**Avalanche Forecasting**

**Team- 52**

Submitted by:

Dwaipayan Dinda(West Bengal University Of Technology)

Suresh Akanksh(Gitam University)

Arunav Das(DBE,Delhi University)

Table of Contents

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | |  | **INTRODUCTION** | |  |  |  |
|  | |  | 1.1 Overview | |  |  |  |
|  | |  | 1.2 Purpose | |  |  |  |
| **2** | |  | **LITERATURE SURVEY** | |  |  |  |
|  | |  | 2.1 Existing problem | | |  |  |
|  | |  | 2.2 Proposed solution | | |  |  |
| **3** | |  | **THEORITICAL ANALYSIS** | |  |  |  |
|  | |  | 3.1 Block diagram | |  |  |  |
|  | |  | 3.2 Hardware / Software designing | | |  |  |
| **4** | |  | **EXPERIMENTAL INVESTIGATIONS** | | |  |  |
| **5** | |  | **FLOWCHART** | |  |  |  |
| **6** | |  | **RESULT** |  |  |  |  |
| **7** | |  | **ADVANTAGES & DISADVANTAGES** | | |  |  |
| **8** | |  | **APPLICATIONS** | |  |  |  |
| **9** | |  | **CONCLUSION** | |  |  |  |
| **10** | |  | **FUTURE SCOPE** | | **`** |  |  |
| **11** | |  | **BIBILOGRAPHY** | |  |  |  |
|  | |  | **APPENDIX** | |  |  |  |
|  |  | | A. Source code | |  |  |  |
|  |  | |  |  |  |  |  |

**INTRODUCTION**

**Overview:**

The Avalanche forecasters use many different types of methods basically depending on the mountain zone they reside in, the size of their operation, and the availability of information, but the basic fundamentals are all the same. First it starts with observations and with the observations, like any other modelling, we try to build a model out of it thus predicting when a possible avalanche might occur depending on the parameters used.

**Purpose**

Between 1885 and 1911, more than 250 men lost their lives in avalanches in the Rogers Pass area of Glacier National Park.

The most crucial part of avalanche forecasting is evaluating the snow, often by digging snow pits and looking for specific weaknesses present in the area. But the problem is, nobody would want to dig a pit on the steepest part of the slope and risk their lives, and also that what holds true for one location may not be true for another location.

So to prevent risking the lives of these people and people around in a locality prone to avalanche, and so that we can inform if there’s any chance for an Avalanche, we try to collect various data, draw a correlation based on the parameters used and then try to form a model which would tell us if an Avalanche is forthcoming.

**LITERATURE REVIEW**

**Existing Problem**

One of the worst accident happened was during the night of March 4, 1910, when a crew of 58 men was faced by an Avalanche. The crew were shoveling snow from the tracks of the Canadian Pacific Railway, the result of an earlier avalanche on Cheops Mountain on the opposite side of the valley. But while performing their work of cleaning the snow, unfortunate enough to their fate, they were buried by a large avalanche that came thundering down the steep slopes of aptly named Avalanche Mountain. All but one of the men perished. Soon afterward construction began on a series of snow sheds designed to protect the railway.

Thus, if there had been some mechanism through which one could tell if an avalanche is coming or there’s a possible chance of avalanche, we could at least be aware, and inform others so that they can be safe and stay protected.

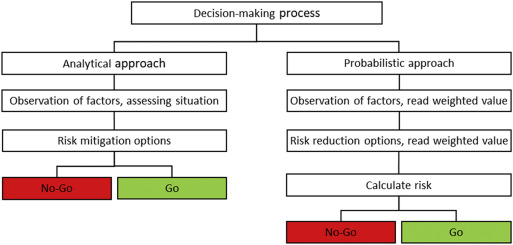
**Proposed Solution**

We collected the data and based on the observations recorded over a period of time, we prepared a model which would at least give us an idea if there is any chance of an Avalanche based on the parameters like Temperature, relative humidity, & snow depth.

**THEORETICAL ANALYSIS**

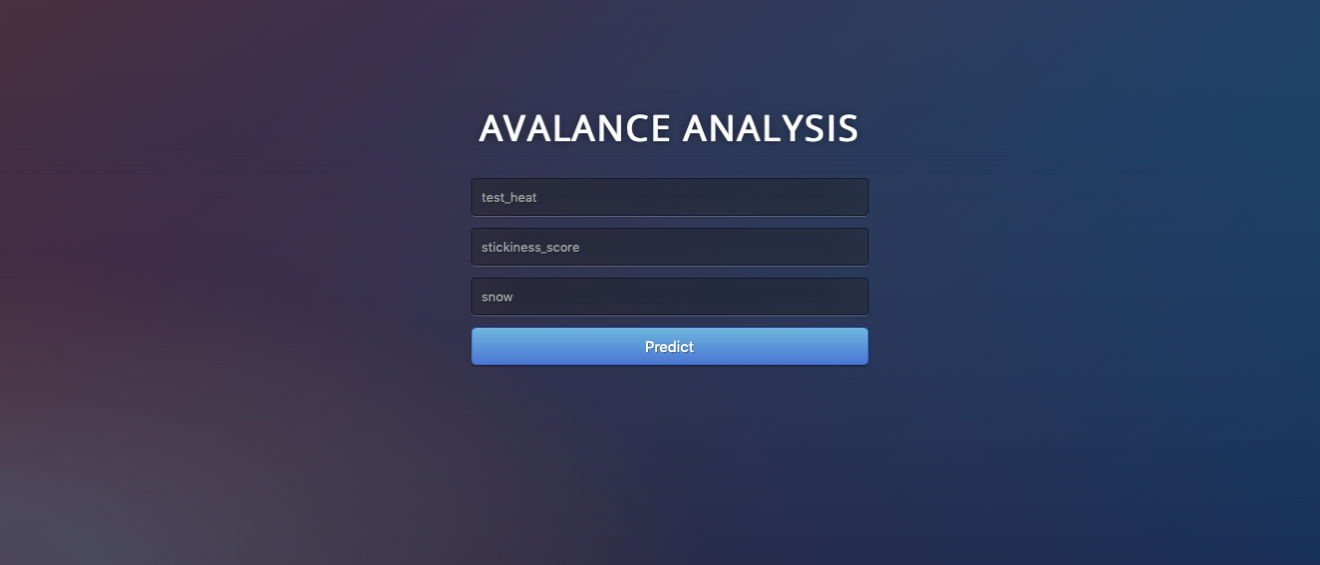
Block diagram:

The data has been collected from Kaggle whose origin can be traced back to <https://www.nwac.us/data-portal/location/stevens-pass/>

****

**Hardware / Software designing**

Using Python platform, we create a regression based Machine Learning model, then verify the accuracy and and then finally deploy our model to predict the outcome. To launch it to a webpage we used FLASK, created an html page and deployed our model on the same.

While training the model, various regression algorithms have been tried and finally we prepared a linear regression model to predict the chance of predicting an Avalanche.

**Deployment**

The model is deployed as a Web Application that calculates the avalanche danger level prediction and predicts the chances of Avalanches occurrence based on input values that correspond to a few days of weather data.

**FLOWCHART:**

**RESULT:**

Finally we have predicted the chance of occurring an Avalanche given the respective inputs.

**ADVANTAGES & DISADVANTAGES**

There are a lot of advantages to the model which we built, as we have been able to tell the chance of an Avalanche occurring given the respective inputs. We can well aware the people and inform them in case there is a high chance of Avalanche occurring in an area based on the parameters we have used. This way, the model can alert people living in Avalanche prone areas. Suppose, if the model predicts a 50% chance of occurring an Avalanche, we can ask them to take preventive measures. And if the prediction rises above 70-80%, we can assume, there is a high chance of an Avalanche occurrence and thus can prevent people from going into that area. For more than 90%, more preventive steps should be taken.

The only disadvantage is that with less amount of data, we may not be accurately predicting the chance of Avalanche occurrence but still it is always better to at least predict something and take precautionary measures than not know of anything.

**APPLICATIONS**

We can use this toinform crews working in an Avalanche prone area.

Alert the snowboarding enthusiasts of any possible Avalanche in case there is a high chance of Avalanche occurring in an area.

It will be helpful in taking precautionary and preventive measures.

Thus it can be used to forecast avalanches.

**CONCLUSION**

We have built a model, then trained the model and performed regression algorithms to the same and thus predicted the chance of avalanche percentage based on the input parameters. Further, we have deployed the same model on a web page where using FLASK. Thus, we have made an Avalanche forecasting model where we can measure the chance of an Avalanche occurence.

**FUTURE SCOPE**

We can further aim to improve this project by improving the model's performance by adding more parameters and more data into it. If we can gather more data from the past or from this point forward if we try and collect more and more data and add it to our model accordingly, we can achieve a great deal of accuracy. With more data points, we would be able to predict the model with more accuracy. If we can gather a data for a period of almost 50 years or more, we might be able to predict the exact time of the Avalanche occurrence and if we can integrate a few more geographical parameters, we might be able to tell the impact of the Avalanche as well. Also, if in future we can collaborate and integrate it with weather satellites and other geo-sensing tools, this could go a long way into the future and help in much earlier predictions within a fraction of time.

**BIBILOGRAPHY**

* https://www.kaggle.com/vinaynomula/avalanche-forecasting-prediction
* https://www.pc.gc.ca/en/pn-np/mtn/securiteenmontagne-mountainsafety/avalanche/prevision-forecasting
* https://www.voile.com/blog/avalanche-forecasting-101/
* https://whyfiles.org/105avalanche/4.html
* https://www.wiegele.com/whats-new/latest-news/avalanche-forecasting-101-by-scott-thumlert
* https://www.voile.com/blog/avalanche-forecasting-101/

**APPENDIX:**

Source Code

|  |
| --- |
| import numpy as np |
|  | from flask import Flask, request, jsonify, render\_template |
|  | import pickle |
|  |  |
|  | app = Flask(\_\_name\_\_) |
|  | model = pickle.load(open('avalance.pkl', 'rb')) |
|  |  |
|  | @app.route('/') |
|  | def home(): |
|  | return render\_template('index.html') |
|  |  |
|  | @app.route('/predict',methods=['POST']) |
|  | def predict(): |
|  |  |
|  | int\_features = [int(x) for x in request.form.values()] |
|  | final\_features = [np.array(int\_features)] |
|  | prediction = model.predict(final\_features) |
|  |  |
|  | output = round(prediction[0], 2)/10000 |
|  |  |
|  | return render\_template('index.html', prediction\_text='AVALANCE WILL BE % {}'.format(output)) |
|  |  |
|  | @app.route('/predict\_api',methods=['POST']) |
|  | def predict\_api(): |
|  |  |
|  |  |
|  | data = request.get\_json(force=True) |
|  | prediction = model.predict([np.array(list(data.values()))]) |
|  |  |
|  | output = prediction[0] |
|  | return jsonify(output) |
|  |  |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | app.run(debug=True) |