# Goal

Our project involves screening clients of Smurf origin who are applying for a luxury hotel in the Bahamas. Smurfs are known for their extravagant spending, tendency to cause damage to hotel property and staff, and overall rowdy behavior.

The challenge is to identify high-risk Smurf clients and prevent them from gaining admission to the hotel to minimize potential losses and maintain the hotel's reputation for excellence.

As data scientists, we are responsible for analyzing the available data and drawing insights that can inform the screening mechanism. This report outlines our methodology, findings, and recommendations for the hotel management.

# Methodology

we collected data on Smurf clients who had stayed in the hotel in the past, including information about their demographics, past behavior, and any damages they caused. This given data was then transferred to two files named “train.csv” and “score.csv”.

These files were not ready to be used on the models. They contained empty rows, duplicates, outliers, or sometimes just wrong information. These rows have been removed or modified from the dataset.

Categorical features have been split up into two columns. Columns that contained true or false values were transformed into 1’s and 0’s. These operations were conducted to ensure that the models have enough accurate information.

# Findings

For the ‘outcome\_profit’ we tried several models and eventually looked which one was best.

|  |  |  |
| --- | --- | --- |
| Model | Train | Validation |
| PolynomialRegressor | 37.0% | 34.1% |
| KNearestNeighbor | 35.0% | 34.3% |
| DecisionTree | 83.6% | 64.6% |
| RandomForestRegressor | 82.5% | 78.9% |
| GradientBoostingRegressor | 85.3% | 75.8% |

As you can see out of this table, we tried 5 different ways to determine the outcome profit. The eventual winner was randomForestRegressor. So this is the model we used to fill in outcome profit.

Next up on the list was outcome damage inc. Just like with the outcome profit we tried different models. As you can see on the graph underneath we came to the conclusion that the polynomial was the best with a score of 69. This is not a lot but it was the best we could do with the data we had at the time.

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

After this we only had to do the different models for outcome damage amount. Here we struggled a lot since the results were just utterly bad and we had no idea how to fix this.

|  |  |  |
| --- | --- | --- |
| Model | Train | Validation |
| PolynomialRegressor | 12.2% | 6.1% |
| KNearestNeighbor | 8.7% | 5.0% |
| DecisionTree | 8.5% | 4.6% |
| RandomForestRegressor | 16.1% | 6.1% |
| GradientBoostingRegressor | 68.8% | -0.3% |

As you can see the results weren’t all the best but we went for RandomForestRegressor since this one had a validation score of 6.1.

# Recommendation/conclusion

The dataset contains information about Smurf clients who have applied for an exclusive hotel in the Bahamas, including their predicted overall revenue, the profit outcome, and the damage amount outcome. The aim of the analysis is to identify the top 150 Smurf clients based on their overall predicted revenue and save the results in a new CSV file named 'top\_150\_smurfs.csv'.

We created a new column named 'overall\_predicted\_revenue' by subtracting the 'outcome\_damage\_amount' column from the 'outcome\_profit' column. This column represents the total revenue that the hotel is predicted to earn from each Smurf client after taking into account the potential damages they may cause.

Next, the DataFrame is sorted in descending order based on the 'overall\_predicted\_revenue' column using the 'sort\_values' method with the 'ascending' parameter set to 'False'. This sorts the dataset from the highest predicted revenue to the lowest.

Now the hotel knows which clients are the best according to our model and can anticipate further decisions on this data.