Project Everest Report

Author: Jakub Bartnik

GitHub URL

https://github.com/jakubbartnik/UCDPA_jakubbartnik

Abstract

This review contains the summary about impact of introducing new safety gear for climbing versus expedition survival that climbed in area of Hymalaya Nepal within last 100 years. Implememntation process is splited between four parts: Setting goeals, data preparation, data exploration, analisis. The review will explain how all that was done by myself and what conclusion I made from results. Report content can be found in python notebook including the code so it is easier to follow steps.

There is another python notebook in repository demonstrating simple webscrapping. File name is Web_scrapping_quotes.ipynb.

Introduction

I decided upon this topic as climbing is, quite simply, my passion. As this is my first Data Science Project I felt that I should choose a topic that I am familiar with. The aim of this research is to ascertain if, after the introduction of safety equipment such as steel carabiners (1910) or dynamic rope and harness (1964), there has been any improvement in Nepal Himalaya area climber's survival rates. My research is based on a Mount Everest dataset that tracks all expeditions that climbed in area of Nepal Himalaya from 1905 to 2019.

The last point is how many helpers should expedition hire in order to be successfull. This will be predicted based on only sucessfull expeditions considering number of members and hired staff data using ML.

Dataset

The Himalayan Database is a compilation of records for all expeditions that have climbed in the Nepal Himalaya. The database is based on the expedition archives of Elizabeth Hawley, a longtime journalist based in Kathmandu, and it is supplemented by information gathered from books, alpine journals and correspondence with Himalayan climbers.

The data cover all expeditions from 1905 through Spring 2019 to more than 465 significant peaks in Nepal. Also included are expeditions to both sides of border peaks such as Everest, Cho Oyu, Makalu and Kangchenjunga as well as to some smaller border peaks. Data on expeditions to trekking peaks are included for early attempts, first ascents and major accidents.

Implementation Process

Setting goals

I like to split my main goal from introduction into smaller chunks that will be easier to archive and at the same time makes this review more structured and hopefully readable.

Goal 1 is to check impact of first climbing carabineer on expedition survival rate. Carabiner was developed in 1910. Therefore I am going to split all dataframe into two sets and compare the falls.

Goal 2 is about introduction of dynamic rope and harness in 1960's. I am going to split all data and compare periods with and without it.

Goal 3 is to figure out how many helpers(hired_staff) should expedition hire in order to be successfull. Result should be done in precentage.

Data preparation

I started to look at the data manually prior writing this review. Those data are splited between three csv files and I belive for my pourpose I want to merge them all in one file. I going to import all libraries in first code box and keep updating it based on functions I want to use later in the code. This hopefully makes a sense later where I dont have to look over all code boxes in python notebook and run each one of them in case of missing one library.

Reading data from all three csv (expeditions.csv, peaks.csv, members.csv). I am using commands like .shape and .info. Those commands showing me column names and few records for examination but at the same time it gives me number of records. Thanks to that I know that expeditions dataframe is 10364 rows and 16 columns, peak.csv is 467 rows and 8 columns and members.csv is 76519 rows and 21 columns. I am

exploring all csv one by one using .head() which shows me five first rows to identify best column I can merge with another dataframe. This is how I identified column ‘peak_id’ to merge expeditions.csv with peak.csv and assigned it to ‘df1’ as temporary dataframe. Next I done the same with my temporary dataframe and members.csv based on ‘expedition_id’ and assigned it to dataframe -’df’. I did used .merge command to combine all three csv’s.

At this point I started looking for empty values in my current dataframe – ‘df’. I am using .isna() function to show all ‘nan’ values. Because I find the column that will make significant meaning on my results I want to take closer look at column ‘death_cause’. That is why I am using df[‘death_cause’].unique() function. Thanks to that I see that value ‘Other’ is not really make more sense then ‘nan’ or ‘unknown’ but it generate more noise so I am replacing all ‘Other’ with ‘nan’.

To limit variables in current dataset that are mainly repeated like year_x and year_y I will drop some of the columns that came after merging. Important columns from expedition.csv is ‘termination_reason’ and ‘expedition_id’, ‘year’. Important columns from peaks.csv is ‘peak_name’ and ‘climbing_status’. Important columns from members.csv is ‘death_cause’.Those I belive are important columns I want to keep. I will drop the following non important columns: trekking_agency, peak_alternative_name, sex, expedition_role, injury_type, injury_height_metres, oxygen_used_x, age, first_ascent_country, hired, success, member_id, highpoint_metres_y, year_y, season_y, peak_name_y, peak_name_x, peak_id_y. Result of dataframe without dropped columns was saved to temporary dataframe ‘cleaning’

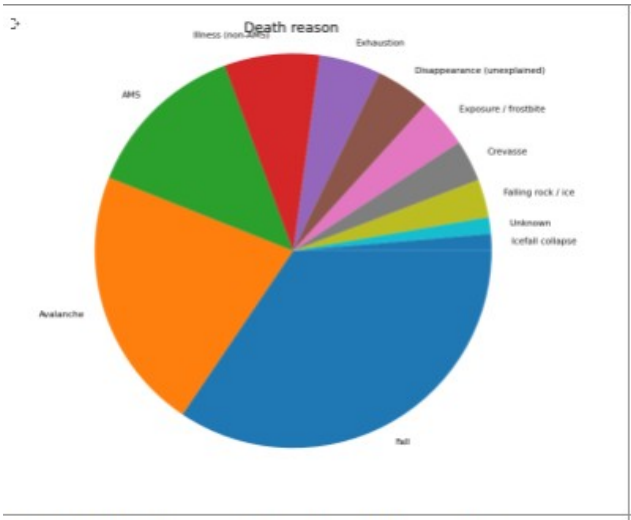
Dropped columns did not stopped me to find duplicates in this dataframe but as this review is about expeditions there was no need to have all members listedwhere most of the values were repeated per expedition. Right now i will be able to find all duplicates that can be dropped and makes more sense for this review. I done that based on field ‘expedition_id’. I sorted values and finally dropped it and result saved to ‘exp_dup’.

Stripping down string from expedition_id using simple regex to gain additional 10 points.

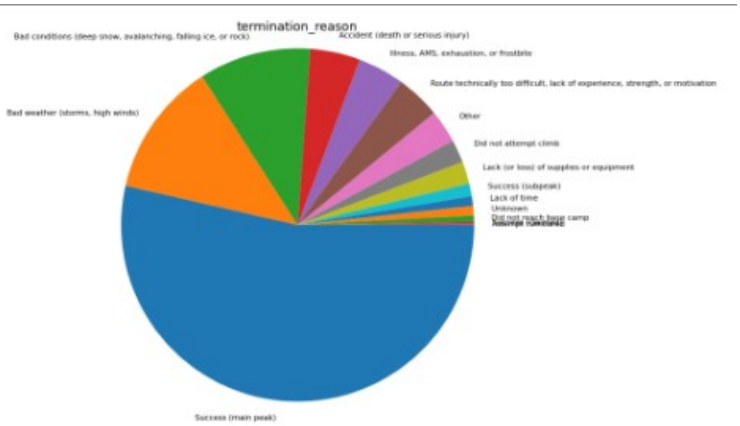
Cleaned data I saved to new file clean_everest.csv to avoid running all code boxes in python notebook.

Data exploration

Cleaned data saved as result of cleaning are loaded as ‘dfc’. Features termination_reson and death_cause are nominal i need to convert it to dummy variables. I am doing it using dataframe option ‘.get_dummies’.

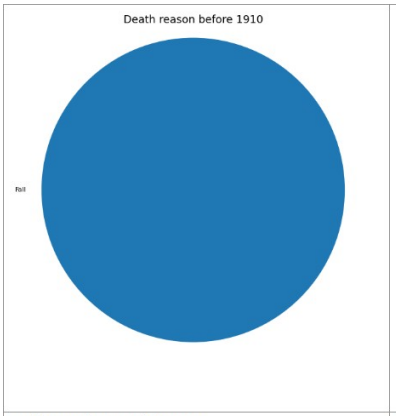


Graph 1. Death reason based on 114 years data

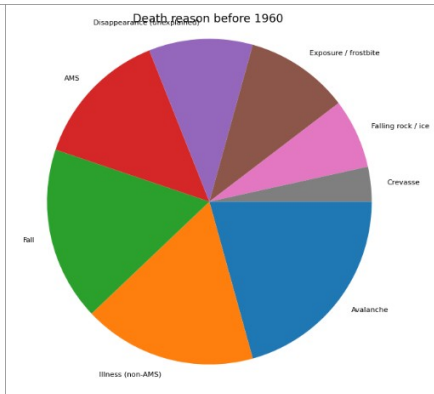


Graph 2. Termination reason based on 114 years data

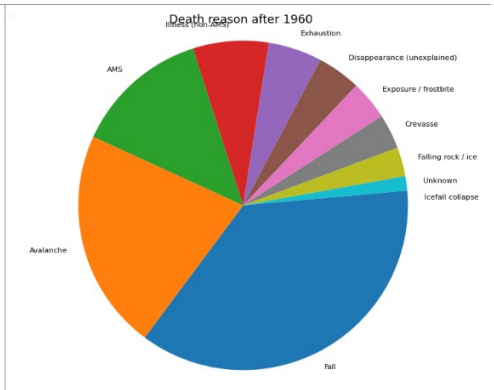
To get feeling an overview current dataset I want to show overall procentage of reasons to die in Himalaya based on all 114 years of experience. I am not really curius about exact procentage value but more generic representaion. I am going to represent this using pie chart. Avalanches, AMS(Alternated Mind status - caused by trauma or intoxication) and Illnes are the main reasons for death after falling. The same chart and same period of time just on termination reason for expeditions. This column will get useful further in this paper when I will try to narrow down ‘goal 3’. Getting back to our goals and split data for goal 1 which is before 1910 assign it to dataframe ‘before1910’ and after 1911 assigned to ‘after1910’ . For goal 2 I will split from 1905 to 1960 and assigned to ‘no_harness_rope’ and from 1961 to 2019 assigned to ‘yes_harnes_rope’. Before in the cleaning part there were columnes dropped as non-important for this review. At the moment as I am thinking to build predictive model I want to drop another column called 'peak_name' as its not numerical. I am doing it per dataset. I want to keep exploring to find at least "apparent effect" that reasons are changing within the years.



Graph 3: Death reason before 1910



Graph 4: Death reason before 1960



Graph 5: Death reason after 1960

The result is actually schocking it appear that falling was the only reason to die in high montains before 1910. Other reasons like frostbite, ilness or avalanche was not the case in this years. However while looking at amount of records used for this graph its obvious that there was not enough observations.

Pie chart for data before 1960 indicate that avalanche and ilness was the biggest reasons to die and falling going into third place.

Graphs 5 and 1 are very similar but graphs 3 and 4 indicating that resons were changing along the years .

Analisys

We have to move to goal 2 as data are more sufficient compares to goal 1. The only way it can be analyzed is to use grouping method. Groping it into categories it should be possible to measure percentage or frequency. Will try to apply hypothesis testing which is method of statistical inference. I am going to carry this analysis using Chi -squared. Referring to goal 2, zero hypothesis is that climbing has became much saver after development of dynamic rope and harness which translates to probability of falling and die is less then 50%. This hypothesis will be marked as H0 and tried to be proved as correct.

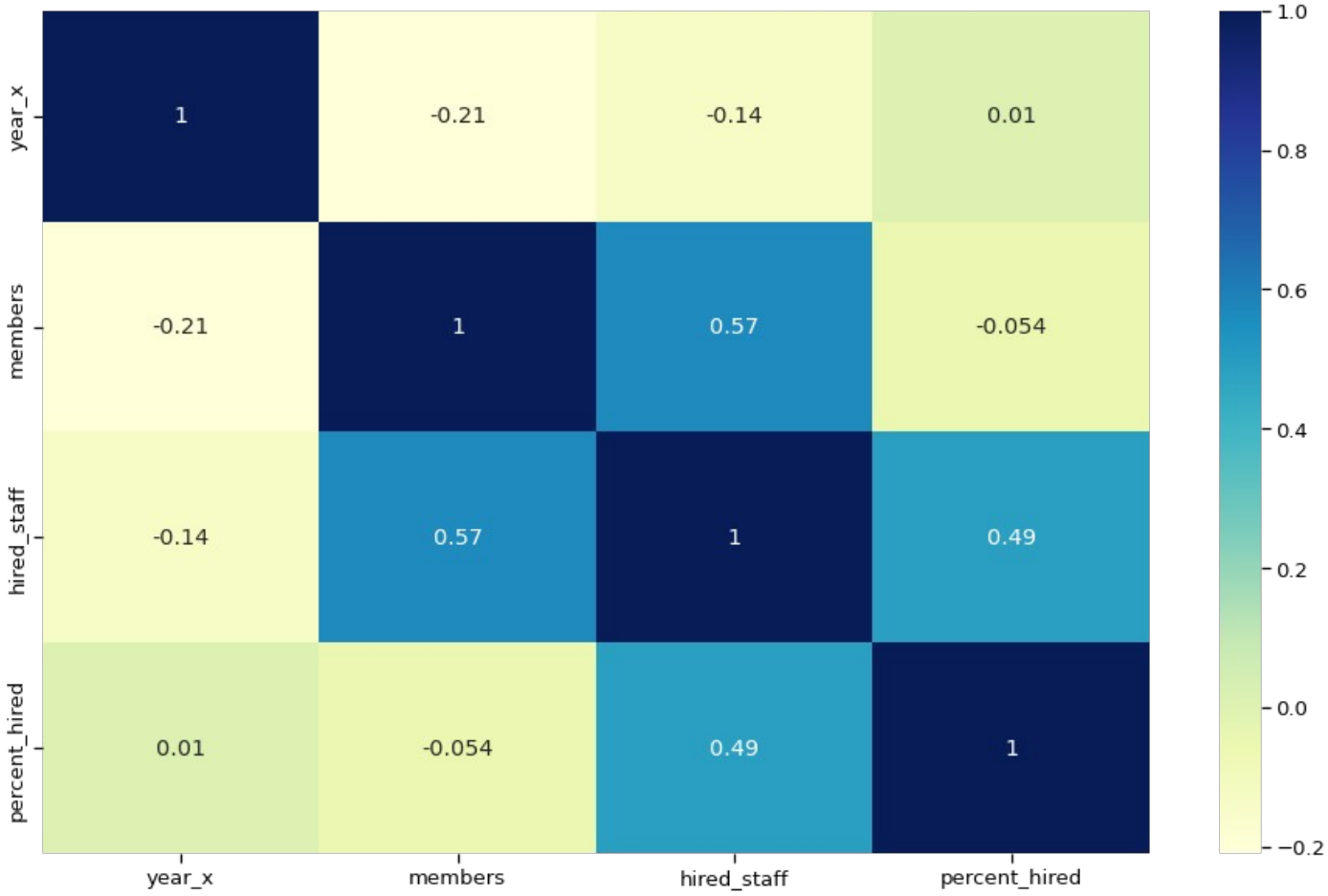
H0 = (.50, .50)

Creating function that will save yes and no for falls in separate feature. Then compared percentage "Falls before 1960 %- 0.3703703703703704 Falls after 1960 %- 0.3672971686890317"

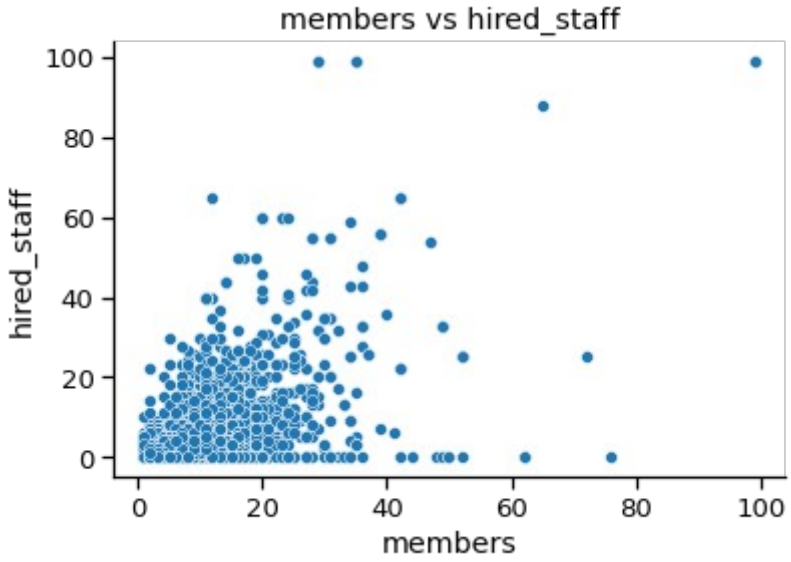
Please refer to results for more findings.

Goal 3. Whats the percentage of hired_staff that need to be hired for expedition to be successful.

Created new dataframe limiting features to year, members, hired_staff, termination reason. Then created another df where only subset for success as a termination reason will be considered and dropped this feature as no longer needed. Using DataFrame.transform() method created new column percent_hired which is calculation for percent of hired staff to members. Describing data there are multiple infinity and NaN values. Replaced infinite values data with NaN and then dropped NaN. Ploted histogram for correlation analysis using scatter matrix function. This actually created more confusion so checked correlation based on heatmap.



At this point decided to take closed look at scatterplot members vs hired_staff:



There is not obvious line that could be drawn. Based on this chart. My predictor is members my outcome should be percent_hired.

Provided variables for training set and splitting X and y into training and testing set 80:20 ratio.

Evaluation of test via comparison. My model explain 42%. Fitting model to the data provided me with answer:

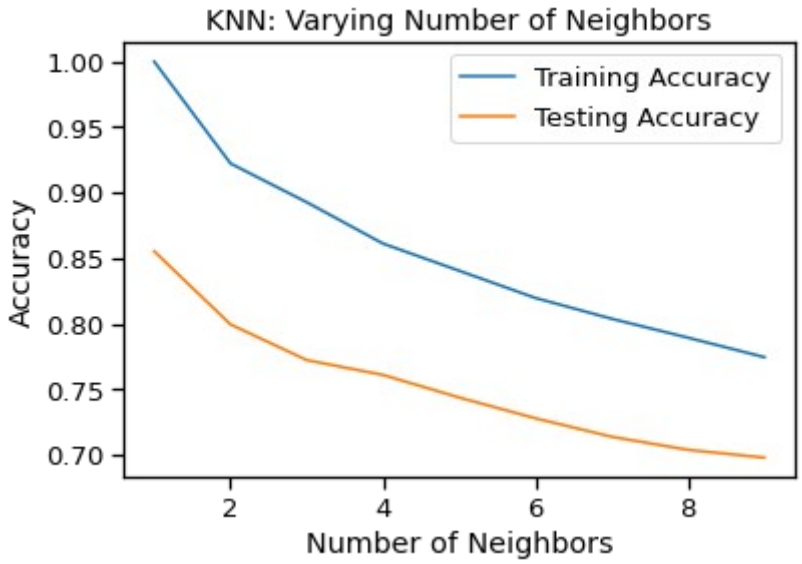
Predictions: [48.90641273 71.55177578], Values: [40. 73.]

Looks like the best way of suceeding the mission is to hire 40 or 70 percent of staff.

Trying to use KNeighborsClassifier to see if I am getting better results**.** Creating arrays for the features and target variable. Fiting the classifier tot he data. Predicting the labels for the X_new and prinited predictions for X_new

Predictions: [100. 0. 100. ... 0. 0. 0.]

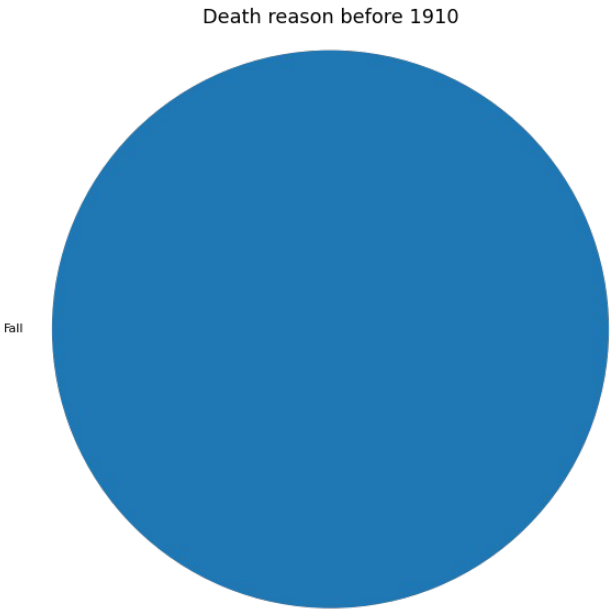
Splitting data and fitting classifier. Checking accuracy for test data. This indicate that KNeighborsClassifier is more efficient model then regression for my case. Plot training accuracies Plot test accuracies



I have to stop here due to time limitation I have for this project.

Results

Goal 1. Problem with currently used data-set is that most of the features representing nominal data. After exploring data on this paper it looks like goal 1 need to be skipped as there is not enough data to support this comparison. This is very clear when we see how many record are in splited dataframe for before 1910 period and from the pie chart



To get goal 1 accomplished it would be much better to analyze climbing data from another part of the world where there is more observations. Unfortunately observations made in pre-computerized years would be spread across various personal diaries so it would be very hard to group it.

Goal 2. Based on analisys when we tried to perform Chi-square analisis. Results were giving us very high numbers that did not tell us much about diference.

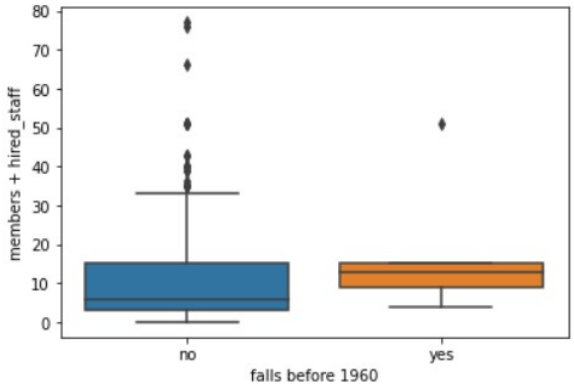
Power_divergenceResult(statistic=240.3846153846154, pvalue=3.242219985667723e-54)

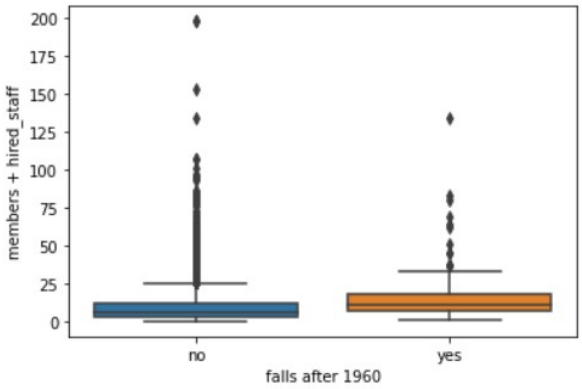
Power_divergenceResult(statistic=9788.692579505301, pvalue=0.0)

When we used library pingouin to do that for us we got few errors like ‘Low count on observed frequencies’ which was displayed for each set of data. There is a difinitally room for improvement on this dataset to get those results without errors. Unfortunatelly Pearson results indicate that there is no difference at all as results are exactly the same.

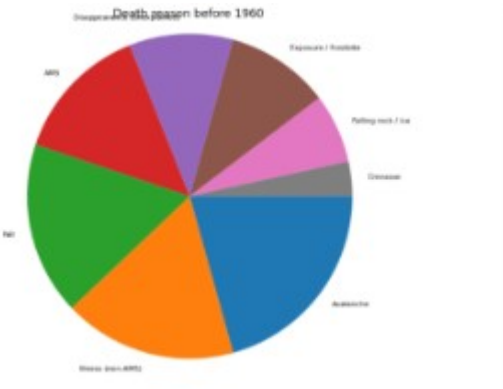
	test	lambda	chi2	dof	pval	cramer	power
0	pearson	1.000000	11.490993	16.0	0.778214	0.210229	0.515164

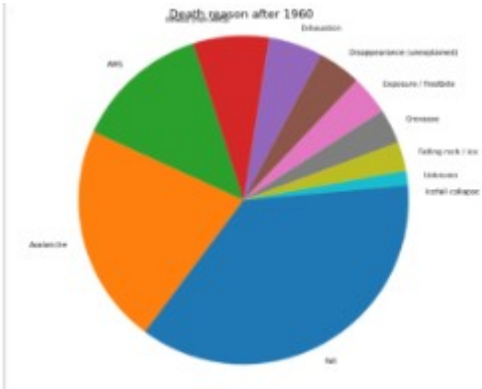
Fortunatelly by looking at graphs falls per members brought idea of comparison between percentage.





Graphs from both periods shows that death reason has changed sligthly. But this is indicating more at apparent effect then significante difference.





Conclusion from above results shows that development of harness and dynamic rope made impact on safety of 0.01% which is not as much as we would expect.

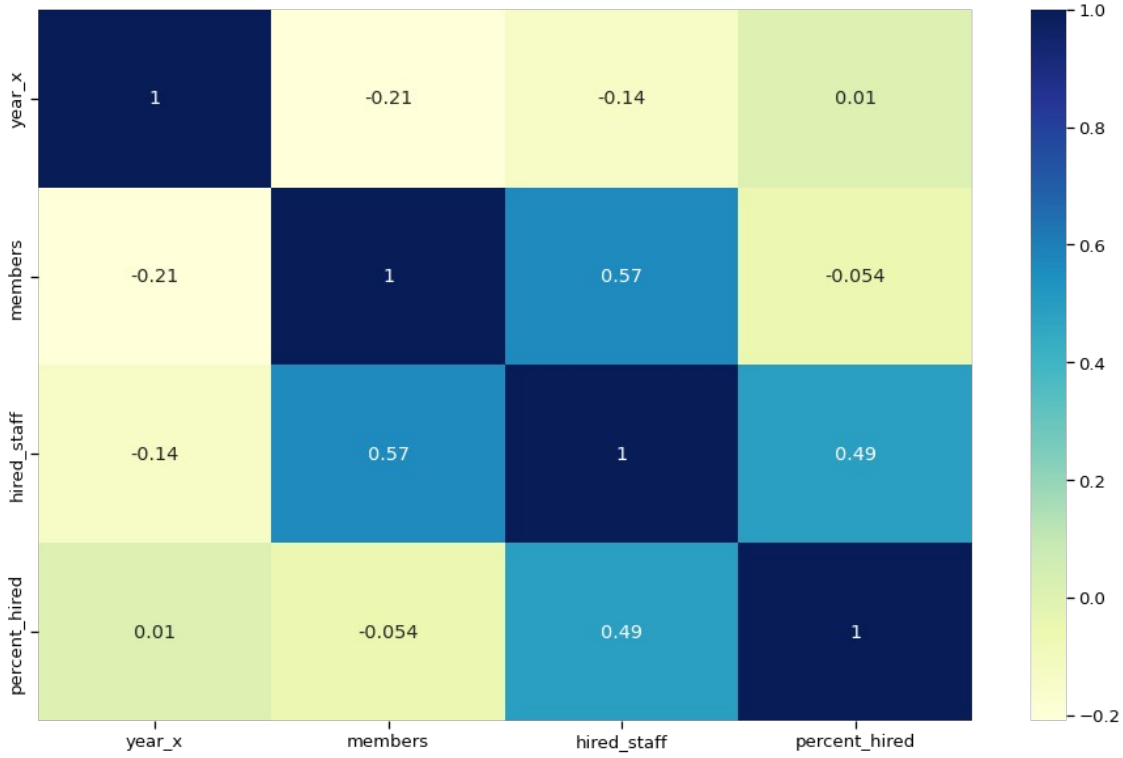
*Things that could be improved but due to time restrain were not.

1. All dataset could be merged based on column Peak_id.

2. To visualise Pie charts percent could be added
3. Peaks_name could be replaced by numerical values
4. Result for death cause is linked to person, removing person_id along with death cause resulting in producing as fall.

Goal 3. Whats the percentage of hired_staff that need to be hired for expedition to be successful?

To get those results I used linear regression and based on scatterplot and scatter matrix there was no obvious line that could be drawn. However heatmap showed clear corelation between members and hired staff.



Decided to split my set for 80:20 ratio and got 42% as accurancy. Fitting model to the data provided me with answer: Predictions: [48.90641273 71.55177578], Values: [40. 73.]. This conclude in answer that the best way to succed the mission is to hire around 50 or 70 percent of staff.

Insights

Goal 1.

- There is not enough observations to provide definite answer if steel carabiner made any difference on climbing safety

Goal 2.

- Death due to fall is actually extreme as climbers are falling on route atttempt basis. Not all falls are dangerous. In dataset fall is listed only as a reason for death but there is no count how many time climbers has fall before they die. One of the reason to die due to fall is not due to break of the rope but most often due to bad protection placing in the rock. As in Nepal we are dealing with Ice climbing it could be due to loose ice.
- Hired staff column and count were skipped, therefore results can be improved.
- Dispite safety new types of rope and harness made contribution to increase popularity of the sport. I assume there are more people surviving those expeditions and can share theres stories across the globe.

Goal 3.

- Regression said 40 or 70 percent I belive this can be decided based on climbers experience. Historically when the nations were desiring to be first that reached the top, expeditions members had different experience, different level of climbing skills as the selection rules for those were not that obvious. That is why I am pretty happy with the result I got. Even with model accurancy 42%.

References

Dataset: https://www.kaggle.com/datasets/majunbajun/himalayan-climbing-expeditions

Convert nominal to numerical values: https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/

Generating more samples: https://machinelearningmastery.com/generate-test-datasets-python-scikit-learn/

Comparing: https://towardsdatascience.com/how-to-compare-two-or-more-distributions-9b06ee4d30bf

Scrapping webpage url: https://quotes.toscrape.com

Hypothesis: https://corporatefinanceinstitute.com/resources/data-science/hypothesis-testing/

Chi-squered: https://www.ling.upenn.edu/~clight/chisquared.htm

Penguin - https://cmdlinetips.com/2020/06/principal-component-analysis-with-penguins-data-in-python/

Book - https://ethanweed.github.io/pythonbook

Supervised learning - https://campus.datacamp.com/courses/supervised-learning-with-scikit-learn/regression-6320c92e-31c3-48fb-9382-6a9169125722?ex=2