Students for Machine Learning in Business

Challenge 2

- Help Predict Wildfires -

Congratulations on deciding to take up the challenge to help the Wildfire Management Branch (WMB), the government branch responsible for managing activities related to controlling and extinguishing wildfires in Alberta. You will be assisting the WMB with the presuppression of wildfires. “Presuppression” refers to the practice of anticipating the resources required for fire suppression before fire occurrences are known.

What do we mean by resources? The province manages several crews of highly trained individuals, as well as equipment such as Air Tankers, Helicopters, Bulldozers, etc. Since some of this equipment may be contracted, this resourcing often incurs high monetary costs, and having additional information about future conditions can allow planning to be done proactively.

Current forecasts used for presuppression planning give insights into the severity of potential fire outbreaks, but not ignition likelihoods. Being able to make predictions about ignition likelihoods would allow the WMB to optimize resource allocation, which is hypothesized to be able to reduce resourcing costs while maintaining a similar level of risk to current practices.

Your challenge is to develop a machine learning algorithm that can forecast the likelihood of fires to occur within a forest area after weather measurements.

This challenge will run from **March 1, 2022 to March 22, 2022**. Your model must be submitted before the deadline to count. The submission details are outlined below in this document. If you're brand new to coding or machine learning, don't feel overwhelmed — we have a comprehensive guide (sent alongside this rules document)! Please feel free to ask any questions, and attend our Q&A sessions via our Discord!

Please use the Challenge 2 Guide document for a walkthrough and tips!

Deadline: March 22, 2022

1st place: $500
2nd place: $350
3rd place: $150

Challenge Rules:

- ❏ Work in a group of 2 people or individually. Remember, collaborating shares the load, but also shares the rewards.
- ❏ Code must be written in Python 3 (modules/packages found on www.pypi.org are permitted).
- ❏ Must be submitted by end of the day (11:59pm MST) on the submission deadline: **March 22, 2022**. We will not accept any late submissions.
- ❏ Code must be submitted on our website, SMLB.org.
- ❏ Code must output in the correct format as covered in the Submission Details section.
- ❏ Submissions must be machine learning models trained only with the dataset provided.
- ❏ Code will be marked based on the lowest error in predictions using the test data set (unreleased) using an AUC metric. More information on marking can be found within this document.
- ❏ Winners will be contacted about payment details promptly after the challenge ends. Winners must provide payment information by April 30, 2022 or the prize will be voided.
- ❏ By participating, you agree not to share the data and only use materials for the educational purpose of this competition.

SMLB is meant to be a collaborative effort between individuals hoping to learn more about machine learning and its applications in the business world, as such, we intend to post examples of winning models on our website. As such, any code submitted for the SMLB Challenge 2 must be under an MIT License, which is described as 'A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.' by choosealicense.com.

Challenge 2 Guide (LINK)

DATA

The first step is to download and understand the data. You can open the data in excel or a similar spreadsheet program. Take your time to understand what each column represents!

Important note: The raw data comes from various weather stations across Alberta, with often multiple weather stations per forest area. The average values shown per forest area are averages of the readings from the active weather stations in that forest area at 3pm on the given date. The test data is the set of data for fires during the years of 2017 and 2018. The training data that you are provided is the fired data between the years of 1980 and 2017.

Data Legend: **All data is per the given date and forest_area.**

Unnamed: 0: Arbitrary index column.
Date: Date of data row.
Forest_area: An ID index for areas in Alberta.
avg_Dry_bulb_temperature: Average air temperature in degrees celsius.
avg_Dew_point: Average dew point in degrees celsius.
avg_Relative_humidity: Average air moisture, expressed as a percentage.
avg_Rain_mm: Average rainfall in millimeters during that day.
avg_Snow_cm: Average snowfall in centimeters during that day.
avg_Hail_mm: Average hail deposition in millimeters during that day.
avg_Precipitation_mm: Sum of rain, snow and hail columns in millimeters.
avg_Wind_speed_kmh: Average wind speed in kilometers per hour.
avg_Wind_gust_kmh: Average gusting wind speed in kilometers per hour.
avg_Wind_azimuth: Average direction of wind in degrees from north. Increasing clockwise.
avg_Ffmc: Represents fuel moisture of forest litter fuels under the shade of a forest canopy.
avg_Dmc: Represents fuel moisture of decomposed organic material underneath the litter.
avg_Dc: Represents drying of fuels deep in the soil.
avg_Isi: A measure of expected rate of fire spread, depends generally on wind speed.
avg_Bui: Represents the amount of fuel available for burning. Depends on DMC and DC.
avg_Fwi: A federal fire danger index.
avg_Edr: A provincial fire risk index.
Is_fire: 0 or 1 representing no/yes if a fire occurred on that day in that forest area.
Is_fire_tmrw: 0 or 1 representing no/yes if a fire occurred the next day in that forest area.
Is_fire_2days: 0 or 1 representing no/yes if a fire occurred in two days in that forest area.
You are responsible for creating an algorithm that predicts the variables “is_fire,” “is_fire_tmrw,” and “is_fire_2days”.

Submission Details

To submit your algorithm, upload your .ipynb file to the SMLB.org website. Please make your submission as clear as possible with comments and dependencies attached. Please be **very** clear on everything that is needed to run the code and in which order. Include a list of the python packages and their links on https://pypi.org/ with the version number that is required to run the submission. The organizers may contact you for clarification after submission. Please keep in mind that any edits to the source code will not be permitted after the deadline, so ensure that the code will run on a machine other than your own before submitting. Send yourself the .ipynb on a different machine and walk through the process of running it on the training set so that you are sure the organizers will be able to do so as well.

IMPORTANT - Code must follow this format: (feel free to use this template exactly):

```
# ================== #
# Variables you need! #
# ================== #

myPredictions = None
# You need this variable! Please store your predictions for the test in this variable!

nameOfTrainFile = "[replace_with_the_data].csv"
# This is the name of the dataset we provided!
# You can use this to train your model!

nameOfTestFile = "[we_will_replace this within your data].csv"
# You won't have this file name!
# We will replace this variable with the test data during evaluation.
# This will be in the same format of the train data, except the "is_fire" column is missing!

...
# ================== #
# Your Code Runs Here #
# ================== #
...
```

```
# ================== #
# Formatting Results  #
# ================== #

# We are very specific on the formatting for your results ( stored in 'myPredictions' variable)!
# It should be the following format:

myPredictions = [
  [P1, P2, P3, ... , PN], # Predictions for is_fire
  [P1, P2, P3, ... , PN], # Predictions for is_fire_tmrw
  [P1, P2, P3, ... , PN], # Predictions for is_fire_2days
  [P1, P2, P3, ... , PN], # Predictions for is_fire_3days
  [P1, P2, P3, ... , PN], # Predictions for is_fire_4days
  [P1, P2, P3, ... , PN], # Predictions for is_fire_5days
  [P1, P2, P3, ... , PN], # Predictions for is_fire_6days
  [P1, P2, P3, ... , PN], # Predictions for is_fire_7days
]

# Each of the elements within this array are 1D arrays themselves!
# Now, within each of the 1D arrays, the values are between 0 and 1.
# They represent the 'probability of a fire' given the parameters for that
# corresponding row in the test data.

# Eg:

# Suppose:

myPredictions[0] = [0.654, 0.923, 0.231, 0.122]

# this means for the first element within the myPredictions array, (so the "is_fire" predictions),
# you believe 0.654, 0.923, 0.231, 0.122 are the corresponding probabilites.

# This means that given the parameters from the first row of test data,
# you believe there is a 65.4% chance of fire!

# For the second set of parameters, you believe theres a 92.3% chance!
```

If needed, instructions on how to set up this format are in the Challenge 2 Guide.

Furthermore, feel free to use our discord server to ask any questions you have!

How will your code be marked?

The marking scheme is simple; we have withheld test data in the same format as the training data provided to you. This test data excludes the target variables. We will run this test data through your code, and in return, will obtain the required output. Your submission should be able to predict the target variables, “is_fire,” “is_fire_tmrw,” and “is_fire_2days” as a floating point value between 0 and 1. This value represents your predicted probability of fire on a given day.

In your program, be sure to include some variables which we can change to test your model (shown in the image above). Your algorithm will be marked based on the AUC, a metric based on the ROC curve. This is an overall test of a classifier model that accounts for different probability cutoffs for when fires are likely to occur. You can read more about AUC here.

To elaborate, here is an example:

Suppose your program is given a test data containing 4 rows.
For the "is_fire" column, your algorithm predicts the values [0.654, 0.923, 0.231, 0.122] for each of the rows within this testing dataset.

Now, consider that in reality, the first 2 rows correspond to days which had a fire, and the remaining 2 rows correspond to days without fire.

Your error would be computed by:

is_fire (not provided in the testing data)	1	1	0	0
Your predictions	0.654	0.923	0.231	0.122

We will generate an ROC curve by comparing these two arrays of data, and then finding the AUC. Your final mark will be a weighted sum of your AUC scores. The formula for your final score is as follows:

$$Final\ Score\ =\ 0.7\ *\ (is\_fire\ AUC)\ +\ 0.2\ *\ (is\_fire\_tmrw\ AUC)\ +\ 0.1\ *\ (is\_fire\_2days\ AUC)$$

Note that your score for the is_fire prediction is weighted far more heavily than your other two predictions.

This calculation will be done for all eight prediction columns, and the winner of the challenge will have the lowest sum of absolute errors. A partial submission will receive the minimum possible value (0) for the AUC of the prediction of a particular variable. This means that the maximum possible score is 1.0.

Additional Tips:

The process of developing algorithms often requires much tinkering and bug fixing. Regardless of your skill level, it is recommended that you try and get a working prototype before you attempt to get too ambitious. Get something that works before you make something that is perfect!

For Beginner Programmers:

- ❏ A detailed walk-through is provided for complete beginners who wish to receive instructions on how to progress!
- ❏ Install libraries such as Pandas and Scikit-Learn libraries which are crucial to machine learning.
- ❏ What is a logistic regression? https://www.youtube.com/watch?v=yIYKR4sgzI8
- ❏ Before moving ahead, split your data into 2 parts: Training and validation! You can learn this here https://youtu.be/fwY9Qv96DJY (Watch up to 3:43, the rest is irrelevant!)
- ❏ Check out Scikit-Learn's "Logistic Regression" module. See what methods it has to offer
    - ❏ Examine the various hyperparameters which you can fine-tune
- ❏ Final Tip: Do all the factors such as (Type, Fuel, Acc, etc.) necessarily make our predictions more accurate?
    - ❏ Some factors may in fact reduce the model's accuracy due to their lack of impact/correlation on the choice!

For Experienced Programmers:

- ❏ Look at the data closely! Data is not uniformly collected from all regions during all times of the year, and this evolves over the course of the training data as weather stations opened and closed, and more data was collected.
- ❏ Consider the significance of seasonality.
- ❏ Consider how you will manage NA values.
- ❏ Consider the role of autocorrelation.

Good Luck! - Students for Machine Learning in Business Team