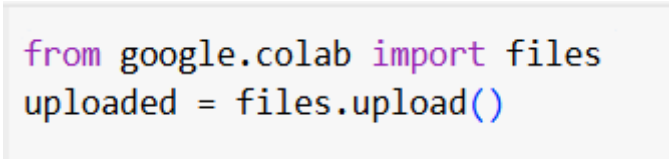
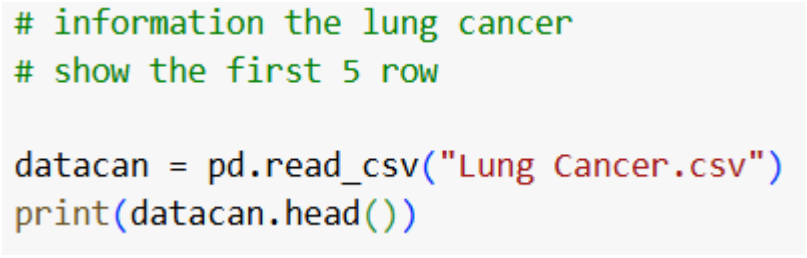
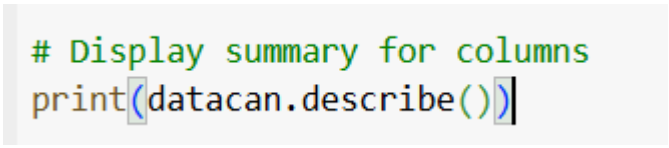
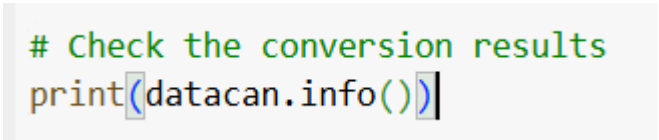
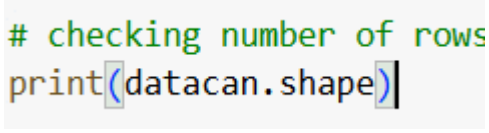
	احنا هنا عملنا تصدير للمكتبات الي بنستخدمه في الكود
	هنا بنحمل الداتا من علي الجهاز علي السيرفر
	هنا بنقر الملف ونعمل اظهار لاول خمس صفوف للملف
	عملنا دا عشان نعرف ملخص لداتا الموجوده في الملف
	عشان نعرف الصفوف الي موجود ونوغ البيانات الموجود واذا فيه حاجه فاضيه ولا لا
	لمعرفه عدد الصفوف ولا الاعمده

``# finding number of uniqu print(datacan.nunique())```	عشان نعرف عدد الاشياء المميزه في الصف
``# Plot histograms for all columns datacan.hist(figsize=(20, 14), bins= 15) plt.show()```	عشان نعمل اشكال بيانيه للكل لصف

``# change str to int number #to make easily more to make compersion between them #change female = 0 , male = 1 datacan['gender'] = datacan['gender'].map({'Male': 1, 'Female': 0}) #change cancer_stage in stage to number datacan['cancer_stage'] = datacan['cancer_stage'].map({'Stage I' : 1, 'Stage II' : 2, 'Stage III' : 3 , 'Stage IV' : 4, 'Stage V' : 5 , 'Stage VI' : 6}) #change smoking_status in stage to number datacan['smoking_status'] = datacan['smoking_status'].map({"Never Smoked": 0, "Passive Smoker": 1,"Former Smoker": 2,"Current Smoker": 3}) #change family_history in stage to number datacan['family_history'] = datacan['family_history'].map({ 'Yes' : 1 , 'No' : 0 }) ##change family_history in stage to number datacan['treatment_type'] = datacan['treatment_type'].map({"Chemotherapy": 0, "Surgery": 1, "Radiation": 2,"Combined": 3 })```	هنا حولنا اي حاجه نص في العواميد دا عشان نعرف نعدل او نعمل حساب في الكود عشان تبقي اسهل في العمليات وميعملش خط
``# Checking for null values (if any) print(datacan.isna().any()) print(datacan.isnull().sum())```	بنشوف ان فيه حاجه فاضيه ولا لا ولو فيه يظهر عددهم
``#show boxplot for col in datacan: plt.figure(figsize=(6, 2)) sns.boxplot(data=datacan, x=col) plt.title(f"Outlier Check: {col}") plt.tight_layout() plt.show()```	احنا استخدامنا الشكل دا عشان يظهر الاشكال عشان برضوا نعرف الاوت ليرس

`` # Country distribution pie chart country_counts = datacan['country'].value_counts().head(10) plt.figure(figsize=(12, 8)) plt.pie(country_counts.values, labels=country_counts.index, autopct='%1.1f%%' plt.title('Top 10 Countries by Lung Cancer Patient Distribution') plt.axis('equal') plt.show() print("Top 10 countries:") print(country_counts) ```	هنا عملنا شكل بياني للعمود الدول لمعرفه اكتر 10 دول مصابه
`` # delete the information str and i dont need that #impact the acurcue the programe for xy in ['end_treatment_date', 'diagnosis_date', 'country','id']: if xy in datacan.columns: datacan.drop(columns=xy, inplace=True) ```	هنا حذفنا الاشياء التي توثر ولا احتاجها في العمليات
``] # Family history vs cancer stage heatmap family_stage_crosstab = pd.crosstab(datacan['family_history'], datacan['cancer_stage'], plt.figure(figsize=(10, 6)) sns.heatmap(family_stage_crosstab, annot=True, cmap='YlOrRd', fmt='.1f', cbar_kws={'label': 'Percentage'}) plt.title('Cancer Stage Distribution by Family History') plt.xlabel('Cancer Stage') plt.ylabel('Family History') plt.show() ```	عملنا شكل بياني عشان نعرف اذا في لو فيه حد عندوا كان حد عندوا تاريخ او لا نعرف مستوي درجه الاصابه
`` #check target variable balance sns.countplot(x='survived',data=datacan) plt.title('Survival Count') plt.show() ```	عشان نعرف عدد الناس الناجيين
`` # correlation matrix correlation_matrix = datacan.corr() plt.figure(figsize=(12, 8)) sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', square=True plt.title('Correlation Matrix') plt.show() ```	عشان نعرف العلاقه بين العواميد والصفوف

`` columns_to_exclude = ['survived'] for column in datacan.select_dtypes(include=['float64', 'int64']).columns: if column not in columns_to_exclude: Q1 = datacan[column].quantile(0.25) Q3 = datacan[column].quantile(0.75) IQR = Q3 - Q1 lower_bound = Q1 - 1.5 * IQR upper_bound = Q3 + 1.5 * IQR datacan = datacan[(datacan[column] >= lower_bound) & (datacan[column] <= upper_b ```	عملنا الكود دا عشان نحذف الاوت لير التي توثر علي عمليات التدريب
`` print(datacan['gender'].value_counts()) print(datacan.groupby('gender')['survived'].value_counts()) # Survival rate by gender print(datacan.groupby('gender')['survived'].mean()) ```	عشان نعرف العلاقه بين بين النوع هياثر علي احتماليه النجاه
`` #Split and scale data X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) scaler = StandardScaler() X_train_scaled = scaler.fit_transform(X_train) X_test_scaled = scaler.transform(X_test) ```	عملنا دا عشان نفصل الداتا شويه للتدريب وشويه عشان نعمل بيهم للتحقق من دقه النموذج
`` #Train and evaluate base model logistic_model = LogisticRegression() logistic_model.fit(X_train_scaled, y_train) y_pred = logistic_model.predict(X_test_scaled) # Accuracy print(f"Logistic Regression Accuracy: {accuracy_score(y_test, y_pred):.2%}") ```	عشان ندرب علي Logistic Regression ونشوف دقه التنبو بتاع النموذج
`` # Confusion Matrix conf_matrix = confusion_matrix(y_test, y_pred) conf_matrix_df = pd.DataFrame(conf_matrix, index=['Actual Negative', 'Actual Positive'], columns=['Predicted Negative', 'Predicted Positive']) print("Confusion Matrix:\n", conf_matrix_df) # Classification Report print("Classification Report:\n", classification_report(y_test, y_pred)) ```	عشان نعرف precision, recall, f1-score, support, Predicted Negative, Predicted Positive, Actual Negative, Actual Positive

`` #train model models = { 'Logistic Regression': LogisticRegression(), 'SVM': SVC(class_weight='balanced'), 'Decision Tree': DecisionTreeClassifier(), 'Random Forest': RandomForestClassifier(), 'KNN': KNeighborsClassifier() } results = [] f1_scores = {} ```	هنا عملنا الكود عشان نعمل تدريب لداتا ونحفظها
`` for name, model in models.items(): model.fit(X_train_scaled, y_train) y_pred = model.predict(X_test_scaled) acc = accuracy_score(y_test, y_pred) prec = precision_score(y_test, y_pred) rec = recall_score(y_test, y_pred) f1 = f1_score(y_test, y_pred) results.append({ 'Model': name, 'Accuracy': acc, 'Precision': prec, 'Recall': rec, 'F1 Score': f1 }) f1_scores[name] = f1 ```	عشان نعمل تدريب لنماذج كلها مع بعض عشان نعرف افضل طريقه نستخدمها نمشي بيها
`` # Visualize performance results_df = pd.DataFrame(results) print("\nModel Comparison Results:") print(results_df) plt.figure(figsize=(10, 6)) for metric in ['Accuracy', 'Precision', 'Recall', 'F1 Score']: plt.plot(results_df['Model'], results_df[metric], marker='o', label=metric) plt.title('Model Performance Comparison') plt.ylabel('Score') plt.ylim(0, 1.1) plt.legend() plt.grid(True) plt.show() ```	عملنا الكود عشان نظهر الفرق بين طرق التدريب علي شكل رسم بياني

`` # F1 Bar Chart plt.figure(figsize=(8, 5)) plt.bar(f1_scores.keys(), f1_scores.values(), color='skyblue') plt.title("F1-score Comparison between Models") plt.xlabel("Model") plt.ylabel("F1-score") plt.ylim(0, 1) plt.grid(axis='y') for i, (model, score) in enumerate(f1_scores.items()): plt.text(i, score + 0.01, f"{score:.2f}", ha='center', fontweight='bold') plt.tight_layout() plt.show() ```	عملنا رسم لمقارنة نماذج بناءً على F1-score
`` # Cross-Validation print("\nCross-Validation Scores:") for name, model in models.items(): scores = cross_val_score(model, X, y, cv=5) print(f"{name}: Mean CV Accuracy = {scores.mean():.2f}") ```	هنا بيعمل التدريب اكثر من مره وبعدين المتوسط عشان نعرف اقرب حاجه للدقه النموذج للحقيقه
`` rus = RandomUnderSampler(random_state=42) X_resampled, y_resampled = rus.fit_resample(X_train, y_train) scaler = StandardScaler() X_resampled_scaled = scaler.fit_transform(X_resampled) X_test_scaled = scaler.transform(X_test) ```	عشان نحل مشكله الاصفار التي طلعت في التدريب في حاله الناجين مطلعهم صفر
`` cm = confusion_matrix(y_test, y_pred) sns.heatmap(cm, annot=True, fmt='d', cmap='Blues') plt.title(f'Confusion Matrix ') plt.xlabel('Predicted') plt.ylabel('Actual') plt.show() # for name, model in models.items(): with open(f'{name.replace(" ", "_").lower()}_model.pkl', 'wb') as f: pickle.dump(model, f) ```	استخدمنا الكود توضيح أداء النموذج في التنبؤ بالفئات الفعلية ويقوم بخفظها

``# Save trained models for name, model in models.items(): with open(f'{name.replace(" ", "_").lower()}_model.pkl', 'wb') as f: pickle.dump(model, f)```	يقوم عمليات التدريب التي تم تنفيذها في الكود
``# save the new values after that remove the text #datacan.to_csv('Lung Cancer.csv')```	يقوم بحفظ اي تعديل في الملف