Module 12 Project — XN Project: Presentation Slide Deck

Shang-Shiun (Brian) Tsai

Northeastern University, Seattle

ALY 6080: Integrated Experiential Learning

Jayash Koshal

Executive Summary

- Project overview
- EDA of two dataset
- Basic research on model building
- Building models with current dataset
- Website design
- Paper research
- Search for more data to optimize the model

Business Problem

- What types of products or services do your sponsor's corporations sell?
- Who are their competitors?
- Problems that the company is concerned about?

Clear Concise Flow

Phase 1
Project overview

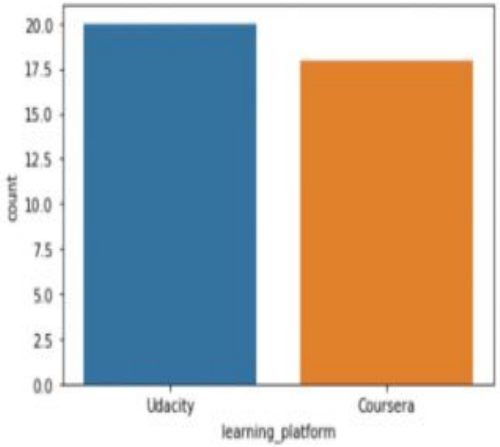
Phase 2
EDA

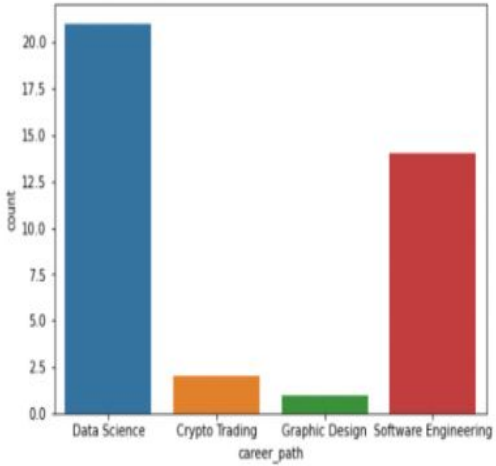
Phase 3
Model building

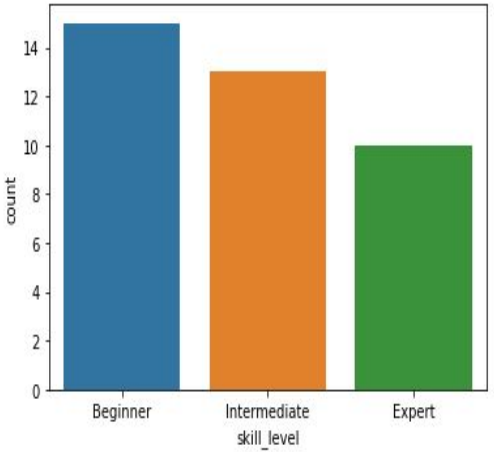
Phase 4
Model optimization & visualization

Phase 5
Future research

Analytics



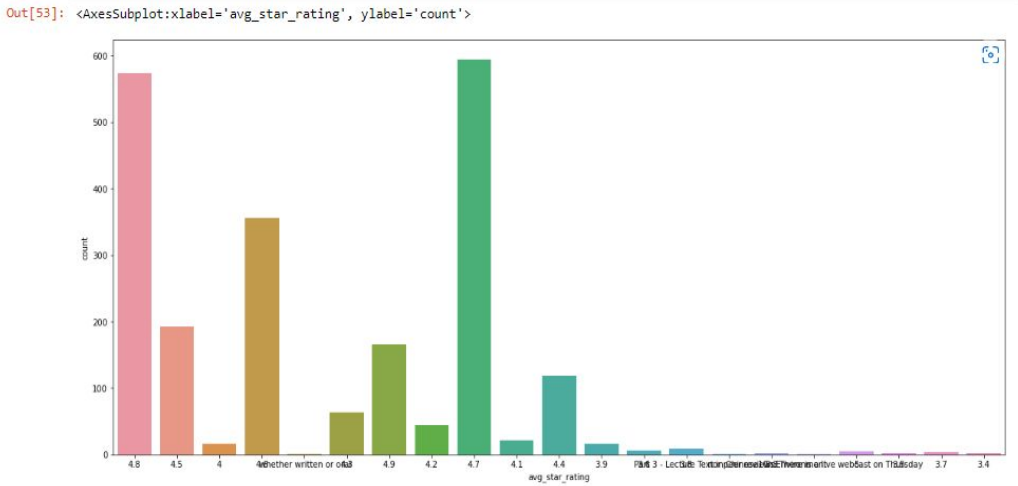


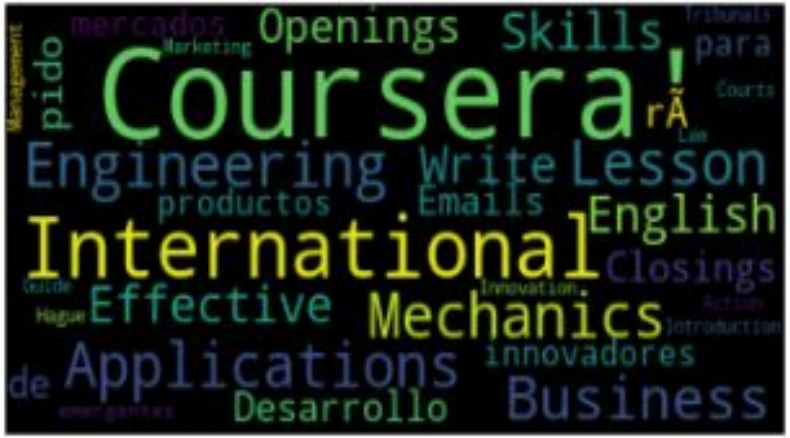


- Users prefer to use Udacity more often than Coursera
- Data science and software engineering are the most popular two topics
- The majority of the learners are beginners

EDA for Coursera Dataset

- Contains 2603 obs., 11 variables
- There are 486 courses have the same instructors
- Most of the course have the ratings higher than 4.5
- Most of the courses that covers name with Engineering, Write, Mechanics, Business





Analysis and Synthesis of the Data

Use Tfidfvectorizer to extract the words, then calculate cosine_similariry matrix, and recommend the most similar results to user, based on user's input key words

```
#Import TfIdfVectorizer from scikit-learn
from sklearn.feature_extraction.text import TfidfVectorizer

#Define a TF-IDF Vectorizer Object. Remove all english stop words such as 'the', 'a'
tfidf = TfidfVectorizer(stop_words='english')

#Replace NaN with an empty string
df['course_name'] = df['course_name'].fillna('')

def remove_pipeline(text):
    return re.sub("\\|", '', text)
# preg_match("/\|/", "This is not a pipe.

df['course_name']=df['course_name'].apply(remove_pipeline)
df['course_name']

#Construct the required TF-IDF matrix by fitting and transforming the data
tfidf_matrix = tfidf.fit_transform(df['course_name'])

# #Output the shape of tfidf_matrix
tfidf_matrix.shape
```

```
(38, 69)
```

```
# Import linear_kernel
from sklearn.metrics.pairwise import linear_kernel

# Compute the cosine similarity matrix
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
```

Recommendations

- Optimize models:
  - Currently built a QA model that recommends a broader answers to users without concise recommendations. We will integrate other models and re-modify in the coming weeks.
- Find more model resources:
  - Searching what and how to improve our models in the coming of weeks with the resources that are available to us.
- Apply competitors dataset to build a more accurate model:
  - Kaggle and other platforms are a good source to find datasets from indirect competitors such as Coursera and Udemy. They will be used to take references.

```
get_recommendations('Predictive Analytics for Business')
```

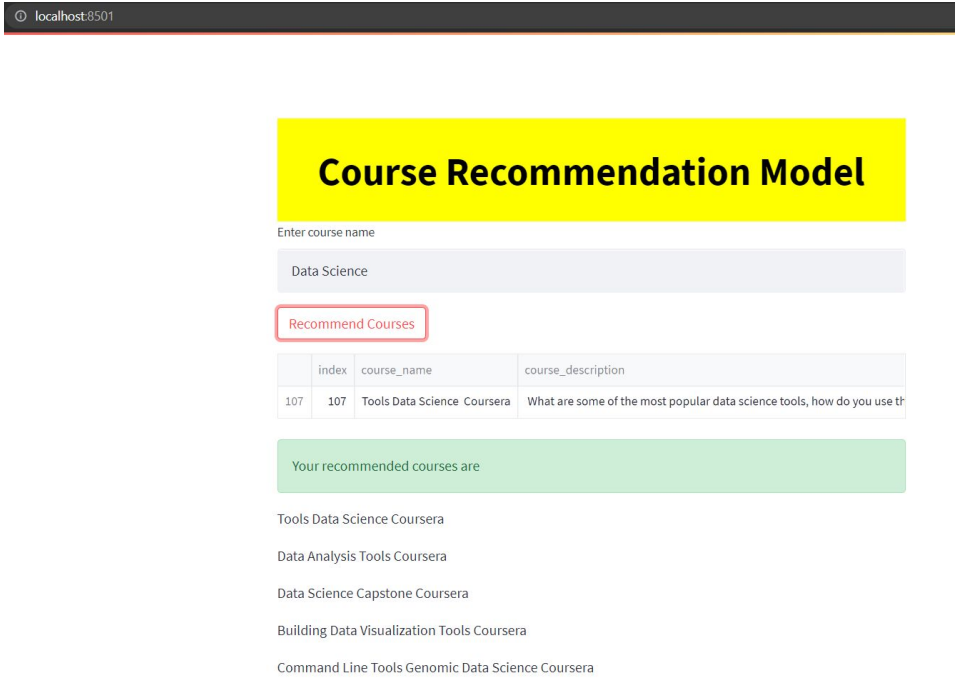
```
2     Business Analytics Online Course  Learn Busine...
18         Data Science for Business & Business Leaders
15              Power BI Data Analytics  Date Analytics
22                 Blockchain Business Models  Coursera
```

Recommendations web

https://chaitraanaik-recommendersystem-temp-ojm2vt.streamlit.app/?fbclid=IwAR3V0wvlU__7Ie2SRTvZv0XFuSUQ2sYJCYZCMVFbGLnlBxfOTLe6kHheVp0

- Our Recommendation model is built using TF-IDF and cosine similarity to find the closest matches to the user inputs.The final model is integrated with a front end GUI spp to provide a highly interactive user experience.

- **Tools used** :Steps include data processing the user input to remove special characters and blank spaces.
    - Also, the non-English courses are filtered from the dataset keeping only the relevant English courses.
    - Using the TF-IDF matrix and cosine similarity, we find the courses that have a high similarity score to the user input.
    - Finally, this is deployed using a front end web app using streamlit library and the matching records are returned according to the user input.
    - The web app runs on the local host in the web browser, thus providing a rich UI



Future Research

- How to make a QA model that recommends and generates more concise answers?
- To add or augmente data to increase the size of dataset?
- Take reference on direct and indirect competitors' QA model & recommendation systems (Course Recommendation System).

Future Research Paper Results

- How to make a QA model that recommends and generates more concise answers?
  - Euclidean distance: Detect the minimum distance of questions from the answers.
  - If improvement is needed, use cosine similarity to improve accuracy score.
- To add or augmente data to increase the size of dataset
  - To augment data to increase dataset is proved to be useful when dataset is of smaller size.
- Take reference on direct and indirect competitors' QA model & recommendation systems (Course Recommendation System).
  - Example datasets include ones from Coursera and Udemy, both provided by stakeholders.

Reference

- *https://storage.googleapis.com/pub-tools-public-publication-data/pdf ...* (n.d.). Retrieved June 14, 2022, from https://b.hatena.ne.jp/entry/s/storage.googleapis.com/pub-tools-public-publication-data/pdf/43438.pdf
- Design, D. (2022). Data-Driven Design: An Integral Part of UX Design :: UXmatters. Retrieved 18 June 2022, from https://www.uxmatters.com/mt/archives/2020/10/data-driven-design-an-integral-part-of-ux-design.php#:~:text=UX%20design%20uses%20research%20data,%2C%20churn%2C%20and%20acquisition%20analytics.
- How to Organize Data Labeling for Machine Learning: Approaches and Tools. (2022). Retrieved 18 June 2022, from https://www.altexsoft.com/blog/datascience/how-to-organize-data-labeling-for-machine-learning-approaches-and-tools/
- Swalin, A. (2018, June 1). *Building a question-answering system from scratch- part 1*. Medium. Retrieved June 18, 2022, from https://towardsdatascience.com/building-a-question-answering-system-part-1-9388aadff507

Thank you