DATA MINING - FINAL REPORT

Forecasting the precipitation type based on other meteorological data

Isabela Negoita S1089659
Radboud University DECEMBER 2023

Contents

1 Abstract

The project aimed to develop a predictive model for determining precipitation types based on other meteorological factors. The classification methods used are Decision Trees and K-Nearest Neighbors, whose parameters have been tuned to achieve optimal performance. In order to overcome the imbalanced dataset, Synthetic Minority Oversampling has been applied, which led to an overall high performance of the model. It was found that the Decision Tree Classifier has a better performance than KNN, indicating that it may be more suitable for the task at hand.

2 Introduction

2.1 The research problem

People's daily activities and living conditions have always been greatly influenced by the weather conditions they have to carry them out in. Due to the importance of being able to prepare for meteorological conditions in advance, weather forecasting has been an area of interest for humans ever since ancient times. The methods have been evolving alongside humankind, going from ancient methods such as predicting the weather from cloud patterns [Wik(2023)] , to making use of today's sophisticated prediction methods in order to make accurate predictions based on historical data[Reilly(2023)].

This project attempts to make use of proven data mining techniques in order to build models which can accurately forecast the weather. More specifically, the task at hand consisted of predicting the precipitation type on the basis of other recorded meteorological factors such as wind bearing or visibility.

2.2 Related work

Weather forecasting has been a significant area of research in Machine Learning in recent years, due to the proven efficacy of such models in comparison conventional modelling methods. The main inspiration behind this research was the project carried out during last year's Frequentist Statistic course. At that time, using the same dataset, an attempt was made to try to predict the apparent temperature. Due to a lacking skillset, the analysis was deemed unsatisfactory and no conclusions could be drawn. This time around a different task was tackled, namely predicting precipitation type using the other meteorological factors.

Due to the increased interest in the topic, various research papers and projects can be found across the internet, which attempt to forecast different aspects of the weather using a multitude of data mining and machine learning techniques. For example, George uses KNN and DTC in order to predict daily rainfall[George(2022)]. The analysis provided in the cited article served as guidance in choosing the algorithms used in this project.

3 Dataset and Preprocessing

3.1 Presenting the data

The project makes use of the Weather dataset which can be found on Kaggle[Muthukumar(2017)]. The CSV file contains 96453 hourly weather entries, spanning across 10 years, from January 1st 2006 to December 31st 2016. The variables represented are:

- *formatted_date*: The date and time of collection
- *summary*: short summary of the weather at the given moment
- *precip*: The precipitation type
- *temp*: The real temperature
- *apparent_temp* The apparent temperature
- *humidity*: The humidity
- *wind_speed*: The speed at which the wind is blowing
- *wind_bearing*: The direction from which the wind originates

- *visibility*: How far ahead one can see
- *air_pressure*: The atmospheric pressure
- *daily_summary*: An overall summary of the day's weather

A column titled ”loud_cover” is also present, but it is not clear what it represents and it was populated only with 0's, therefore it was clear from the start that it would be removed during pre-processing.

3.2 Preparing the data

First of all, the data from the CSV file was retrieved using the pandas library. From the beginning, ”loud_cover” was removed, alongside the column containing the time and date of collection, as it was deemed irrelevant to the analysis.

Upon further inspection, the ”daily_summary” and ”summary” variables were removed as well, as it was decided that they had too many distinct values which would have unnecessarily hindered the classification task.

In order to be able to run analysis on the categorical target variable ”precip”, its values were mapped to numerical values as follows:

Listing 1: Transforming the categorical variables

```
weather_mapping = {'none': 0, 'snow': 1, 'rain': 2}
weather['precip'] = weather['precip'].map(weather_mapping)
```

The distribution of the classes of the target feature ”precip” in the original dataset was the following:

- *Rain*: 85224
- *Snow*: 10712
- *None*: 517

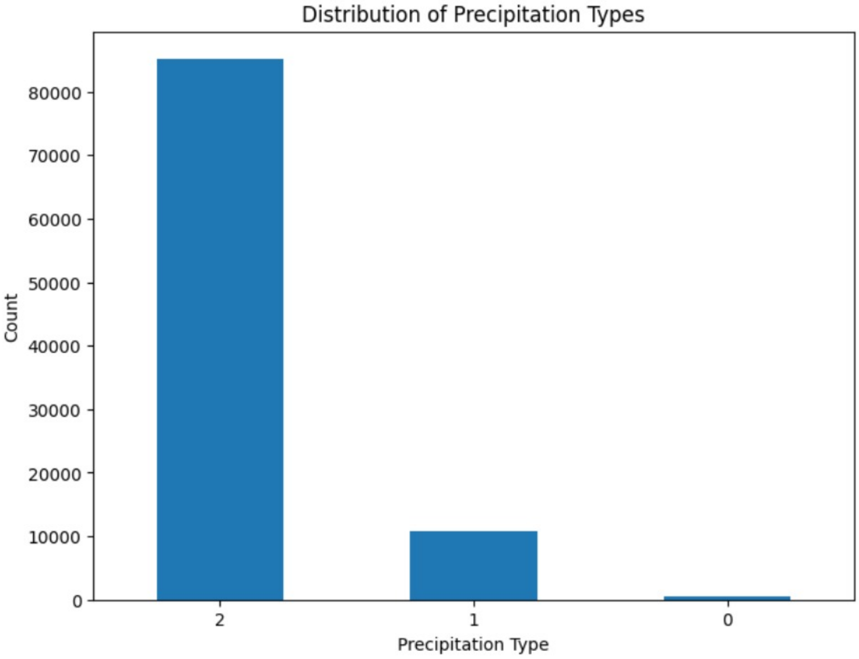


Figure 1: The original class distribution

In order make up for the obvious class imbalance, the Synthetic Minority Oversampling Technique, or SMOTE, belonging to the python library *imblearn*, was applied. The technique synthesises new instances for the minority classes through linear interpolation[Gee(2023)]. Furthermore, taking into account the large size of the ”weather” dataset and the high computational costs of K-Nearest Neighbours, shortened as KNN, it was decided to re-sample the data, randomly selecting 1000 instances of each class to use in the training of the models.

4 Methods

A major impediment when working with this dataset for the Frequentist Statistics project last year was that the regression methods used at the time assumed linearity. Therefore, for this project it was decided to employ classification algorithms which would be able to handle a non-linear decision boundary. Subsequently, the choices made were Decision Tree Classifier and K-Nearest Neighbours.

4.1 Decision Tree Classifier

Decision Trees are built by repeatedly splitting the data into purer subsets, based on a purity measure (in the present case, the GINI index was used). The first step in building an accurate DTC model is finding the optimal tree-depth, at which complex patterns in the data are captured without risking overfitting. In order to find this value, 10-fold cross-validation was used, resulting in an optimal depth of 6 levels, which was then used to build the decision tree, which can be seen in 2.

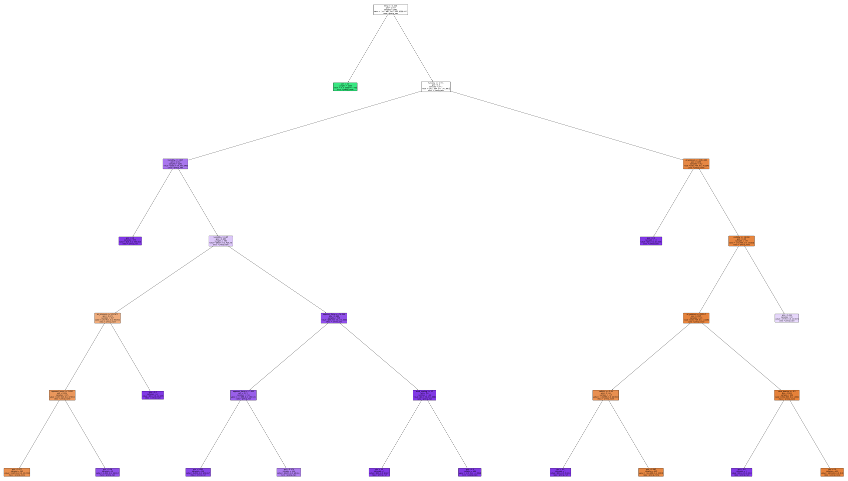


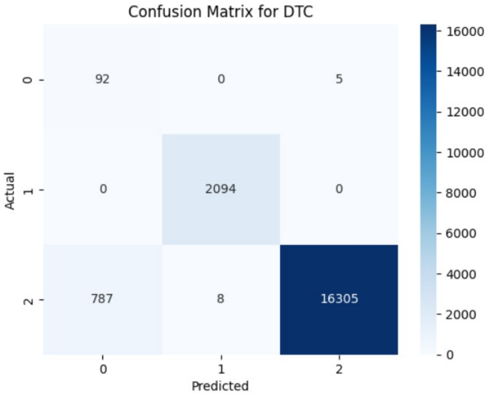
Figure 2: The Decision Tree

4.2 K-Nearest NeighbourS Classifier

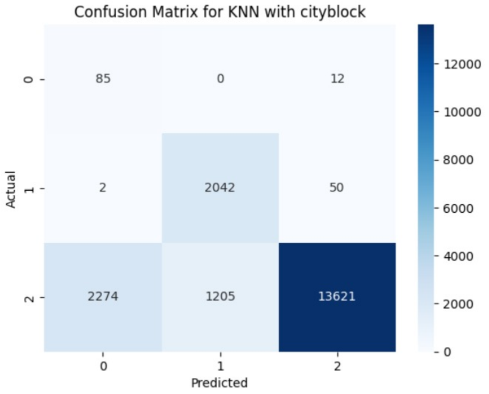
KNN is an algorithm which ”tries to predict the correct class for the test data by calculating the distance between the test data and all the training points”[Christopher(2021)]. In order to achieve the best performance, it is essential to determine the optimal number of neighbors used in the classification, as well as the appropriate distance measure. As a means to determine these, Leave-One-Out Cross-Validation was applied for three distance metrics: euclidean, cosine and cityblock. The results indicated that the optimal k across the board was 1, and since cityblock yielded the lowest classification error, it was chosen to build the final model.

5 Results

In order to get a representative overview of the performance of the algorithms, the confusion matrices and classification reports were computed. It can be observed that among the two algorithms, DTC performed better by a small margin.



(a) DTC Confusion Matrix



(b) KNN Confusion Matrix

(a) Classification report of DTC

	none	snow	rain
Precision	0.10	1.00	1.00
Recall	0.95	1.00	0.95
F1-score	0.19	1.00	0.98
Accuracy		0.96	

(b) Classification report of KNN

	none	snow	rain
Precision	0.04	0.63	1.00
Recall	0.88	0.98	0.80
F1-score	0.07	0.76	0.88
Accuracy		0.82	

6 Discussion

Initially, the model was trained on a stratified sample taken out of the entire dataset. This led to a high accuracy, however upon the inspection of the classification report, it was found that the recall was low (<0.5 for class 0 / no precipitation) and no instances of class '0' were classified correctly, which indicated that the model was biased towards the majority class.

The decision to address the class imbalance through SMOTE led to an overall better performance of the models. The confusion matrices and classification reports provided a detailed insight into the models' performance on each class. The DTC demonstrated strong precision, recall, and F1-score for all classes, resulting in an impressive overall accuracy of 96%. On the other hand, KNN, while achieving a respectable accuracy of 82%, showed lower precision and recall for class 0. DTC outperforming KNN was expected, as decision trees are generally better at capturing non-linear boundaries and less sensitive to outliers. The accuracy of KNN could potentially be improved by using a larger training sample, however this was not feasible in this project due to its high computational costs.

7 Conclusion

In conclusion, this project revealed the dependence between the precipitation type and the other recorded factors. The models chosen performed with high accuracy, reinforcing the idea that the algorithms suited the task at hand well,as well as revealing the correlation between the features used in the prediction model. Moreover, the task highlighted the importance of analysing the nature of the data, specifically the distribution of classes in the training sample. By synthesising instances of the minority class, the issue of overfitting was remedied, succeeding in building accurate, unbiased prediction models which can be generalised well.

References

[Wik(2023)] 2023. https://en.wikipedia.org/wiki/Weather$_{f}orecasting$

[Gee(2023)] 2023. https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/

[Christopher(2021)] Antony Christopher. 2021. K-Nearest Neighbor. https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4

[George(2022)] Allu Niya George. 2022. *Decision tree V/s KNN in rainfall predictionnbsp;: Python, R, Weka.* https://medium.com/@alluniya26/decision-tree-v-s-knn-in-rainfall-prediction-python-r-weka-6e4a9c6c66f6

[Muthukumar(2017)] J. Muthukumar. 2017. *Weather dataset.* https://www.kaggle.com/datasets/muthuj7/weather-dataset

[Reilly(2023)] Jon Reilly. 2023. *Using machine learning for accurate weather forecasts in 2023.* https://www.akkio.com/post/weather-prediction-using-machine-learning