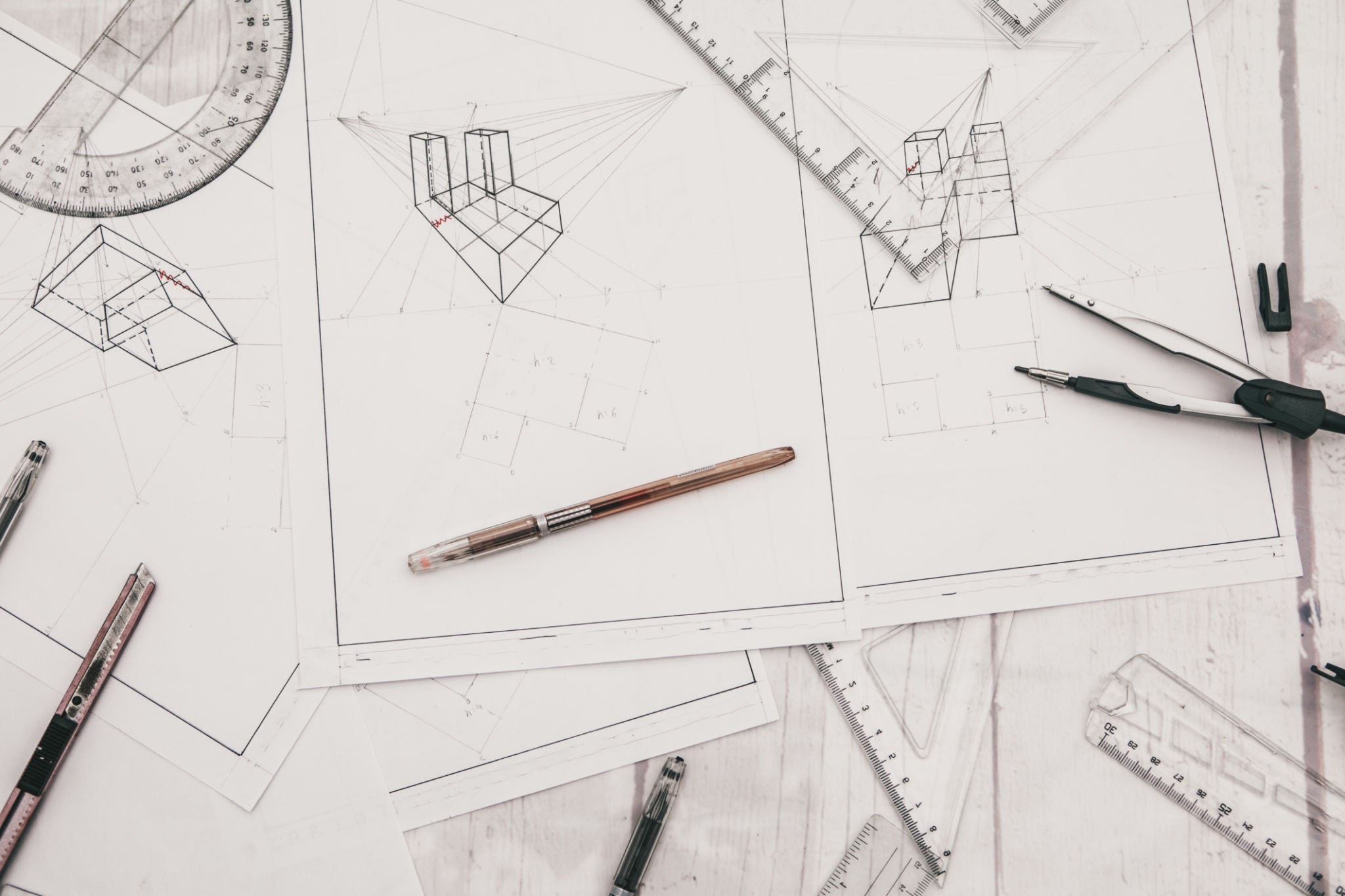
Overview



import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.metrics import classification\_report, roc\_curve, auc

import matplotlib.pyplot as plt

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Conv1D, Flatten, Dropout, MaxPooling1D

from tensorflow.keras.utils import to\_categorical

# Step 1: Load and preprocess the dataset

df = pd.read\_csv("LungCancerDataset.csv") # Add path if needed

# Encode categorical variables

# data=data.replace({'LUNG\_CANCER': {'YES': 1, 'NO': 0}})

df['LUNG\_CANCER'] = df['LUNG\_CANCER'].map({'YES': 1, 'NO': 0})

df['GENDER'] = df['GENDER'].map({'M': 0, 'F': 1})

# Split data into features and target

X = df.drop('LUNG\_CANCER', axis=1)

y = df['LUNG\_CANCER']

# Normalize features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Reshape data for CNN (add channel dimension)

X\_reshaped = X\_scaled.reshape(X\_scaled.shape[0], X\_scaled.shape[1], 1)

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_reshaped, y, test\_size=0.2, random\_state=42)

# Step 2: Build the CNN model

model = Sequential([

Conv1D(32, kernel\_size=3, activation='relu', input\_shape=(X\_train.shape[1], 1)),

MaxPooling1D(pool\_size=2),

Dropout(0.3),

Conv1D(64, kernel\_size=3, activation='relu'),

MaxPooling1D(pool\_size=2),

Flatten(),

Dense(64, activation='relu'),

Dropout(0.3),

Dense(1, activation='sigmoid')

])

# Compile the model

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

# Step 3: Train the model

history = model.fit(X\_train, y\_train, epochs=12, batch\_size=16, validation\_split=0.2, verbose=1)

# Step 4: Evaluate the model

y\_pred\_proba = model.predict(X\_test)

y\_pred = (y\_pred\_proba > 0.5).astype(int)

# Classification metrics

print(classification\_report(y\_test, y\_pred, target\_names=['No Cancer', 'Cancer']))

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay

# Compute the confusion matrix

cm = confusion\_matrix(y\_test, y\_pred)

# Display the confusion matrix

disp = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=['No Cancer', 'Cancer'])

disp.plot(cmap='Blues')

plt.title('Confusion Matrix')

plt.show()

# Step 5: Accuracy and Loss Graph

plt.figure(figsize=(12, 5))

# Accuracy Graph

plt.figure(figsize=(8,6))

plt.plot(history.history['accuracy'], label='Train Accuracy')

plt.plot(history.history['val\_accuracy'], label='Validation Accuracy')

plt.title('Model Accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

# Loss Graph

#plt.subplot(1, 2, 2)

plt.figure(figsize=(8,6))

plt.plot(history.history['loss'], label='Train Loss')

plt.plot(history.history['val\_loss'], label='Validation Loss')

plt.title('Model Loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.tight\_layout()

plt.show()

# calculate metrics

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, roc\_curve, auc

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

# metrics plotting

metrics = {

'Accuracy': accuracy,

'Precision': precision,

'Recall': recall,

'F1 Score': f1

}

# print metrics

for metric, value in metrics.items():

print(f'{metric}: {value}')

# create dataframe

df\_metrics = pd.DataFrame(list(metrics.items()), columns=['Metric', 'Value'])

# plot metrics as bar chart

plt.figure(figsize=(6, 4))

plt.bar(df\_metrics['Metric'], df\_metrics['Value'])

plt.xlabel('Metrics')

plt.ylabel('Value')

plt.title('Performance Metrics')

plt.ylim(0.25, 1)

plt.tight\_layout()

plt.show()

# Step 7: ROC Curve

fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_proba)

roc\_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, label=f'ROC Curve (AUC = {roc\_auc:.2f})', color='darkorange')

plt.plot([0, 1], [0, 1], color='navy', linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic (ROC) Curve')

plt.legend()

plt.grid()

plt.show()

from sklearn.metrics import precision\_recall\_curve, auc

# Compute precision, recall, and thresholds

precision, recall, thresholds = precision\_recall\_curve(y\_test, y\_pred\_proba)

# Compute the Area Under the Curve (AUC) for Precision-Recall

pr\_auc = auc(recall, precision)

# Plot Precision-Recall Curve

plt.figure(figsize=(8, 6))

plt.plot(recall, precision, label=f'PR Curve (AUC = {pr\_auc:.2f})', color='blue')

plt.xlabel('Recall')

plt.ylabel('Precision')

plt.title('Precision-Recall Curve')

plt.legend()

plt.grid()

plt.show()

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.metrics import confusion\_matrix

all\_classes = [0, 1]

# Compute confusion matrix

cm = confusion\_matrix(y\_test, y\_pred, labels=all\_classes)

# Initialize arrays to store FPR and FNR

fpr = []

fnr = []

# Calculate FPR and FNR for each class

for i in range(len(all\_classes)):

TP = cm[i, i] # True Positives

FP = cm[:, i].sum() - TP # False Positives

FN = cm[i, :].sum() - TP # False Negatives

TN = cm.sum() - (TP + FP + FN) # True Negatives

# Compute FPR and FNR

fpr\_i = FP / (FP + TN) if (FP + TN) > 0 else 0

fnr\_i = FN / (FN + TP) if (FN + TP) > 0 else 0

fpr.append(fpr\_i)

fnr.append(fnr\_i)

# Print FPR and FNR for each class

for idx, class\_label in enumerate(all\_classes):

print(f"Class {class\_label}:")

print(f" FPR (False Positive Rate): {fpr[idx]:.4f}")

print(f" FNR (False Negative Rate): {fnr[idx]:.4f}")

# Save FPR and FNR to an Excel file

df = pd.DataFrame({

'Class': all\_classes,

'FPR (False Positive Rate)': fpr,

'FNR (False Negative Rate)': fnr

})

#df.to\_excel('result/FPV\_FNR.xlsx', index=False)

# Plotting FPR and FNR as a bar chart for each class

x\_labels = [f"Class {class\_label}" for class\_label in all\_classes]

x = np.arange(len(all\_classes)) # Position for bars

# Bar width

bar\_width = 0.35

plt.figure(figsize=(8, 6))

plt.bar(x - bar\_width / 2, fpr, bar\_width, label='FPR (False Positive Rate)', color='salmon')

plt.bar(x + bar\_width / 2, fnr, bar\_width, label='FNR (False Negative Rate)', color='grey')

# Chart formatting

plt.title('FPR and FNR for Each Class', weight='bold', fontsize=14)

plt.xlabel('Class', weight='bold', fontsize=12)

plt.ylabel('Rate', weight='bold', fontsize=12)

plt.xticks(x, x\_labels)

plt.legend()

plt.tight\_layout()

plt.show()

Background

# BACKGROUND

## History with the company

In a paragraph or less, provide an overview of the history with this company. This could include interactions, key events, communications, and milestones.

## Key decision-makers

| Name | Title | POC | Contact information | Notes |
| --- | --- | --- | --- | --- |
| Person | Title | Person | Add information | Add notes |
| Person | Title | Person | Add information | Add notes |
| Person | Title | Person | Add information | Add notes |

## Current product and services

* List the types of products and services this customer prefers
* Preference 2
* Preference 3

Account strategy

# ACCOUNT STRATEGY

Indicate how the account strategy plan aligns with the overall goals of the account. Explain how the strategy will be used to pursue specific initiatives to help achieve company goals.

| ACTIONS | RESOURCES | METRICS | DUE DATE | OWNER |
| --- | --- | --- | --- | --- |
| Add action | Add resources | * Metric 1 * Metric 2 | Date | Person  Person |
| Add action | Add resources | * Metric 1 * Metric 2 | Date | Person  Person |
| Add action | Add resources | * Metric 1 * Metric 2 | Date | Person  Person |
| Add action | Add resources | * Metric 1 * Metric 2 | Date | Person  Person |
| Add action | Add resources | * Metric 1 * Metric 2 | Date | Person  Person |

Competitive and SWOT analysis

# COMPETITIVE ANALYSIS

## Describe their competitors

* List companies offering similar products or services. Be specific and include large and small competitors.
* Consider alternative products or services that might meet the same customer needs

## Summarize their competitive advantage

* Clearly articulate how your client differentiates itself from the competition
* Highlight market opportunities
* Support the competitive advantages with data whenever possible

| INNOVATIVE | | | |
| --- | --- | --- | --- |
| HIGHER PRICE | Competitor | Company Competitor Competitor | LOWER PRICE |
| CompetitorCompetitor | Competitor |
| CONVENTIONAL | | | |

# SWOT ANALYSIS

| Strengths | Weaknesses |
| --- | --- |
| Internal factors that give the account a competitive advantage.  Focus on unique capabilities, resources, or market position.  Examples: Strong brand reputation, innovative products, loyal customer base, efficient operations, skilled workforce. | Internal factors that can hinder the account's progress.  Identify areas for improvement or potential vulnerabilities.  Examples: Limited market share, outdated technology, high costs, lack of skilled personnel, weak distribution network. |
|  |  |
| Opportunities | Threats |
| External factors that the account can use for growth or improvement.  Identify emerging trends, market gaps, or favorable conditions.  Examples: New market segments, technological advancements, changing regulations, economic growth, competitor weaknesses. | External factors that can negatively impact the account.  Identify potential risks and challenges.  Examples: Intense competition, economic downturn, changing customer preferences, disruptive technologies, regulatory changes. |

Interaction log

# INTERACTION LOG

| Date | POC | Type of interaction | Notes | Next steps |
| --- | --- | --- | --- | --- |
| Date | Person | Meeting | *Interested in new product* | Ongoing |
| Date | Person | Phone call | Notes | Resolved |
| Date | Person | Email | Notes | Escalated |

# ACTION ITEMS

| Action item | Assigned to | Deadline | Status |
| --- | --- | --- | --- |
| *Send customer new catalog* | Person | Date | Completed |
| Action item 2 | Person | Date | Completed |
| Action item 3 | Person | Date | Completed |

# MEETING NOTES

*To add a meeting notes building block, type “@meeting notes”. Search for or select a meeting to add attendees, notes, and action items.*