, 1], X[:,4], c=y_kmeans, s=50, cmap='viridis')


centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
```




```
from sklearn.metrics import pairwise_distances_argmin


def find_clusters(X, n_clusters, rseed=2):
    # 1. Randomly choose clusters
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]

    while True:
        # 2a. Assign labels based on closest center
        labels = pairwise_distances_argmin(X, centers)

        # 2b. Find new centers from means of points
        new_centers = np.array([X[labels == i].mean(0)
                                for i in range(n_clusters)])

        # 2c. Check for convergence
        if np.all(centers == new_centers):
            break
        centers = new_centers
```

```
    return centers, labels


Error =[]
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i).fit(X)
    kmeans.fit(X)
    Error.append(kmeans.inertia_)
import matplotlib.pyplot as plt
plt.plot(range(1, 11), Error)
plt.title('Elbow method')
plt.xlabel('No of clusters')
plt.ylabel('Error')
plt.show()
```