

## A3.1 SVM y Multiple Testing

```
In [2]: import pandas as pd
from scipy.stats import ttest_ind
from statsmodels.stats.multitest import multipletests
import numpy as np
from scipy.stats import f_oneway
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report
```

1. Se carga el conjunto de datos con todos los genes para poder trabajar con ellos. Después se revisa que no haya huecos en los huecos en cada columna, y se imprime la confirmación en pantalla como un valor booleano.

```
In [3]: # Cargar el archivo CSV
df = pd.read_csv('A3.1 Khan.csv')
# Mostrar dimensiones y valores faltantes. La función .isnull() regresa el mismo DataFrame
# .any() returns True if at least one value in the specified axis is True. applied
# To grab a particular column, one has to
# .any().any()

print("¿Hay huecos?: ", df.isnull().any().any())
```

¿Hay huecos?: False

A través de la lógica que los métodos de DataFrame posee, podemos proceder seguramente con las instrucciones siguientes al saber que no hay huecos.

Se calculan y ordenan las diferencias absolutas correctamente, y se muestran los 10 genes principales.

```
In [4]: # Suponemos que La última columna es la variable de salida con valores del 1 al 4
# Asegúrate de renombrarla si no está claramente nombrada
target_col = df.columns[-1]
print("\nClases únicas en la columna objetivo:", df[target_col].unique())

# Separar clases 2 y 4
class_2 = df[df[target_col] == 2].drop(columns=target_col)
class_4 = df[df[target_col] == 4].drop(columns=target_col)

# Calcular diferencia de medias
diff_means = (class_2.mean() - class_4.mean()).abs().sort_values(ascending=False)

# Mostrar Los 10 genes con mayor diferencia
print("\nTop 10 genes con mayor diferencia de medias entre clase 2 y 4:")
print(diff_means.head(10))
```

Clases únicas en la columna objetivo: [2 4 3 1]

Top 10 genes con mayor diferencia de medias entre clase 2 y 4:

X187	3.323151
X509	2.906537
X2046	2.424515
X2050	2.401783
X129	2.165185
X1645	2.065460
X1319	2.045941
X1955	2.037340
X1003	2.011337
X246	1.837830

dtype: float64

Los genes listados (por ejemplo X187 , X509 , X2046 ) presentan mayores diferencias de expresión, lo cual sugiere que podrían ser biomarcadores entre los tipos de cáncer 2 y 4.

2. Cálculo del estadístico t y corrección por múltiples pruebas. Se realiza t-test (Welch) por gen, y se aplican Bonferroni, Holm y Benjamini-Hochberg (FDR) con alpha=0.05.

```
In [7]: # Calcular t-test para cada gen
t_stats = []
p_values = []

for gene in class_2.columns:
    t, p = ttest_ind(class_2[gene], class_4[gene], equal_var=False)
    t_stats.append(t)
    p_values.append(p)

# Correcciones por múltiples pruebas
methods = ['bonferroni', 'holm', 'fdr_bh']
corrections = {}

for method in methods:
    reject, p_corr, _, _ = multipletests(p_values, alpha=0.05, method=method)
    corrections[method] = {
        'genes_significativos': df.columns[:-1][reject],
        'pvalores_corregidos': p_corr[reject]
    }

# Mostrar resultados
for method, res in corrections.items():
    print(f"\nMétodo: {method}")
    print("Genes con diferencias significativas:", list(res['genes_significativos']))
    print(len(res['genes_significativos']))
```

Método: bonferroni

Genes con diferencias significativas: ['X2', 'X36', 'X67', 'X129', 'X174', 'X187', 'X188', 'X229', 'X246', 'X251', 'X338', 'X348', 'X368', 'X372', 'X373', 'X380', 'X430', 'X433', 'X509', 'X545', 'X554', 'X558', 'X566', 'X603', 'X655', 'X714', 'X762', 'X910', 'X951', 'X971', 'X1003', 'X1021', 'X1023', 'X1055', 'X1070', 'X1093', 'X1105', 'X1110', 'X1112', 'X1132', 'X1194', 'X1196', 'X1207', 'X1217', 'X1298', 'X1319', 'X1327', 'X1330', 'X1372', 'X1389', 'X1416', 'X1610', 'X1626', 'X1634', 'X1645', 'X1706', 'X1708', 'X1723', 'X1738', 'X1799', 'X1888', 'X1896', 'X1911', 'X1924', 'X1954', 'X1955', 'X1980', 'X2046', 'X2050', 'X2115', 'X2146', 'X2247']

72

Método: holm

Genes con diferencias significativas: ['X2', 'X36', 'X67', 'X129', 'X174', 'X187', 'X188', 'X229', 'X246', 'X251', 'X338', 'X348', 'X368', 'X372', 'X373', 'X380', 'X430', 'X433', 'X509', 'X545', 'X554', 'X558', 'X566', 'X603', 'X655', 'X714', 'X762', 'X910', 'X951', 'X971', 'X1003', 'X1021', 'X1023', 'X1055', 'X1070', 'X1093', 'X1105', 'X1110', 'X1112', 'X1132', 'X1194', 'X1196', 'X1207', 'X1217', 'X1298', 'X1319', 'X1327', 'X1330', 'X1372', 'X1389', 'X1416', 'X1610', 'X1626', 'X1634', 'X1645', 'X1706', 'X1708', 'X1723', 'X1738', 'X1799', 'X1888', 'X1896', 'X1911', 'X1924', 'X1954', 'X1955', 'X1980', 'X2046', 'X2050', 'X2115', 'X2146', 'X2247']

72

Método: fdr\_bh

Genes con diferencias significativas: ['X2', 'X3', 'X29', 'X36', 'X52', 'X67', 'X80', 'X89', 'X119', 'X129', 'X131', 'X139', 'X141', 'X146', 'X151', 'X153', 'X165', 'X166', 'X169', 'X174', 'X187', 'X188', 'X191', 'X214', 'X217', 'X229', 'X230', 'X244', 'X246', 'X251', 'X256', 'X258', 'X315', 'X323', 'X335', 'X336', 'X338', 'X340', 'X348', 'X365', 'X368', 'X372', 'X373', 'X378', 'X380', 'X384', 'X388', 'X390', 'X407', 'X409', 'X419', 'X426', 'X430', 'X433', 'X437', 'X443', 'X465', 'X469', 'X481', 'X482', 'X483', 'X500', 'X501', 'X503', 'X509', 'X518', 'X533', 'X534', 'X542', 'X545', 'X554', 'X558', 'X566', 'X598', 'X603', 'X604', 'X607', 'X636', 'X642', 'X650', 'X655', 'X696', 'X702', 'X714', 'X715', 'X733', 'X744', 'X746', 'X747', 'X753', 'X758', 'X760', 'X761', 'X762', 'X774', 'X779', 'X789', 'X795', 'X803', 'X808', 'X820', 'X828', 'X832', 'X851', 'X857', 'X867', 'X883', 'X891', 'X910', 'X941', 'X951', 'X964', 'X971', 'X973', 'X979', 'X982', 'X1003', 'X1012', 'X1020', 'X1021', 'X1022', 'X1023', 'X1030', 'X1037', 'X1039', 'X1046', 'X1055', 'X1063', 'X1070', 'X1074', 'X1089', 'X1090', 'X1093', 'X1100', 'X1105', 'X1110', 'X1112', 'X1113', 'X1125', 'X1132', 'X1151', 'X1155', 'X1158', 'X1160', 'X1164', 'X1187', 'X1194', 'X1196', 'X1201', 'X1203', 'X1204', 'X1206', 'X1207', 'X1217', 'X1220', 'X1225', 'X1227', 'X1228', 'X1238', 'X1252', 'X1254', 'X1258', 'X1263', 'X1283', 'X1286', 'X1292', 'X1298', 'X1301', 'X1319', 'X1325', 'X1327', 'X1330', 'X1331', 'X1345', 'X1353', 'X1372', 'X1389', 'X1416', 'X1464', 'X1489', 'X1493', 'X1496', 'X1497', 'X1498', 'X1517', 'X1518', 'X1525', 'X1536', 'X1574', 'X1597', 'X1606', 'X1610', 'X1613', 'X1626', 'X1634', 'X1643', 'X1644', 'X1645', 'X1646', 'X1655', 'X1670', 'X1671', 'X1673', 'X1677', 'X1691', 'X1697', 'X1706', 'X1708', 'X1714', 'X1723', 'X1727', 'X1729', 'X1734', 'X1738', 'X1771', 'X1772', 'X1775', 'X1778', 'X1799', 'X1821', 'X1822', 'X1831', 'X1850', 'X1853', 'X1854', 'X1855', 'X1857', 'X1867', 'X1870', 'X1878', 'X1882', 'X1888', 'X1896', 'X1901', 'X1906', 'X1909', 'X1910', 'X1911', 'X1914', 'X1917', 'X1920', 'X1924', 'X1929', 'X1931', 'X1937', 'X1942', 'X1945', 'X1954', 'X1955', 'X1967', 'X1979', 'X1980', 'X1991', 'X1994', 'X1995', 'X2000', 'X2020', 'X2039', 'X2046', 'X2047', 'X2049', 'X2050', 'X2060', 'X2080', 'X2081', 'X2083', 'X2086', 'X2088', 'X2093', 'X2096', 'X2099', 'X2105', 'X2114', 'X2115', 'X2117', 'X2120', 'X2146', 'X2148', 'X2149', 'X2159', 'X2172', 'X2181', 'X2199', 'X2227', 'X2230', 'X2235', 'X2247', 'X2248', 'X2253', 'X2262', 'X2275', 'X2278', 'X2295', 'X2300', 'X2301', 'X2303']

296

Método	# Genes significativos
bonferroni	72
holm	72
Benjamini-Hochberg	296

A través de esto podemos notar que a medida que el método es menos conservador, aumenta el número de genes significativos.

3. Se utilizará ANOVA multicas para las 4 clases y se corrige FDR con BH. El análisis de varianza identifica genes con expresión diferencial entre los cuatro tipos de cáncer.

```
In [9]: # Estratificar por clase
grouped = [df[df[target_col] == c].drop(columns=target_col) for c in sorted(df[target_col].unique())]

# Realizar ANOVA para cada gen
p_values_anova = []
for gene in df.columns[:-1]:
    f_stat, p = f_oneway(*[group[gene] for group in grouped])
    p_values_anova.append(p)

# Corrección Benjamini-Hochberg para ANOVA
reject_anova, pvals_corr_anova, _, _ = multipletests(p_values_anova, alpha=0.05, method='BH')
genes_signif_anova = df.columns[:-1][reject_anova]

print("Genes significativamente diferentes entre las 4 clases (ANOVA + BH):")
print(list(genes_signif_anova))
print("Cantidad total: ", len(list(genes_signif_anova)))
```

Genes significativamente diferentes entre las 4 clases (ANOVA + BH):

```
['X1', 'X2', 'X3', 'X9', 'X12', 'X17', 'X21', 'X22', 'X27', 'X29', 'X32', 'X33', 'X36', 'X37', 'X45', 'X46', 'X50', 'X52', 'X54', 'X56', 'X60', 'X64', 'X65', 'X66', 'X67', 'X68', 'X70', 'X73', 'X74', 'X75', 'X76', 'X77', 'X78', 'X80', 'X81', 'X84', 'X85', 'X88', 'X89', 'X90', 'X92', 'X93', 'X94', 'X95', 'X96', 'X97', 'X98', 'X99', 'X100', 'X107', 'X108', 'X116', 'X119', 'X121', 'X123', 'X124', 'X125', 'X127', 'X129', 'X130', 'X131', 'X132', 'X135', 'X139', 'X141', 'X142', 'X144', 'X145', 'X146', 'X147', 'X151', 'X152', 'X153', 'X154', 'X156', 'X159', 'X160', 'X165', 'X166', 'X167', 'X169', 'X171', 'X174', 'X175', 'X178', 'X182', 'X185', 'X186', 'X187', 'X188', 'X190', 'X191', 'X192', 'X194', 'X198', 'X200', 'X201', 'X204', 'X205', 'X207', 'X208', 'X212', 'X213', 'X214', 'X216', 'X217', 'X219', 'X220', 'X222', 'X224', 'X225', 'X229', 'X230', 'X231', 'X233', 'X234', 'X235', 'X236', 'X239', 'X240', 'X243', 'X244', 'X245', 'X246', 'X247', 'X248', 'X251', 'X252', 'X253', 'X254', 'X255', 'X256', 'X257', 'X258', 'X259', 'X260', 'X262', 'X264', 'X266', 'X268', 'X269', 'X272', 'X275', 'X276', 'X277', 'X278', 'X279', 'X280', 'X281', 'X288', 'X289', 'X291', 'X294', 'X302', 'X307', 'X308', 'X309', 'X310', 'X312', 'X314', 'X315', 'X321', 'X323', 'X325', 'X326', 'X328', 'X335', 'X336', 'X338', 'X339', 'X340', 'X345', 'X347', 'X348', 'X354', 'X357', 'X358', 'X359', 'X362', 'X364', 'X365', 'X368', 'X370', 'X371', 'X372', 'X373', 'X374', 'X376', 'X378', 'X380', 'X381', 'X383', 'X384', 'X385', 'X388', 'X390', 'X393', 'X397', 'X399', 'X400', 'X401', 'X402', 'X403', 'X404', 'X405', 'X407', 'X408', 'X409', 'X411', 'X412', 'X417', 'X418', 'X419', 'X421', 'X422', 'X423', 'X424', 'X425', 'X426', 'X428', 'X430', 'X431', 'X433', 'X435', 'X437', 'X439', 'X442', 'X443', 'X444', 'X445', 'X447', 'X449', 'X450', 'X451', 'X454', 'X457', 'X459', 'X462', 'X463', 'X465', 'X468', 'X469', 'X472', 'X473', 'X474', 'X477', 'X479', 'X480', 'X481', 'X482', 'X483', 'X485', 'X490', 'X492', 'X493', 'X494', 'X497', 'X498', 'X499', 'X500', 'X501', 'X502', 'X503', 'X504', 'X508', 'X509', 'X510', 'X511', 'X512', 'X518', 'X521', 'X522', 'X523', 'X528', 'X530', 'X532', 'X533', 'X534', 'X536', 'X538', 'X542', 'X543', 'X544', 'X545', 'X546', 'X552', 'X553', 'X554', 'X555', 'X558', 'X561', 'X566', 'X567', 'X571', 'X572', 'X573', 'X575', 'X582', 'X584', 'X585', 'X586', 'X589', 'X590', 'X594', 'X597', 'X598', 'X603', 'X604', 'X607', 'X608', 'X612', 'X618', 'X621', 'X625', 'X629', 'X632', 'X634', 'X635', 'X636', 'X637', 'X638', 'X639', 'X642', 'X643', 'X645', 'X649', 'X650', 'X651', 'X655', 'X657', 'X658', 'X662', 'X665', 'X666', 'X667', 'X671', 'X672', 'X682', 'X683', 'X687', 'X688', 'X689', 'X693', 'X694', 'X695', 'X696', 'X697', 'X700', 'X701', 'X702', 'X704', 'X705', 'X707', 'X708', 'X712', 'X714', 'X715', 'X716', 'X719', 'X724', 'X725', 'X727', 'X729', 'X730', 'X731', 'X733', 'X734', 'X737', 'X739', 'X741', 'X742', 'X744', 'X745', 'X746', 'X747', 'X749', 'X750', 'X751', 'X753', 'X756', 'X758', 'X760', 'X761', 'X762', 'X766', 'X768', 'X774', 'X778', 'X779', 'X780', 'X783', 'X785', 'X788', 'X789', 'X791', 'X794', 'X795', 'X796', 'X797', 'X799', 'X800', 'X801', 'X803', 'X805', 'X806', 'X808', 'X812', 'X817', 'X818', 'X819', 'X820', 'X823', 'X824', 'X825', 'X826', 'X828', 'X829', 'X831', 'X832', 'X833', 'X836', 'X837', 'X838', 'X841', 'X842', 'X845', 'X846', 'X847', 'X849', 'X850', 'X851', 'X853', 'X855', 'X857', 'X859', 'X861', 'X865', 'X867', 'X871', 'X872', 'X873', 'X875', 'X876', 'X879', 'X880', 'X883', 'X891', 'X901', 'X903', 'X905', 'X906', 'X907', 'X910', 'X912', 'X913', 'X916', 'X917', 'X919', 'X922', 'X928', 'X930', 'X931', 'X932', 'X933', 'X935', 'X937', 'X939', 'X940', 'X941', 'X942', 'X943', 'X945', 'X947', 'X948', 'X951', 'X953', 'X960', 'X962', 'X963', 'X964', 'X965', 'X966', 'X967', 'X970', 'X971', 'X972', 'X973', 'X974', 'X975', 'X976', 'X979', 'X980', 'X982', 'X983', 'X988', 'X990', 'X998', 'X999', 'X1000', 'X1001', 'X1002', 'X1003', 'X1005', 'X1006', 'X1007', 'X1008', 'X1009', 'X1011', 'X1012', 'X1013', 'X1016', 'X1020', 'X1021', 'X1022', 'X1023', 'X1024', 'X1025', 'X1028', 'X1030', 'X1033', 'X1035', 'X1036', 'X1037', 'X1039', 'X1041', 'X1042', 'X1046', 'X1049', 'X1051', 'X1055', 'X1057', 'X1060', 'X1063', 'X1064', 'X1066', 'X1067', 'X1069', 'X1070', 'X1072', 'X1073', 'X1074', 'X1076', 'X1078', 'X1082', 'X1083', 'X1084', 'X1085', 'X1086', 'X1088', 'X1089', 'X1090', 'X1091', 'X1093', 'X1094', 'X1096', 'X1098', 'X1099', 'X1100', 'X1101', 'X1105', 'X1106', 'X1110', 'X1112', 'X1114', 'X1116', 'X1118', 'X1120', 'X1122', 'X1125', 'X1132', 'X1133', 'X1135', 'X1142', 'X1143', 'X1144']
```

144', 'X1149', 'X1151', 'X1154', 'X1155', 'X1156', 'X1157', 'X1158', 'X1159', 'X1160', 'X1162', 'X1164', 'X1165', 'X1167', 'X1169', 'X1183', 'X1185', 'X1186', 'X1187', 'X1188', 'X1189', 'X1190', 'X1194', 'X1195', 'X1196', 'X1197', 'X1198', 'X1201', 'X1203', 'X1204', 'X1206', 'X1207', 'X1208', 'X1209', 'X1210', 'X1211', 'X1213', 'X1217', 'X1220', 'X1221', 'X1223', 'X1224', 'X1225', 'X1226', 'X1227', 'X1228', 'X1230', 'X1235', 'X1236', 'X1237', 'X1238', 'X1242', 'X1244', 'X1245', 'X1246', 'X1250', 'X1252', 'X1254', 'X1257', 'X1258', 'X1261', 'X1262', 'X1263', 'X1268', 'X1269', 'X1272', 'X1273', 'X1274', 'X1279', 'X1280', 'X1281', 'X1282', 'X1283', 'X1284', 'X1286', 'X1287', 'X1292', 'X1295', 'X1296', 'X1297', 'X1298', 'X1299', 'X1301', 'X1304', 'X1308', 'X1309', 'X1312', 'X1315', 'X1317', 'X1318', 'X1319', 'X1320', 'X1324', 'X1325', 'X1327', 'X1330', 'X1331', 'X1335', 'X1336', 'X1339', 'X1343', 'X1345', 'X1346', 'X1347', 'X1348', 'X1353', 'X1354', 'X1356', 'X1359', 'X1360', 'X1366', 'X1368', 'X1369', 'X1371', 'X1372', 'X1373', 'X1375', 'X1377', 'X1379', 'X1382', 'X1384', 'X1385', 'X1386', 'X1387', 'X1389', 'X1392', 'X1393', 'X1394', 'X1395', 'X1396', 'X1401', 'X1402', 'X1403', 'X1405', 'X1406', 'X1414', 'X1415', 'X1416', 'X1417', 'X1419', 'X1420', 'X1422', 'X1423', 'X1424', 'X1427', 'X1431', 'X1434', 'X1436', 'X1437', 'X1438', 'X1440', 'X1442', 'X1443', 'X1449', 'X1452', 'X1453', 'X1457', 'X1458', 'X1460', 'X1464', 'X1465', 'X1469', 'X1471', 'X1474', 'X1479', 'X1481', 'X1484', 'X1486', 'X1487', 'X1488', 'X1489', 'X1490', 'X1493', 'X1494', 'X1496', 'X1497', 'X1498', 'X1500', 'X1501', 'X1502', 'X1504', 'X1509', 'X1510', 'X1511', 'X1514', 'X1515', 'X1517', 'X1518', 'X1521', 'X1522', 'X1524', 'X1525', 'X1530', 'X1531', 'X1535', 'X1536', 'X1538', 'X1542', 'X1545', 'X1553', 'X1554', 'X1556', 'X1562', 'X1563', 'X1565', 'X1570', 'X1572', 'X1573', 'X1574', 'X1575', 'X1576', 'X1577', 'X1579', 'X1584', 'X1587', 'X1594', 'X1595', 'X1597', 'X1598', 'X1599', 'X1601', 'X1606', 'X1607', 'X1608', 'X1610', 'X1613', 'X1619', 'X1621', 'X1624', 'X1626', 'X1627', 'X1628', 'X1634', 'X1635', 'X1636', 'X1640', 'X1642', 'X1643', 'X1644', 'X1645', 'X1646', 'X1647', 'X1648', 'X1649', 'X1651', 'X1654', 'X1655', 'X1657', 'X1659', 'X1661', 'X1662', 'X1664', 'X1665', 'X1668', 'X1670', 'X1671', 'X1673', 'X1674', 'X1675', 'X1677', 'X1678', 'X1680', 'X1682', 'X1683', 'X1685', 'X1691', 'X1693', 'X1697', 'X1698', 'X1700', 'X1701', 'X1702', 'X1704', 'X1706', 'X1707', 'X1708', 'X1709', 'X1711', 'X1714', 'X1716', 'X1717', 'X1721', 'X1723', 'X1725', 'X1727', 'X1728', 'X1729', 'X1730', 'X1731', 'X1734', 'X1735', 'X1736', 'X1738', 'X1739', 'X1740', 'X1741', 'X1743', 'X1746', 'X1747', 'X1748', 'X1749', 'X1750', 'X1751', 'X1754', 'X1756', 'X1760', 'X1761', 'X1764', 'X1765', 'X1768', 'X1770', 'X1772', 'X1773', 'X1775', 'X1776', 'X1778', 'X1782', 'X1783', 'X1784', 'X1785', 'X1788', 'X1790', 'X1791', 'X1792', 'X1795', 'X1797', 'X1799', 'X1800', 'X1801', 'X1803', 'X1804', 'X1806', 'X1808', 'X1810', 'X1811', 'X1813', 'X1814', 'X1820', 'X1821', 'X1822', 'X1823', 'X1826', 'X1827', 'X1828', 'X1831', 'X1833', 'X1834', 'X1837', 'X1839', 'X1841', 'X1842', 'X1843', 'X1846', 'X1847', 'X1851', 'X1853', 'X1854', 'X1855', 'X1856', 'X1857', 'X1858', 'X1859', 'X1860', 'X1861', 'X1862', 'X1863', 'X1864', 'X1865', 'X1867', 'X1868', 'X1869', 'X1870', 'X1873', 'X1875', 'X1876', 'X1878', 'X1880', 'X1881', 'X1882', 'X1884', 'X1886', 'X1888', 'X1890', 'X1891', 'X1894', 'X1896', 'X1899', 'X1901', 'X1902', 'X1905', 'X1906', 'X1907', 'X1908', 'X1909', 'X1910', 'X1911', 'X1912', 'X1914', 'X1915', 'X1916', 'X1917', 'X1920', 'X1924', 'X1928', 'X1929', 'X1930', 'X1931', 'X1932', 'X1934', 'X1937', 'X1942', 'X1945', 'X1949', 'X1951', 'X1954', 'X1955', 'X1956', 'X1960', 'X1962', 'X1963', 'X1964', 'X1965', 'X1967', 'X1968', 'X1973', 'X1974', 'X1975', 'X1977', 'X1978', 'X1979', 'X1980', 'X1981', 'X1987', 'X1990', 'X1991', 'X1992', 'X1993', 'X1994', 'X1995', 'X1997', 'X1998', 'X2000', 'X2001', 'X2006', 'X2007', 'X2012', 'X2016', 'X2017', 'X2018', 'X2019', 'X2020', 'X2021', 'X2022', 'X2031', 'X2032', 'X2039', 'X2040', 'X2041', 'X2042', 'X2046', 'X2047', 'X2049', 'X2050', 'X2051', 'X2058', 'X2059', 'X2060', 'X2070', 'X2072', 'X2080', 'X2081', 'X2082', 'X2083', 'X2086', 'X2087', 'X2088', 'X2089', 'X2092', 'X2093', 'X2094', 'X2095', 'X2096', 'X2099', 'X2101', 'X2102', 'X2104', 'X2105', 'X2114', 'X2115', 'X2116', 'X2117', 'X2118', 'X2120', 'X2122', 'X2123', 'X2127', 'X2131', 'X2132', 'X2134', 'X2136', 'X2138', 'X2139', 'X2140', 'X2144', 'X2145', 'X2146', 'X2148', 'X2149', 'X2150', 'X2152', 'X2157', 'X2159', 'X2161', 'X2162', 'X2163', 'X2166', 'X2168', 'X2170', 'X2171', 'X2172', 'X2175', 'X2176',

```
'X2183', 'X2186', 'X2191', 'X2192', 'X2193', 'X2194', 'X2195', 'X2196', 'X2197', 'X2198', 'X2199', 'X2200', 'X2201', 'X2203', 'X2205', 'X2207', 'X2208', 'X2209', 'X2211', 'X2213', 'X2214', 'X2217', 'X2218', 'X2219', 'X2220', 'X2223', 'X2226', 'X2227', 'X2228', 'X2230', 'X2231', 'X2233', 'X2234', 'X2235', 'X2236', 'X2240', 'X2241', 'X2247', 'X2248', 'X2249', 'X2253', 'X2258', 'X2259', 'X2263', 'X2265', 'X2273', 'X2275', 'X2276', 'X2277', 'X2278', 'X2279', 'X2281', 'X2282', 'X2285', 'X2286', 'X2289', 'X2294', 'X2295', 'X2297', 'X2299', 'X2300', 'X2301', 'X2302', 'X2303', 'X2304']  
Cantidad total: 1162
```

**4. Entrenamiento de modelo SVM con kernel lineal. Se seleccionan los top 10 genes del paso 2 (reconociendo la fuga de datos) y se comparan kernels lineal, polinomial (grado 3) y RBF.**

```
In [13]: # Selección de características: top 10 genes con mayor diferencia de medias  
top_genes = diff_means.head(10).index.tolist()  
  
X = df[top_genes]  
y = df[target_col]  
  
# Separar datos  
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2)  
  
# Modelos SVM  
kernels = {  
    'lineal': SVC(kernel='linear'),  
    'polinomial': SVC(kernel='poly', degree=3),  
    'radial': SVC(kernel='rbf')  
}  
  
resultados = {}  
  
for nombre, modelo in kernels.items():  
    modelo.fit(X_train, y_train)  
    y_pred = modelo.predict(X_test)  
    resultados[nombre] = classification_report(y_test, y_pred, output_dict=True)  
    print(f"\nClasificación con kernel {nombre}:\n")  
    print(classification_report(y_test, y_pred))
```

Clasificación con kernel lineal:

	precision	recall	f1-score	support
1	0.50	0.50	0.50	2
2	1.00	1.00	1.00	6
3	0.75	0.75	0.75	4
4	1.00	1.00	1.00	5
accuracy			0.88	17
macro avg	0.81	0.81	0.81	17
weighted avg	0.88	0.88	0.88	17

Clasificación con kernel polinomial:

	precision	recall	f1-score	support
1	1.00	0.50	0.67	2
2	1.00	1.00	1.00	6
3	0.80	1.00	0.89	4
4	1.00	1.00	1.00	5
accuracy			0.94	17
macro avg	0.95	0.88	0.89	17
weighted avg	0.95	0.94	0.93	17

Clasificación con kernel radial:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	2
2	1.00	1.00	1.00	6
3	1.00	1.00	1.00	4
4	1.00	1.00	1.00	5
accuracy			1.00	17
macro avg	1.00	1.00	1.00	17
weighted avg	1.00	1.00	1.00	17

## 5. Comparación de resultados de los tres modelos anteriores (lineal, polinomial y radial).

```
In [14]: # Comparar precisión macro
scores = {
    k: round(v['macro avg']['f1-score'], 3)
    for k, v in resultados.items()
}

pd.DataFrame.from_dict(scores, orient='index', columns=['F1 macro']).sort_values(by
```

Out[14]:

	F1 macro
radial	1.000
polinomial	0.889
lineal	0.812

El kernel radial obtienen F1 macro = 1.00, mientras que el polinomial cae a 0.89 y el lineal a 0.81; por lo tanto, para este conjunto de características, un RBF es más adecuado. Los genes identificados como más diferenciales coinciden con aquellos que mejor separaron las clases en el SVM, y el kernel RBF resulta óptimo para este subset.