CSC111 Project 2 Proposal: JetSetGo

Xin Lei Lin, Evan Su, Rick Chen, Yi He (Robert) Li

March 5, 2024

Problem Description and Research Question

Nowadays there are millions of people travelling by plane everyday. By planning our flight reasonably, we can save tons of time and money, which not only make our lives more efficient but also improve our life quality. Thus, a good flight recommending system is essential for most of the people. Therefore, our group project is an efficient flight recommending system. Our project aims to provide the best flight route for every user from a student to a billionaire. We plan to use several different algorithms to enhance the flight search experience.

The final goal of our project is taking the user and its location as input, recommend similar or different places to go and based on user's current location, and its characteristics, all while recommend the shortest, the cheapest and the most-valued flight to the user, ensuring a wonderful flight experience.

Computational Plan

Graphs and trees represent the central way that transportation lines and airports will be represents. Each vertex will represent airports around the world, and the edges in the graph will represent the existence of a flight between two airports. This data will come from the pyflightdata API. [1]

We will also create a class object for each destination containing information about the climate and costs of living at each location (retrieved from "Global Cost of Living" [2] and "Average Temperature of Cities" [3]). Our recommend system will utilize clustering algorithms to recommend similar locations to the user.

The K-nearest neighbour (KNN) algorithm is a widely used machine learning algorithm[4]. It belongs to the class of instance-based or lazy learning algorithms (it does not learn an explicit model from the training data, but stores the entire training dataset for prediction).The basic idea is to classify new data points based on the class labels of their closest k neighbours in the training data. KNN involves two hyperparameters, K value and distance function.
K value : how many neighbours participate in the KNN algorithm. k should be tuned based on the validation error.
Distance function : Euclidean distance is the most used similarity function.

Support Vector Machine or SVM is a popular Supervised Learning algorithms used for Classification and Regression[5]. SVM aims to create a decision boundary for an n-dimensional space into classes for future data points. SVM chooses the extreme points/vectors that help in creating the best decision boundary. These extreme cases are called support vectors.

We will be employing libraries like Tensorflow and/or Scikit-learn, in addition to data manipulation libraries like numpy and pandas.
KNNs can be implemented with Scikit-Learn, a machine learning library with supervised models and data manipulation tools, such as train_test_split() for our training and validation data, while the KNN module renders an easy-to-use KNN model [6].
Furthermore, we will use libraries like Jupyter, to run Jupyter notebooks, a library which enables us to run certain parts of a same file disjointly, instead of having to running our import statements every time. [7]

To find "optimal" flights, we will use Dijkstra's shortest path algorithm [8] on a weighted, directed graph to determine the best path to match a set of criteria, (cheapest route or fastest route).Dijkstra's is useful for directed graphs [9], since many flights are one-way and not bidirectional. We will also filter real-time flight data so that each flight/edge in the graph is at a time that is before any of its subsequently connected flights.

This assumes that there exists a flight path between two airports. To ensure this, we can use graph traversal techniques like BFS and DFS [10]. We could use DFS to determine whether there exists a series of flights connecting two airports.

Kosajaru's algorithm will be useful, as a secondary graph structure, containing various information for strongly connected components. This will ensure faster computational search of the large database, in order to determine the existence of a flight between two cities, by commuting through hubs of central airports. [11]

To display our final results to the user, we will use a graphical plotting library, notably `plotly` [12], to display an interactive world map and superimpose our suggested flight paths and points of interest onto it. This allows the user to visualize potential flight paths and hover over them for more information. In terms of gathering the user's criteria for optimal flights, we will using a simple command-line system. We might also implement a simple user system where each user is uniquely identified by their name and their flight preferences are saved into a JSON file.

References

[1] 'pyflightdata API documentation'. pyflightdata, 2018. https://pyflightdata.readthedocs.io/en/latest/pyflightdata.html.

[2] 'Global Cost of Living'. kaggle, 2022. https://www.kaggle.com/datasets/mvieira101/global-cost-of-living

[3] 'Average Temperature of Cities' kaggle, 2022. https://www.kaggle.com/datasets/swapnilbhange/average-temperature-of-cities

[4] "What is the k-nearest neighbors (KNN) algorithm?". IBM, 2024.

https://www.ibm.com/topics/knn: :text=The%20k%2Dnearest
%20neighbors%20(KNN)%20algorithm

[5] Finley, Thomas, and Thorsten Joachims. "Supervised clustering with support vector machines." Proceedings of the 22nd international conference on Machine learning. 2005.
https://dl.acm.org/doi/abs/10.1145/1102351.1102379
casatoken=dQhUVlYlS3YAAAAA:2HiXOF90ou5rq89-
SN_ujcsGk9PQKsiIm8hzGw_6EUSAynAGLjOukpo6H6iW6Hwy7FnDUz3xkJ4z

[6] "Learn." Scikit, scikit-learn.org/stable/. Accessed 2 Mar. 2024.

[7] "Tensorflow." TensorFlow, www.tensorflow.org/. Accessed 2 Mar. 2024.

[8] 'Dijkstra's Algorithm'. Wikipedia, 23 Feb. 2024. Wikipedia, https://en.wikipedia.org/w/index.php?
title=Dijkstra%27s_algorithm&oldid=1209859384.

[9] 'Directed Graph'. Wikipedia, 26 Nov. 2023. Wikipedia, https://en.wikipedia.org/w/index.php?
title=Directed_graph&oldid=1186925478.

[10] 'Graph Traversal'. Wikipedia, 2 Feb. 2024. Wikipedia, https://en.wikipedia.org/w/index.php?
title=Graph_traversaloldid=1202449254.

[11] GfG. "Strongly Connected Components."
GeeksforGeeks, GeeksforGeeks, 17 Jan. 2024,
www.geeksforgeeks.org/strongly-connected-components/.

[12] Lines. https://plotly.com/python/lines-on-mapbox/. Accessed 2 Mar. 2024.