BOLT UBC BOOTCAMP

Team Name : SMRH

Team Members:

- Srijan Sanghera
- Muzammil Chunawala
- Ruhani Mittal
- Heral Kumar (DNP)

Peak Mountain Resort Overview:

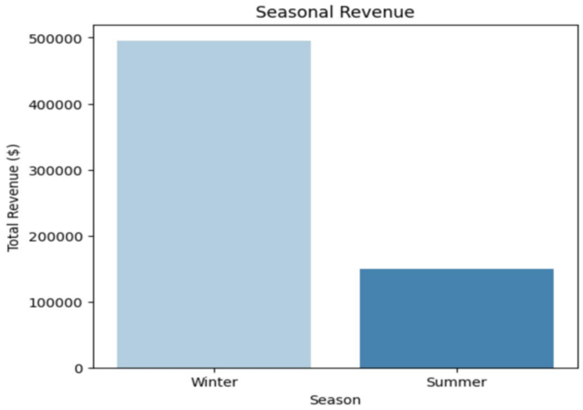
Company Background

- Peak Mountain Resort is a medium-sized, year-round resort known for winter sports, summer adventures, and family-friendly activities
- Originally focused on skiing and snowboarding, the resort expanded to include hiking, mountain biking, zip-lining, climbing, and educational exhibits to adapt to industry trends and financial pressures.
- Despite its strong brand and guest engagement, the resort faces challenges such as aging infrastructure, rising operational costs, and safety risks.
- To stay competitive, Peak Mountain Resort seeks cost-effective, data-driven solutions to improve guest safety, emergency response, and financial sustainability.

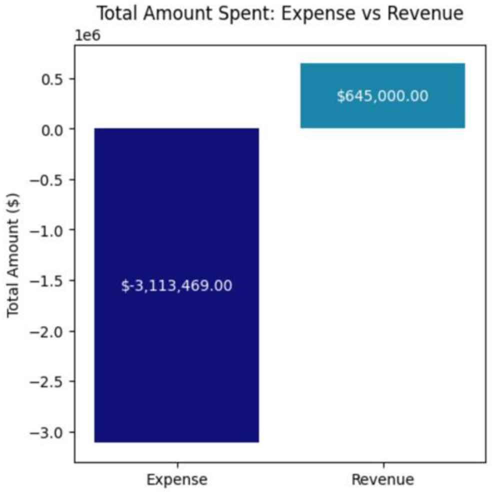
Business Model

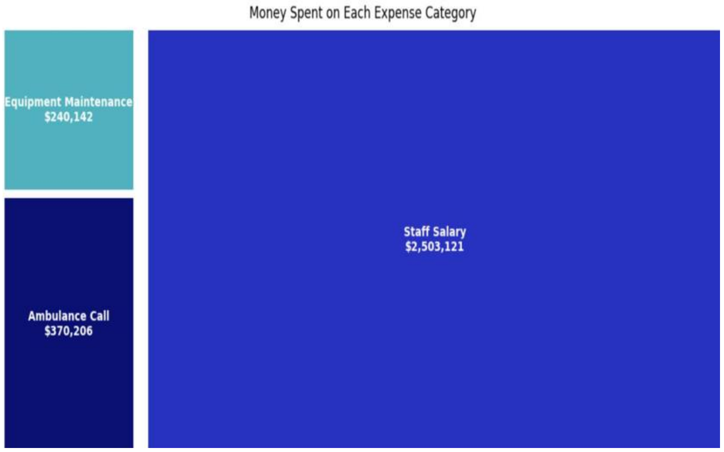
Peak Mountain Resort operates on a seasonal revenue model, generating earnings from winter and summer activities:

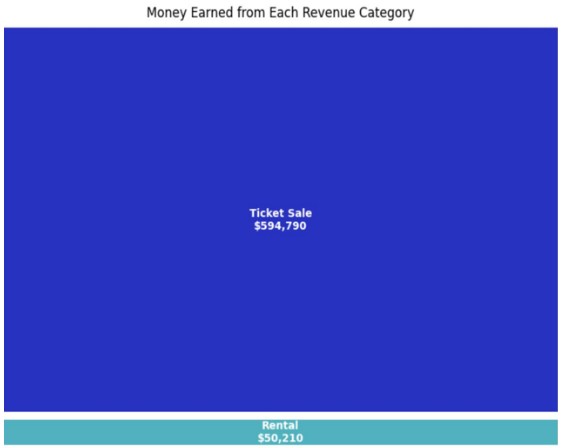
- Lodging: Revenue from ski-in/ski-out accommodations, hotels, and cabins.
- Food & Beverage: Restaurants, cafés, après-ski lounges, and mountain-view dining experiences.
- Retail & Rentals: Sales of branded merchandise, outdoor gear, and equipment rentals.



Financial Overview







Total deficit = - $ 2,468,469

- ❖ Revenue vs. Expenses Imbalance: Total revenue ($645K) is far lower than staff salaries alone ($2.5M), indicating a need for cost-cutting and revenue growth.

- ❖ **Staff Salary Dominates:** The largest share of expenses, over 75% of all expenses, is allocated to staff salaries.
- ❖ **Equipment Maintenance and Emergency services cost:** Ambulance calls (11%) and equipment maintenance (7%) are smaller but significant costs which can be minimised to increase margins.

- ❖ **One-Dimensional Revenue Stream:** Despite offering multiple services like lodging, rentals and restaurants 92% of revenue is driven by ticket sales alone, most of it coming in the winter season. Pointing to need for growing other businesses.

Challenges Facing Peak Mountain Resort

Aging Infrastructure & High Maintenance Costs

The resort spends approximately $240,000 in equipment repairs and maintenance costs with its share being 7.7% of all expenses.

Financial Constraints Impacting Operations

The resort is operating at a deficit of $2.4 million, which prevents it from upgrading infrastructure as well as invest appropriately in updating safety protocols.

Rising Safety Liabilities

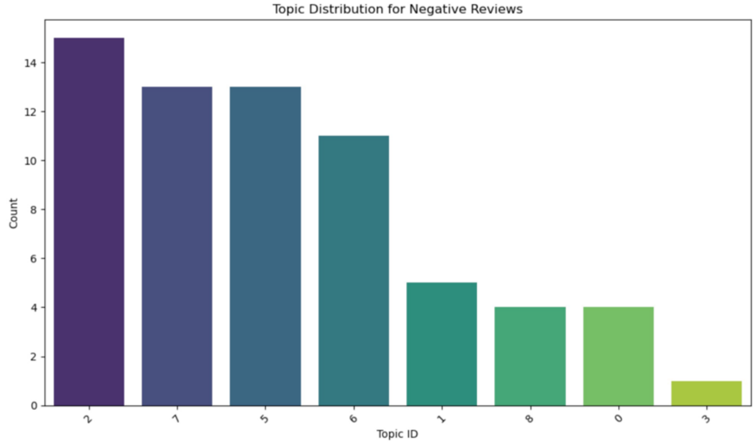
The increasing frequency of severe injuries, coupled with the resort's remote location, increasing resort's exposure to significant liability claims which has already led to $370,000 on ambulance calls.

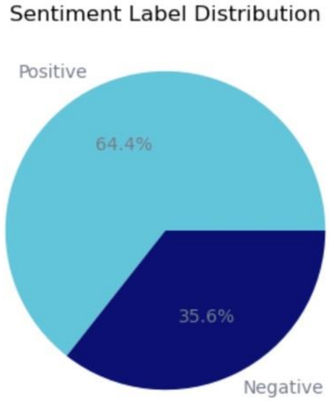
Slow Emergency Response

The resort's current average ambulance response time which is 15 minutes is extremely slow, exacerbated by the resort's remote location which directly jeopardizes guest safety.

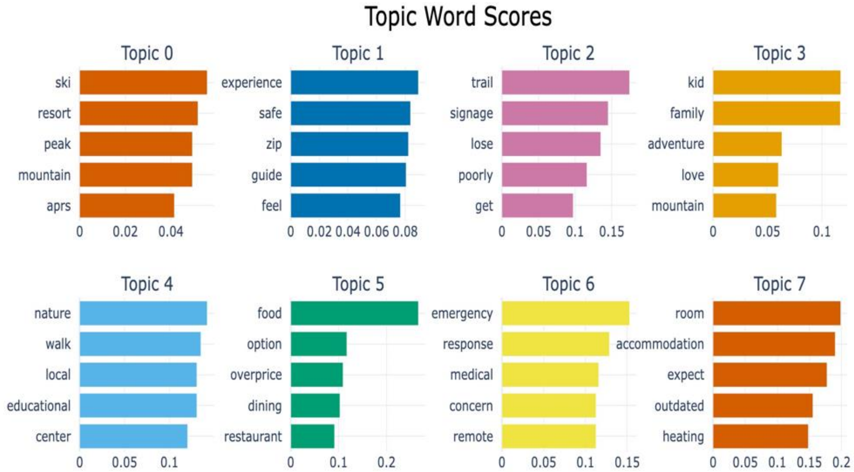
NLP Model:

Sentiment Analysis & Topic Modelling





- ❖ Cleaned the text and applied lemmatization after removing stop words and punctuation using SpaCy. Also, removed duplicates.
- ❖ Utilized the sentiment-analyzer of DistilBERT model for binary sentiment classification through confidence scores.
- ❖ Extract key topics from the cleaned reviews using BERTopic and identified topics associated with negative reviews.



Top 5 Customer Complaints relating to:

- ❖ Poor Trail signage
- ❖ Expensive Food
- ❖ Substandard Lodging
- ❖ Slow Emergency Response
- ❖ Aged Equipment

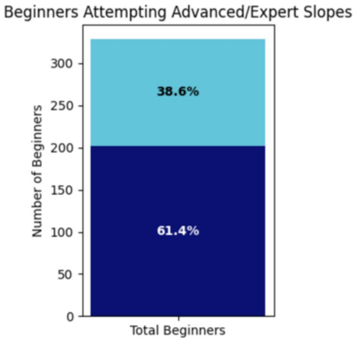
Data-Driven Improvement: Leverage customer insights to prioritize spending for better satisfaction

Advanced Analytics Used: NLP and sentiment analysis utilizing BERT will inform strategic decision-making

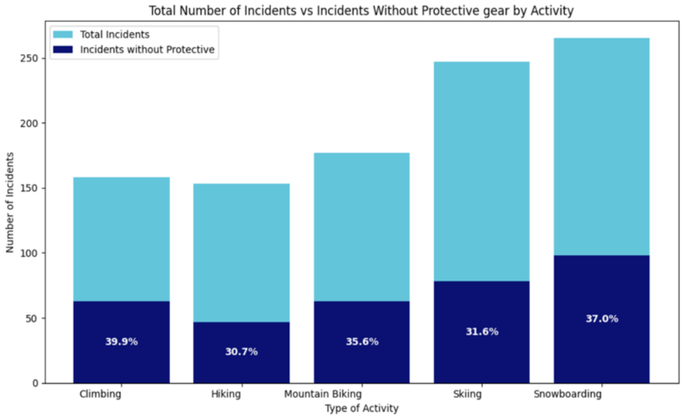
Data-Driven Safety Analysis

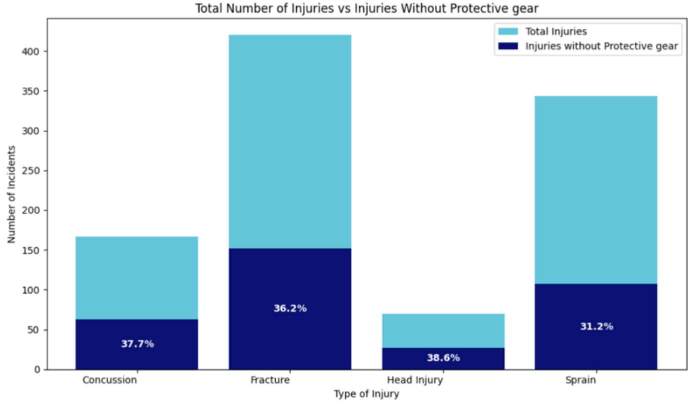
Injury and Severity Analysis

- Skiing and snowboarding had the highest number of incidents, approximately 250 each.
- Devil's Drop and Shadow Valley experienced the most severe cases of injury.
- The highest number of incidents without gear occurred in skiing and snowboarding, while climbing and snowboarding had the highest percentage rates.



High frequency (61.4%) of beginners attempting advanced/expert level slopes





Recommendation

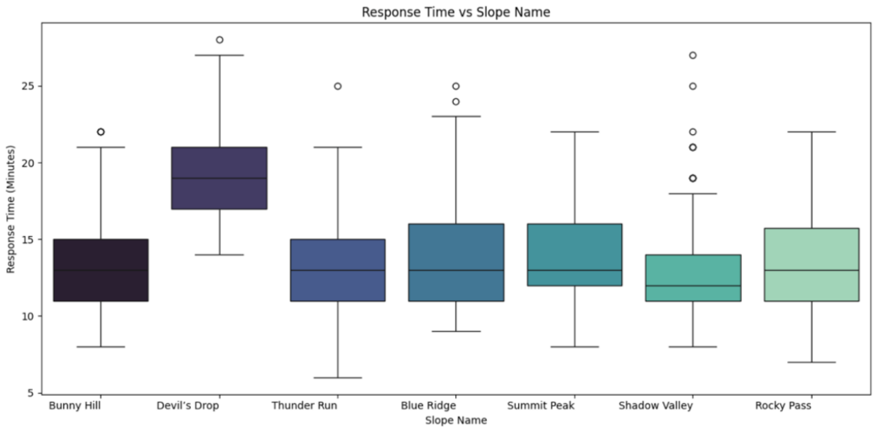
- Protective gear should be made compulsory
- Beginners should not be allowed to allowed/expert slopes.

Business Model Recommendation

Improving Emergency Response Times

Relocate First-Aid Stations

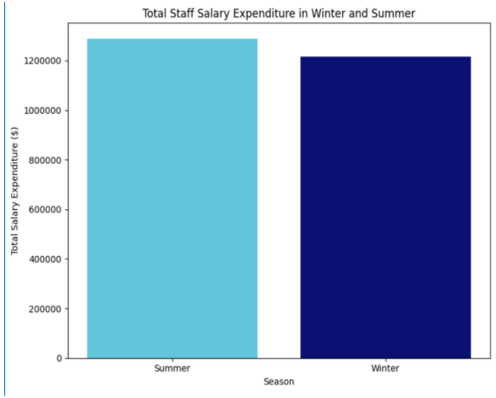
- Move from high-traffic areas to high-risk locations (Devil's Drop & Shadow Run)
- Introduce a strategic staging area leveraging partnership with medical centre
- Reduce response time from an average of 15 minutes to 7 minutes



Strategic Allocation of Resources

Align summer salary spending with its 20% revenue share.

- Discontinue all activities except Para-gliding and Via Ferrata to eliminate 75% of salary costs during summer.
- Reallocate savings to purchase and operate 2 on-site ambulances.



Business Model Recommendation

Aging Infrastructure and High Maintenance Costs

Generate Additional Revenue

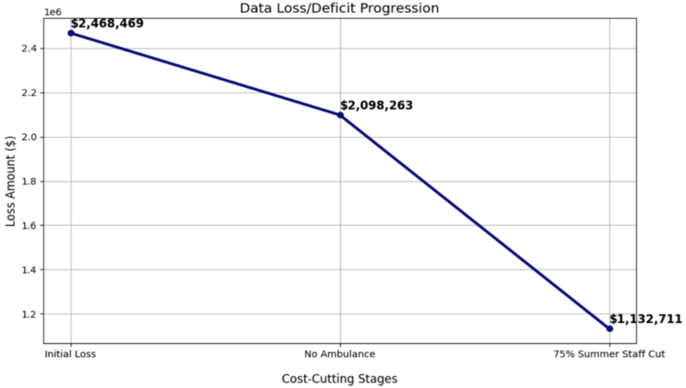
Generate revenue through mandatory protective gear.

Estimated revenue = $ 17,500

1.
- **Energy-Efficient Cost Reductions**: Solar Panels for Lodge Heating, electricity & heating costs
- **Summer Lodge Rental**: Convert underused resort spaces into event venues for corporate retreats, weddings to generate off-season revenue.

2.
- **Phased Infrastructure Upgrades** Instead of Immediate Replacements
- **Prioritize Critical Fixes**: Focus spending on essential infrastructure (e.g., ski lifts & safety barriers) while delaying non-essential upgrades (e.g., aesthetic lodge renovations)

Long Term Resilience



- ❖ First-year changes: Sign exclusion of liability waivers from guests to cut down on costs associated with ambulance calls.
- ❖ Reduce the summer staff by 75% in-order to cut down on the disproportionate expenses during summer.

Five-year path to breaking even: Increase the ticket prices year on year by 5% and adjust according to rising inflation and operating costs.

Conclusion: Recommendations in Effect

Discontinue all activities except Para-gliding and Via Ferrata to eliminate 75% of salary costs during summer.

Reallocate savings to purchase and operate 2 on-site ambulances.

Sign exclusion of liability waivers from guests to cut down on costs associated with ambulance calls.

Beginners should not be allowed to allowed/expert slopes.

Protective gear should be made compulsory

Move first aid from high-traffic areas to high-risk locations (Devil's Drop & Shadow Run)

Drawbacks

Long Break-Even Period: The resort is still 5 years away from profitability, which may not satisfy stakeholders seeking quicker returns.

Reduced Service Offerings – Cutting hiking and trail running, mountain biking, climbing, bouldering, and whitewater rafting & kayaking may impact guest experience and reduce **summer revenue potential**.

Appendix - 1 (Methodologies for NLP)

Cleaning Review Data and making Sentiment Analysis and plotting its graphs

```python
# Load SpaCy model
nlp = spacy.load("en_core_web_sm")

# Read the CSV file
df = pd.read_csv("./data/reviews.txt", sep="\t", names=['review'])

def clean_review(text):
    if not isinstance(text, str): #handle potential NaN values
        return ""

    # Remove numbers at the beginning
    text = re.sub(r"^\d+\.\s", "", text)

    # SpaCy processing
    doc = nlp(text.lower())
    lemmas_no_stop = [token.lemma_ for token in doc if not token.is_punct and not token.is_space and not token.is_stop]
    cleaned_text = " ".join(lemmas_no_stop)

    return cleaned_text

# Apply cleaning and sentiment analysis
df['cleaned_review'] = df['review'].apply(clean_review)

df.drop_duplicates(subset='cleaned_review', keep='first', inplace=True)
df.reset_index(drop=True, inplace=True)

sentiment_pipeline = pipeline("sentiment-analysis", model="distilbert-base-uncased-finetuned-sst-2-english")

def get_bert_sentiment(text):
    """Classifies sentiment using BERT."""
    if text.strip() == "":
        return "NEUTRAL", 0.5  # Assume neutral for empty text

    result = sentiment_pipeline(text)[0]  # Returns label & score
    return result['label'], result['score']

# Apply BERT sentiment analysis
df[['bert_label', 'bert_score']] = df['cleaned_review'].apply(lambda x: pd.Series(get_bert_sentiment(x)))

# Convert Hugging Face labels to match previous format
df['sentiment_label'] = df['bert_label'].map(lambda x: "Positive" if x == "POSITIVE" else "Negative" if x == "NEGATIVE" else "Neutral")

# Count distribution
sentiment_counts = df['sentiment_label'].value_counts()
```

```python
# Plot Pie Chart
plt.figure(figsize=(4, 4))
colors = {'Positive': '#63c5da', 'Neutral': 'gray', 'Negative': '#0a1172'}
df['sentiment_label'].value_counts().plot.pie(autopct='%1.1f%%', colors=[colors[label] for label in sentiment_counts.index],textprops={'color': '#757c88'})
plt.ylabel("")
plt.title("Sentiment Label Distribution")
plt.show()
```

- ❖ Read customer reviews from a text file (CSV with tab separation)
- ❖ Clean the text by removing numbers at the beginning, lowercasing, and applying lemmatization while removing stopwords and punctuation using SpaCy and removed duplicate reviews.
- ❖ Assign a confidence score and map results to a custom format.
- ❖ Visualize sentiment distribution using bar and pie charts.

Appendix – 2

Utilize BERTopic to do topic modelling

```
from bertopic import BERTopic
cleaned_reviews_list = df['cleaned_review'].tolist()
topic_model = BERTopic(verbose=True)
topics, probs = topic_model.fit_transform(cleaned_reviews_list)


# Get topic information
topic_info = topic_model.get_topic_info()
print(topic_info)


# Visualize barchart of topic distribution
topic_model.visualize_barchart()
```

```
negative_review_topics = df['topic'][df['sentiment_label']=="Negative"]
print(negative_review_topics)
negative_review_topics_counts = negative_review_topics[negative_review_topics != -1].value_counts()
negative_review_topics_counts = negative_review_topics_counts.sort_values(ascending=False)
plt.figure(figsize=(10, 6))  # Adjust the figure size as needed
sns.barplot(x=negative_review_topics_counts.index.astype(str), y=negative_review_topics_counts.values, palette='viridis')
plt.xlabel("Topic ID")
plt.ylabel("Count")
plt.title("Topic Distribution for Negative Reviews")
plt.xticks(rotation=45)  # Rotate x-axis labels for readability if needed
plt.tight_layout() #prevents labels from being cut off.
plt.show()
```

- ❖ Extract key topics from the cleaned reviews using BERTopic
- ❖ Analyze and visualize the distribution of topics across all reviews
- ❖ Identify and analyze topics associated with negative reviews to gain insights into major concerns.
- ❖ Plot bar chart for negative review topic distribution.

Appendix – 3

Cleaning Dataset

- During data cleaning, we identified cases in the incidents dataset where the same incident was associated with 5-6 ambulance calls across different seasons. This was identified as a data error. To correct this, we retained only the ambulance calls that matched the season of the incident, ensuring the expenditure on ambulances was neither overestimated nor underestimated.
- Then, for each incident, we combined the ambulance expenses into a single entry by summing the amounts for the same Incident ID. We retained only the first Transaction ID for each incident, effectively removing duplicate entries associated with the same incident.

```
ambulance_calls = expense[expense['Category'] == 'Ambulance Call']
print(ambulance_calls.shape)
non_ambulance_data = expense[expense['Category'] != 'Ambulance Call']
print(non_ambulance_data.shape)

ambulance_with_incident = ambulance_calls.merge(
    incident[['Incident_ID', 'Season']],
    on='Incident_ID',
    how='left',
    suffixes=('', '_incident')
)

matching_ambulance_calls = ambulance_with_incident[
    ambulance_with_incident['Season'] == ambulance_with_incident['Season_incident']
].drop(columns=['Season_incident'])

matching_ambulance_calls.head()
```

```
combined_ambulance_calls = (
    matching_ambulance_calls
    .sort_values(by=['Incident_ID', 'Transaction_ID'])
    .groupby('Incident_ID', as_index=False)
    .agg({
        'Transaction_ID': 'first',
        'Transaction_Type': 'first',
        'Category': 'first',
        'Amount': 'sum',
        'Date': 'first',
        'Season': 'first',
        'Weather': 'first',
        'Customer_ID': 'first'
    })
)

print(combined_ambulance_calls.shape)
combined_ambulance_calls.head()
```

```
final_expense = pd.concat([non_ambulance_data, combined_ambulance_calls], ignore_index=True)
final_expense.head()
```