Classification with Nearest Neighbors.

In this question, you will use the scikit-learn's KNN classifer to classify real vs. fake news headlines. The aim of this question is for you to read the scikit-learn API and get comfortable with training/validation splits. We will use a dataset of 1298 "fake news" headlines (which mostly include headlines of articles classified as biased, etc.) and 1968 "real" news headlines, where the "fake news" headlines are from https://www.kaggle.com/mrisdal/fake-news/data and "real news" headlines are from https://www.kaggle.com/therohk/million-headlines. The data were cleaned by removing words from titles not part of the headlines, removing special characters and restricting real news headlines after October 2016 using the word "trump". The cleaned data are available as clean_real.txt and clean_fake.txt in hw1_data.zip on the course webpage. It is expected that you use these cleaned data sources for this assignment. You will build a KNN classifier to classify real vs. fake news headlines. Instead of coding the KNN yourself, you will do what we normally do in practice — use an existing implementation. You should use the KNeighborsClassifier included in sklearn. Note that figuring out how to use this implementation, its corresponding attributes and methods is a part of the assignment.

- **(a)** Write a function load_data which loads the data, preprocesses it using a **CountVectorizer** (http://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text), and splits the entire dataset randomly into 70% training, 15% validation, and 15% test examples.
- **(b)** Write a function select_knn_model that uses a KNN classifer to classify between real vs. fake news. Use a range of k values between 1 to 20 and compute both training and validation errors, leaving other arguments to KNeighborsClassifier at their default values. Generate a plot showing the training and validation accuracy for each k. Report the generated plot in your write-up. Choose the model with the best validation accuracy and report its accuracy on the test data.
- **(c)** Repeat part (b), passing argument metric='cosine' to the KNeighborsClassifier. You should observe an improvement in accuracy. How does metric='cosine' compute the distance between data points, and why might this perform better than the Euclidean metric (default) here? Hint: consider the dataset ['cat', 'bulldozer', 'cat cat cat'].