PROJECT PROPOSAL
Predicting Binary Outcomes
GitHub Repo: https://github.com/Kebudi/Data1030-Final-Project.git

The Problem and the Data

In 2017, Professor Drazen Prelec of MIT Psychology department published a study on predicting outcome of binary questions. In his paper he suggested that if we have data on: the question, participant's answer, participant's guess on how many people will give the correct answer, and participant's confidence in his own answers, we would be able to build a bayesian model that predicts the answer of the binary question accurately. While he reached an accuracy of 75% with his bayesian model, Professor Joachim Krueger of Brown Psychology department argued that there is an easier and more successful way of achieving a higher accuracy than 75%, if another model is used. As he started working on this model, he also reached out to me, suggesting that I should work on a machine learning model, that might deliver even better results than his new model.

The main goal of the project is to be able to come up with a model that, given we have the participant's answer, his confidence in his answer and confidence in the sample group, we would be able to guess whether the true answer to the question is True or False. **The target variable for the project is the true answer.** Given that the target variable is not a continuous variable and that the project is supervised, we can confidently assume that the project is a **classification problem**. We want to classify the question to be True or False, given the sample populations response to the three feature.

The total number of participants across the 3 datasets is 97, with 7 features for each one of them in every dataset = [‘binary_question, 'topic', 'subject', 'own', 'actual', 'meta', 'confidence’]. The data sets come in 3 pieces, with each one on a different topic: (1) Binary questions on the capitals of states in the USA surveyed at MIT [*example question: is the capital of Pennsylvania Philadelphia?*], (2) Binary trivia questions on many different subject surveyed at Princeton University, [*example question: is George Bush Jr. the youngest president in the US history?]* and (3) Binary question on malicious moles on the body asked to dermatologist [*example question: is this mole malicious?].* the variables in each set is defined as:

- **Binary Question:** the question asked to the participants. This is a unique question that is only asked within the dataset it is found in. Every participant in that dataset has answered it.
- **Topic:** the topic of the question *(example: states, dermatology, space, physics…).* This is a categorical variable, with a single value in dataset_1 (‘state’), single value in dataset_3 (‘dermatology), and 80 different values in dataset_2, depending on which category the trivia question belongs to.
- **Subject:** the unique ID for the participant
- **Own:** the response of the participant to the binary question
- **Actual**: the correct response to the binary question

- **Meta:** the continuous variable between 0-100, the percentage of the study group the participant thinks will get this question right
- **Confidence:** the continuous variable between 50-100, the percentage of confidence (probability of being right) the participant has for the response he has given. This variable has a minimum value of 50% because in a binary question your success possibility cannot go below 0.5.

Preparing the Data for Processing

Given that our target variable is the actual values of the questions, we need to restructure the datasets to be a single dataset with the unique binary questions operating as the index. Furthermore we need to have the meta and confidence continuous variables to be bucketed into categories. This way we would be able to normalize the different datasets into a comparable form by dividing the number of people who fall into the bucket by the number of unique people in the dataset. Furthermore we normalize each question's TRUE answer by counting the number of times the question was answered as TRUE and dividing it by the number of people in the dataset. This way we can train the model to understand the questions hardness level for the group.

The final dataset looks like this once prepared for processing:

unique_question	own	meta*[0, 0.05]	meta*[0.05, 0.1]…(18 more)	confidence*[0.5, 0.55]	confidence*[0.55, 0.6]…(8 more)	number_of_people	topic	actual score
unique question IDs	number of people who said TRUE to the question/ total group size	number of people in this group / total group size	number of people in this group / total group size	number of people in this group / total group size	number of people in this group / total group size	number of people in the dataset	string of topic	1/0

Processing the Data

When we pre-process the data, we will use the following encoders on the following categories:

- **Binary Question:** no encoder will be used on it as it this variable will serve as the index of the dataset
- **Topic:** OneHotEncoder, this will be used to understand if topic of a certain type have more impact in the model or not. Given that topics are given as strings and cannot be ordered, OneHotEncoder is the type we will use to encode it.
- **Own:** no pre-processing is necessary, the variable is continuous between [0,1]
- **Actual**: TARGET VARIABLE, no pre-processing is necessary, the variable is binary
- **Meta:** no pre-processing is necessary, the data is continuous between [0-1] for all 20 categories (buckets).
- **Confidence:** no pre-processing is necessary, the data is continuous between [0-1] for all 10 categories (buckets).

- **Meta:** the continuous variable between 0-100, the percentage of the study group the participant thinks will get this question right
- **Confidence:** the continuous variable between 50-100, the percentage of confidence (probability of being right) the participant has for the response he has given. This variable has a minimum value of 50% because in a binary question your success possibility cannot go below 0.5.

Preparing the Data for Processing

Given that our target variable is the actual values of the questions, we need to restructure the datasets to be a single dataset with the unique binary questions operating as the index. Furthermore we need to have the meta and confidence continuous variables to be bucketed into categories. This way we would be able to normalize the different datasets into a comparable form by dividing the number of people who fall into the bucket by the number of unique people in the dataset. Furthermore we normalize each question's TRUE answer by counting the number of times the question was answered as TRUE and dividing it by the number of people in the dataset. This way we can train the model to understand the questions hardness level for the group.

The final dataset looks like this once prepared for processing:

unique_question	own	meta*[0, 0.05]	meta*[0.05, 0.1]…(18 more)	confidence*[0.5, 0.55]	confidence*[0.55, 0.6]…(8 more)	number_of_people	topic	actual score
unique question IDs	number of people who said TRUE to the question/ total group size	number of people in this group / total group size	number of people in this group / total group size	number of people in this group / total group size	number of people in this group / total group size	number of people in the dataset	string of topic	1/0

Processing the Data

When we pre-process the data, we will use the following encoders on the following categories:

- **Binary Question:** no encoder will be used on it as it this variable will serve as the index of the dataset
- **Topic:** OneHotEncoder, this will be used to understand if topic of a certain type have more impact in the model or not. Given that topics are given as strings and cannot be ordered, OneHotEncoder is the type we will use to encode it.
- **Own:** no pre-processing is necessary, the variable is continuous between [0,1]
- **Actual**: TARGET VARIABLE, no pre-processing is necessary, the variable is binary
- **Meta:** no pre-processing is necessary, the data is continuous between [0-1] for all 20 categories (buckets).
- **Confidence:** no pre-processing is necessary, the data is continuous between [0-1] for all 10 categories (buckets).

- **Number of people:** StandardScaler, this is a numerical value for each data set that does not have a numerical maximum limit. Thus, I will use StandardScaler to scale it.

You can find the processed dataset that is ready to train the machine on the GitHub repository. Size of the dataset is 210 rows x 116 columns.

Accuracy Target and Balance

With 210 total unique questions, 113 of them have 0 as their actual response and 97 have 1 as their actual response
df_final.groupby('actual').size() #0: 113, 1: 97
This makes the balance (looking at 0s); 113/210 = 0.5380952380952381 = 53.8%
This means that the model should beat at least 53.8% accuracy.

Issues to Tackle

The main goal of the project is to be able to create a model that once deployed on a question that has not been seen before, return the predicted answer with a high accuracy. We believe that this can be achieved, if the model works, by just asking the participants to write their response to the binary question (1/0), their meta prediction (between 0-1) and their confidence in their answer (between 0.5-1). Yet, one of the issues that needs to be tackled is the distribution of knowledge among the participants. All 3 datasets that will be used in this model are from either studies in prestigious universities (Princeton or MIT), or they are questions asked to the experts in their fields. (dataset 3, pictures of malignant moles asked to dermatologists). This introduces a non-normal distribution of knowledge. Therefore, we need to incorporate a value in the final dataset that quantifies the number of people in the dataset which could be considered to be experts in the given topic.

One suggestion by Professor Joachim Krueger is to look at the individual subject int the raw datasets and compare the correctness of their answer, with their meta and confidence. If the subject is correct, and has a very high confidence while arguing that a very low percentage of the population will get the question right (meta score), it is likely that the subject is an expert in the matter. However this new variable needs to be studied before being added to the final training model.