Building a Color Classifier in the Wolfram Language.

Author: Christopher Athaide, EYE Numerics, LLC

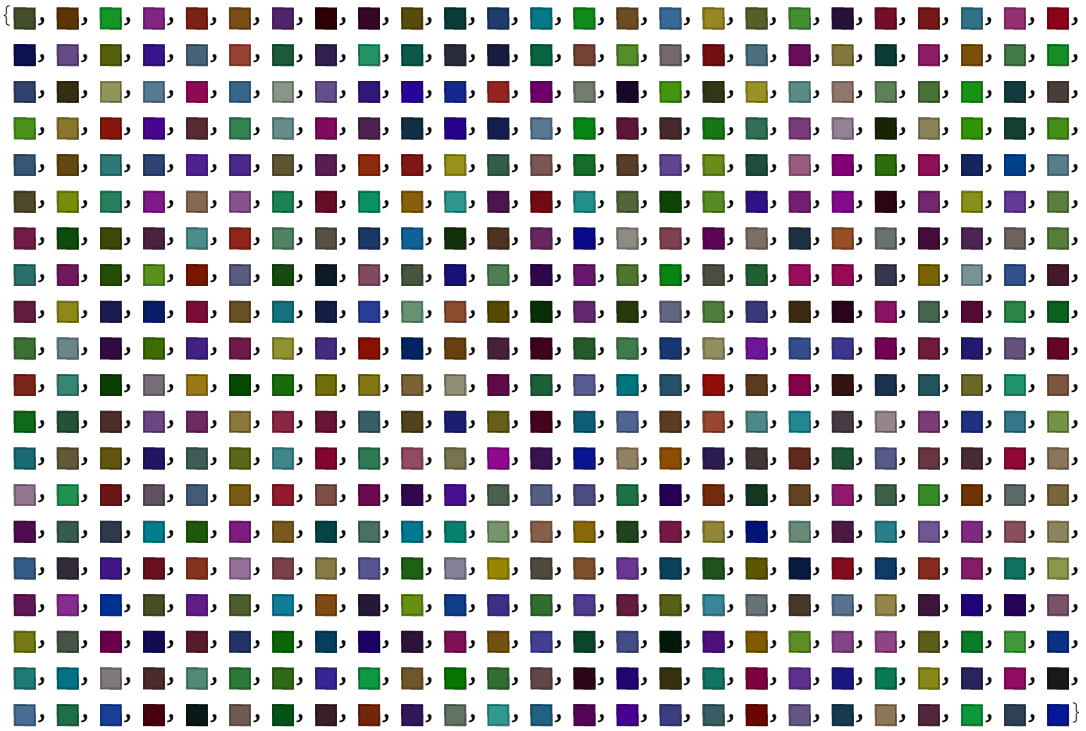
In this workbook, we will build a simple color classifier to illustrate the power of machine learning. There are 2 color types: Dark and Light. We will train a model by providing it with a set of labeled inputs. Then we will test out the model on some random colors.

In[333]:=
```
(* This function converts a list of colors from the
 form RGBColor[r, g, b] to the Hexadecimal format "#AABBCC" *)
mkHexColors[colorList_List] := Block[{hexList},
   hexList = colorList //. RGBColor[r_, g_, b_] → {r, g, b} //.
      {r___, g_Real, b___} :→ {r, BaseForm[Round[g * 255], 16], b} //.
     {r___, f_[g_, 16], b___} :→ {r, ToString[f[g, 16]], b};
   StringJoin @@@ (Map[ToUpperCase[ToString[StringSplit[#, "\n"][[1]]]] &,
        hexList, {2}] //. {y___, x_String, z___} /; StringLength[x] == 1 :→
        {y, "0" ~~ x, z}) /. (x_String :→ "#" ~~ x)
  ];
```

First, we create a random sample of dark colors.

In[334]:=
```
darkColors =
 RGBColor /@ {"#454E2A", "#5D3503", "#139827", "#832485", "#7D2114", "#7B4D17", "#51246C",
   "#2D0003", "#350726", "#574C0C", "#0B3C39", "#1E3C6E", "#057985", "#108C1D", "#6C4D23",
   "#386998", "#998624", "#556027", "#428E31", "#281139", "#770E29", "#76181A", "#317189",
   "#973073", "#8E011A", "#0C1150", "#674B84", "#57600F", "#39158D", "#47627A", "#994336",
   "#1B5B39", "#321F4E", "#249467", "#0C584D", "#2A2D3C", "#181C40", "#086441",
   "#754233", "#558D29", "#74676E", "#740F12", "#4A707F", "#680E5B", "#84773D",
   "#0C3C36", "#8C1E63", "#7B4F0C", "#437949", "#178E25", "#33426D", "#352E12",
   "#92945E", "#557B93", "#900A53", "#386889", "#869789", "#644E8C", "#32177A",
   "#280599", "#16288D", "#942421", "#6D0169", "#737C6D", "#19072A", "#42950E",
   "#313617", "#989227", "#5A8C87", "#90776C", "#5E8058", "#477135", "#189414",
   "#133B3E", "#493F38", "#4B941B", "#89772B", "#89160D", "#48058F", "#572B31",
   "#368153", "#658E8B", "#810A5E", "#4C234F", "#122F44", "#28038A", "#161E4F",
   "#587893", "#068817", "#5B1738", "#46292A", "#177716", "#306E55", "#783D7A",
   "#938194", "#182405", "#8A8359", "#319606", "#153F31", "#438F18", "#375574",
   "#644812", "#2E7875", "#2F4474", "#4F2293", "#4D288B", "#605434", "#5B1C53",
   "#8F2A0A", "#7F1814", "#97921C", "#315E48", "#7B5753", "#166C29", "#583E2A",
   "#614492", "#698C1A", "#1E4E3B", "#985C80", "#830074", "#2C6D0E", "#8F1056",
   "#15255E", "#00408D", "#557D8C", "#504929", "#778C0C", "#277F5E", "#81198B",
   "#856954", "#86538C", "#1B8253", "#690C28", "#0B9065", "#8A5F0B", "#2F978F",
   "#4E1450", "#720C13", "#22908D", "#51663B", "#0D4603", "#578C24", "#311186",
   "#762075", "#85098C", "#2B0514", "#72276F", "#8A8F19", "#643997", "#5C7F3F",
```

```
"#741C48", "#0D4C0B", "#3F4606", "#4A223F", "#4A8D8C", "#922519", "#4E8464",
"#565143", "#1B3868", "#11609A", "#133209", "#4D3320", "#68255E", "#0A098B",
"#8D8B83", "#81414F", "#5D0555", "#7D6F66", "#1C3143", "#984F26", "#68736F",
"#460D3B", "#4D2554", "#6F615B", "#577E39", "#2A706C", "#6F195F", "#244E07",
"#58911A", "#771702", "#585C7E", "#154B0F", "#0D1B27", "#804B61", "#46563E",
"#171178", "#4F7E53", "#2E074D", "#69176D", "#51752F", "#078613", "#4D4C40",
"#215F34", "#990D5A", "#980956", "#36374C", "#786405", "#759398", "#33518E",
"#45192C", "#631D41", "#8D8817", "#1A1B51", "#081D6A", "#7A0D38", "#675124",
"#15717B", "#131D45", "#2C3D98", "#669472", "#8B4D2B", "#574A00", "#082F05",
"#642972", "#273A0C", "#61677F", "#4D7E3D", "#3A3878", "#3D2C12", "#28041D",
"#891363", "#45654E", "#550B35", "#2E8448", "#09601F", "#3C7035", "#6C858B",
"#310B41", "#3E6C04", "#412183", "#6F194B", "#91922D", "#452B80", "#881102",
"#042365", "#69410F", "#47203A", "#3E041E", "#255B2B", "#407C49", "#193873",
"#918F60", "#6E1599", "#354D8C", "#3C3690", "#730052", "#701A3C", "#231A6F",
"#65537D", "#650629", "#7B2619", "#368673", "#0B3E05", "#766D79", "#997916",
"#054902", "#186D0B", "#706F07", "#817815", "#796332", "#928F77", "#600B48",
"#1C5F38", "#585E94", "#01757E", "#29526A", "#920D09", "#5E3B1F", "#86004A",
"#341410", "#1C3350", "#1F555B", "#6C6828", "#219269", "#7B5740", "#0F6B1B",
"#245238", "#4C2D29", "#6B4683", "#6D2962", "#89773B", "#892644", "#691437",
"#395E65", "#51431A", "#1A1F73", "#665F19", "#45031E", "#0C5F76", "#516595",
"#5C3D21", "#974732", "#4C878A", "#218790", "#473945", "#95848C", "#7E3C76",
"#1F3181", "#347789", "#759344", "#196C73", "#645936", "#615610", "#211564",
"#415B59", "#5D6619", "#3E888C", "#830630", "#2E7A54", "#934A65", "#787451",
"#920891", "#39134E", "#0A1593", "#918265", "#8E4D04", "#2D1B54", "#3E3633",
"#622A1E", "#1D5336", "#565A88", "#673641", "#482A37", "#910836", "#8A755A",
"#927692", "#20914F", "#6C1715", "#605261", "#435B76", "#795C16", "#911A2D",
"#7D4F45", "#6C0A52", "#2E094E", "#491190", "#49604E", "#565F82", "#4B4C84",
"#1E7045", "#250055", "#732B0D", "#123920", "#614421", "#921971", "#3E6048",
"#388B29", "#6F3306", "#5A6A68", "#7A663D", "#4F0D50", "#365C4E", "#2F394B",
"#007E8E", "#1D5908", "#7E1B80", "#7A591B", "#054444", "#4A7060", "#047791",
"#078170", "#77956B", "#886049", "#8A6A0B", "#1F441F", "#741A44", "#928537",
"#021476", "#698A76", "#4F1843", "#297F88", "#715696", "#7E2B7F", "#88505C",
"#8D835C", "#365D87", "#322B38", "#411788", "#691020", "#87341E", "#957198",
"#7B414A", "#867B47", "#575592", "#1C6311", "#838299", "#968602", "#504A3E",
"#7F512D", "#683793", "#0B415A", "#1F531E", "#5E5600", "#081A3F", "#840D1D",
"#0E3B68", "#862B1F", "#841D66", "#12715E", "#8B974A", "#5E1759", "#862C8F",
"#023196", "#494D23", "#611C86", "#4D5E2E", "#0F7D96", "#804A14", "#261936",
"#678F10", "#113C89", "#3F2E88", "#2E6E2E", "#503C8E", "#631C3F", "#576015",
"#398699", "#657276", "#473928", "#58738F", "#938649", "#3D173B", "#20037B",
"#250356", "#7A536B", "#757A15", "#485348", "#69024A", "#140B55", "#581B2B",
"#243466", "#0C6605", "#063D60", "#1D0266", "#2D1436", "#7E1256", "#715211",
"#414092", "#0B472B", "#424C85", "#021B08", "#4C117E", "#7F5B01", "#5E8D27",
"#844487", "#8F4484", "#5F5E19", "#0D7C28", "#44983C", "#0B3184", "#207A78",
"#037286", "#807B79", "#4B2C2B", "#518978", "#2B7B38", "#316A16", "#372497",
"#0E9842", "#70572B", "#097602", "#326E31", "#614745", "#2D061A", "#230473",
"#383110", "#127464", "#7C033F", "#5E318E", "#1A1684", "#097B54", "#888818",
"#292460", "#920B64", "#1B1819", "#486E95", "#1C6F49", "#193E97", "#4A010F",
"#091818", "#6F5B54", "#065318", "#3B1E26", "#732609", "#32175D", "#607463",
"#2F9990", "#1F6283", "#56045A", "#4A0191", "#3C3E84", "#385D64", "#6B0603",
"#665587", "#173A4F", "#8E7659", "#54263E", "#0C9739", "#2E4057", "#061899"}
```

Out[334]= 

Next, we create a random sample of light colors.

In[335]:=
```
lightColors =
  RGBColor /@ {"#CBF7BE", "#D4BFDD", "#C1EDC4", "#CDD6C2", "#E8F3E4", "#C3D7D0", "#C7B7D6",
    "#E6CEE5", "#DEF8CE", "#FCEEC4", "#BCFFDA", "#DCFCE4", "#BEF9B8", "#F2F2C0", "#D9B8C0",
    "#D5DEBC", "#C2D0B3", "#F7EAF6", "#E8C7E1", "#E8CCF2", "#C0F7B6", "#C4DAE9", "#C7C4E9",
    "#C2F4FB", "#FFC9E3", "#BEF3C9", "#FDF5BC", "#DAEBCE", "#DFECD5", "#FABAB9", "#DBDAC6",
    "#EFD0CE", "#E1EBE3", "#DAD6F3", "#F3D7DE", "#FBF3C9", "#CCEFD3", "#E6EAB4",
    "#E0C9D1", "#DBE9DA", "#F4D6CC", "#D1D3B8", "#E6B7B8", "#E1FFBF", "#D3F9C6",
    "#C5EACA", "#BFFDD1", "#D7F4EC", "#FBE1D7", "#C4BBB6", "#F0DEFD", "#D3CDB7",
    "#CBC6F5", "#F1C2CC", "#C4E1D8", "#DDE7C7", "#DFF9BD", "#E4E8EB", "#EBD6F1",
    "#ECE0D8", "#F7F8E5", "#E6CCBD", "#CAF7D9", "#DDB9DC", "#CDE8E5", "#C4D1FB",
    "#D4C3F6", "#D6F3F6", "#B8C2E7", "#E0E1F8", "#F3CFB5", "#C7CABC", "#B4D8DF",
    "#E0CDF9", "#C0F8E7", "#C8F7CA", "#B5DAF8", "#E0CAB4", "#FBD5B3", "#B9BBEB",
    "#E8B3CD", "#E6BCD1", "#F0BCC9", "#C0F7FD", "#C7B7B6", "#B9CBDB", "#E8EFFC",
    "#DFF9F2", "#E2B4E8", "#E5F9F9", "#D2DBFF", "#F6C8F9", "#D9EBF6", "#D3F9E4",
    "#C0E7FA", "#C1B6F6", "#F5DBCA", "#E2FBE2", "#D4D9D0", "#E6FDF2", "#D4D8C0",
    "#C7E9D5", "#C1C7DA", "#B8F4D7", "#C1F7DC", "#B5C6B4", "#E3F6D6", "#FBCAD4",
    "#C3B7FB", "#E1C1EF", "#ECD2E0", "#E2C6E6", "#FFCAFA", "#C9B5C7", "#D6FCCE",
    "#E8D3C4", "#B9B5C7", "#EDF0FE", "#D9EAF3", "#D1EDDE", "#C2E7CD", "#BAE9F8",
    "#B9E6EA", "#C7E3F8", "#E2D3F3", "#E3EDE2", "#CDBCBC", "#F2D1D5", "#B4E4C1",
    "#B5EFCA", "#CBF5C9", "#F9EFE3", "#DFD2CC", "#C2C3C1", "#DBB7B7", "#FFB3CB",
    "#D8D1C8", "#E6DED9", "#D8E5E8", "#B7F6FA", "#E8D3D9", "#DBF4E8", "#FFCFEE",
    "#F9D7FD", "#F1B3FE", "#E2BFC9", "#E5E0F0", "#D3C8E8", "#E9F6BF", "#F0DAEE",
    "#C5F0FD", "#B6DCB5", "#F2B5DD", "#B3CAC8", "#C9DAC1", "#B9BBF6", "#E9DCF8",
    "#E4C1B9", "#FBE8DF", "#CFB8ED", "#E0D5E2", "#EDFDB6", "#E8B4CB", "#C9BAE7",
    "#E1B3C8", "#D8EBE8", "#E6D6FB", "#E4C9C5", "#DEF1E3", "#FBBBCC", "#BBC9E6",
```

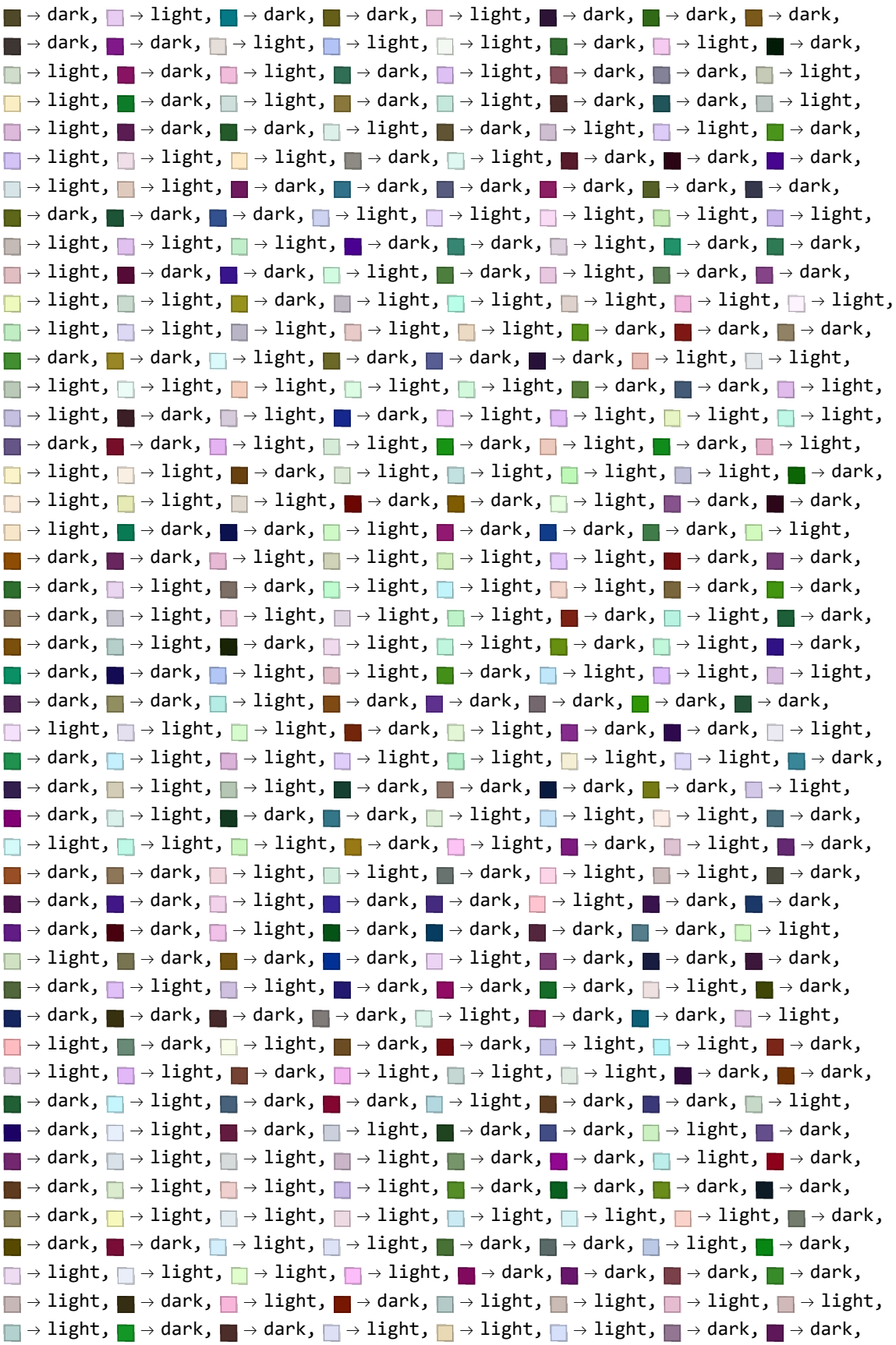
"#C1F7DB", "#CECCD5", "#DEBCFC", "#EDFAFC", "#DBFAFB", "#CEC0DE", "#D7ECD8",
"#F4E0DC", "#DFDDD3", "#FCF3FE", "#B3F3BB", "#DFC0F5", "#D7DBDB", "#E1BBF5",
"#EFC1C6", "#DCC3D9", "#F8B6DA", "#FDEDE7", "#DDF0E9", "#C8DBDC", "#CBEBF4",
"#CCEFC4", "#C3B3FD", "#CFCCE1", "#C5C3C8", "#DEE0F5", "#B8E0E9", "#E5F3D1",
"#E1BFF7", "#DDF5EB", "#D3F3CA", "#B5DFEF", "#D3F0DD", "#F7D8C8", "#D9C9CC",
"#D1DDCC", "#B3D9DC", "#BBE7EA", "#EDCABD", "#CAF4CB", "#BFD5C4", "#ECD4F4",
"#F8E6CA", "#CFF3D5", "#D4C8C0", "#E2BDD1", "#FBD1B4", "#FFEAC5", "#C7D9C9",
"#EDF3F4", "#C1F2D2", "#B6CFCA", "#E4DCCA", "#CBD0DC", "#BEE7C6", "#D2F9BE",
"#E4B8D9", "#EBDDCF", "#FFCCC4", "#FFDCEA", "#B3D2CF", "#D7DDCE", "#E6FEE0",
"#E1D9CD", "#D4F2E4", "#E8CBC9", "#F6E6CB", "#F3C0B6", "#CFCBBA", "#C4E1FC",
"#D0E2C3", "#F8FDEA", "#CFDFDB", "#FCD1D4", "#DDE1CC", "#CBB6D2", "#D4FAFA",
"#E0B5FD", "#F9D2FA", "#C8B6ED", "#FEBCC1", "#D7C2F6", "#C9DBCE", "#DEEED7",
"#C9DAE7", "#C4C1DF", "#F3F9F2", "#CAB4F4", "#C7B9C9", "#EABBB3", "#D6CBDA",
"#FFBCB8", "#DEE4E4", "#F0DFE8", "#E1BBF1", "#CCCDB7", "#C2D3DB", "#DCB9CA",
"#EAE5C7", "#E2F0B9", "#DBD1BB", "#C2CCEB", "#C6FADB", "#F6CBF1", "#EEBDC5",
"#F5FCD1", "#E0E7D6", "#D1F0BC", "#C5F5FD", "#E2E5C9", "#F3B3EF", "#E0BEC1",
"#D1B9B9", "#EAEFBE", "#BDC7B5", "#F6DFFD", "#E4D4D1", "#C4E6E2", "#F8C3F2",
"#F5F6CB", "#C0E7D1", "#CBCEC7", "#CDD3F0", "#E6EACE", "#F3D2EC", "#F1E5B7",
"#F1D6D1", "#C2B8EE", "#FACFDE", "#C5CBB7", "#BCC6C2", "#C6D7F4", "#F6E1FC",
"#EAE3CD", "#DABCD8", "#E0DCC8", "#B5C1D2", "#E8B9D5", "#C3E5EC", "#B9C0CC",
"#B3C6CD", "#C8B4F7", "#EEBFF0", "#CFBBE2", "#E2EBF0", "#C1EECB", "#C1C7E2",
"#C2FBFF", "#CCE6C6", "#C2FDCC", "#DFF5E3", "#F2BDDC", "#E7F4C3", "#DABEE2",
"#DDEAD7", "#D5FCD5", "#FACCCC", "#EEBAC4", "#DDD8F4", "#F9DBF5", "#DBB3D9",
"#BCB7D6", "#C0D2D1", "#E1FECD", "#F0C6E9", "#E9B5BF", "#D7E0FB", "#CEC2D9",
"#D3F9DD", "#DFD9F9", "#B8F3E2", "#BFFDDC", "#D9F8DA", "#F0CBF3", "#F8CBD4",
"#BCEFEB", "#C0BFD2", "#C4C8CF", "#B6D0C9", "#EDDAE3", "#CAD2C8", "#C6C4CE",
"#F7F9BC", "#C6EBB5", "#F7CBD7", "#D3EEFC", "#D9FDC3", "#F6D9B3", "#FCC4DA",
"#BEF8E7", "#F6E6FA", "#E0CFE4", "#B9FEE8", "#E4DEDD", "#C1C1D9", "#F5F0D5",
"#B6BBDF", "#D3CDE9", "#BFB9C4", "#C2E1E1", "#EAEEF8", "#FDD5E8", "#B4E9B7",
"#D7D8BA", "#BED4F0", "#F1C1EA", "#B9C7D0", "#CBE8E1", "#F4BDFC", "#D9F0EB",
"#BFB7ED", "#D0CFE1", "#DBF0F7", "#E1EBE9", "#DDCBF8", "#CCF0C2", "#E0D8CC",
"#B4F7C7", "#B3C7F6", "#BCEDCD", "#EDDAC4", "#FEC5E5", "#CDD1C9", "#ECFCF6",
"#EFCEE0", "#B3EAEE", "#F8EADF", "#CFBED2", "#BBB3E7", "#E6C7B7", "#BEC1DB",
"#B8EDD4", "#D5D5C3", "#D8FBC4", "#B7F0C2", "#FFC4CF", "#CDBEF4", "#C2BBD2",
"#D9ECE1", "#ECFCF0", "#FAEAD8", "#C4C6FA", "#C4E8DB", "#CCC7BE", "#D6EEC2",
"#FCC2CB", "#EFD7BE", "#ECD6D5", "#FEDEB7", "#B8E1BE", "#EEDCF3", "#BFD9C1",
"#F9F2CC", "#D8B3D9", "#C3FBFC", "#F6CBBD", "#CDEDDD", "#C9F4E3", "#FBD2C9",
"#C8BFEC", "#EFE1E1", "#C5C0E0", "#D0DAC5", "#D6E2E9", "#DAE9E3", "#E5D2BE",
"#C2D7D7", "#E6D7CF", "#B4C3F6", "#BCD4D9", "#F1F1C3", "#DEE2F7", "#F1B7C5",
"#E0F2F5", "#E8F1F6", "#F9DFBD", "#F8EBFE", "#E0BBEB", "#DAB5B9", "#E2B8F9",
"#B9DCC5", "#FCECFC", "#E7EFE3", "#CFC5D9", "#F5CFBC", "#CBC5B8", "#CABBB5",
"#E4C7FA", "#D9E2E9", "#B6ECE6", "#DEC4E3", "#B7D1D6", "#FDC3F5", "#DFD2DF",
"#C5D8D3", "#CDF7FA", "#C5B4D9", "#E4C4D3", "#E9DAB5", "#F0C9F9", "#D7CBED",
"#E4FCFD", "#BFF6DD", "#B5CFF8", "#EBD6E8", "#F8CCC6", "#D8BFD1", "#FDBDFA",
"#B7F4C5", "#D3C4C6", "#B6DAE0", "#BFD2CB", "#CFFAC7", "#D8DBD4", "#C7D3C7",
"#E2BAFC", "#F4B8C5", "#D0F5E9", "#E6B3F2", "#D2FCE1", "#DEC3D2", "#E5F5E2",
"#EBE9F1", "#FAF5BF", "#E8BEDA", "#D8FBF4", "#BBCAB8", "#E0CABE", "#EEF9BD"}

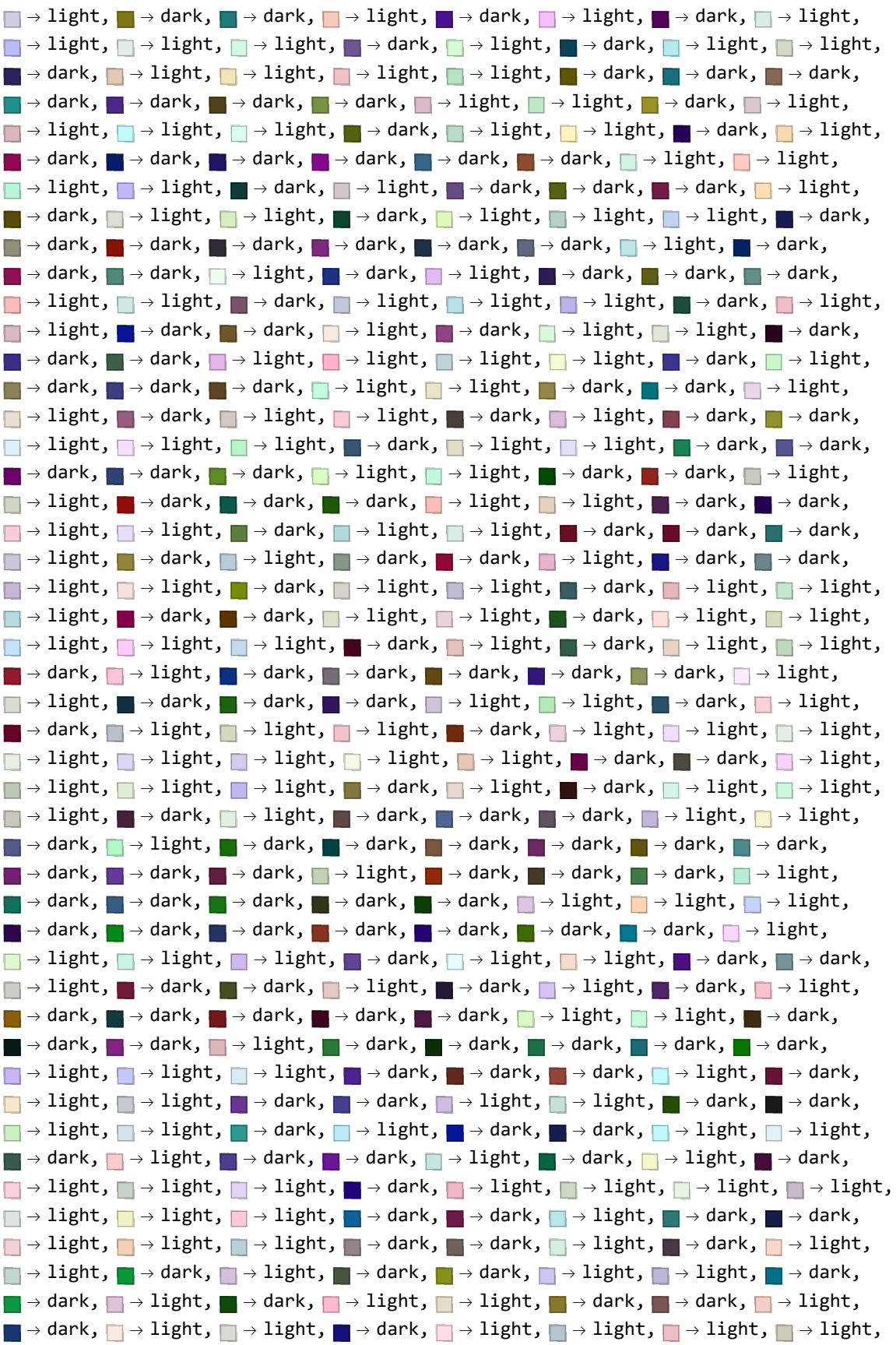
Out[335]=

Now, we create the training set comprising both sets of light and dark colors.

In[336]:=
```
trainingSet =
 RandomSample[Join[Thread[darkColors → "dark"], Thread[lightColors → "light"]],
  Length[darkColors] + Length[lightColors]]
```

Out[336]=


→ dark, → light, → dark, → dark, → light, → dark, → dark, → dark,
→ dark, → dark, → light, → light, → light, → dark, → light, → dark,
→ light, → dark, → light, → dark, → light, → dark, → dark, → light,
→ light, → dark, → light, → dark, → light, → dark, → dark, → light,
→ light, → dark, → dark, → light, → dark, → light, → light, → dark,
→ light, → light, → light, → dark, → light, → dark, → dark, → dark,
→ light, → light, → dark, → dark, → dark, → dark, → dark, → dark,
→ dark, → dark, → dark, → light, → light, → light, → light, → light,
→ light, → light, → light, → dark, → dark, → light, → dark, → dark,
→ light, → dark, → dark, → light, → dark, → light, → dark, → dark,
→ light, → light, → light, → light, → light, → light, → light, → light,
→ light, → light, → light, → light, → light, → dark, → dark, → dark,
→ dark, → dark, → light, → dark, → dark, → dark, → light, → light,
→ light, → light, → light, → light, → light, → dark, → dark, → light,
→ light, → dark, → light, → dark, → light, → light, → light, → light,
→ dark, → dark, → light, → light, → dark, → light, → dark, → light,
→ light, → light, → dark, → light, → light, → light, → light, → dark,
→ light, → light, → light, → dark, → dark, → light, → dark, → dark,
→ light, → dark, → dark, → light, → dark, → dark, → dark, → light,
→ dark, → dark, → light, → light, → light, → light, → dark, → dark,
→ dark, → light, → dark, → light, → light, → light, → dark, → dark,
→ dark, → light, → light, → light, → light, → dark, → light, → dark,
→ dark, → light, → dark, → light, → light, → dark, → light, → dark,
→ dark, → dark, → light, → light, → dark, → light, → light, → light,
→ dark, → dark, → dark, → light, → dark, → dark, → dark, → dark,
→ light, → light, → light, → dark, → light, → dark, → dark, → light,
→ dark, → light, → light, → light, → light, → light, → light, → dark,
→ light, → light, → dark, → light, → dark, → dark, → dark, → light,
→ dark, → dark, → dark, → dark, → dark, → dark, → dark, → dark,
→ dark, → dark, → light, → light, → dark, → light, → light, → dark,
→ dark, → dark, → light, → dark, → dark, → light, → dark, → dark,
→ dark, → dark, → light, → dark, → dark, → dark, → dark, → light,
→ light, → dark, → dark, → dark, → light, → dark, → dark, → dark,
→ dark, → light, → light, → dark, → dark, → dark, → light, → dark,
→ dark, → dark, → dark, → dark, → light, → dark, → dark, → light,
→ light, → dark, → light, → dark, → dark, → light, → light, → dark,
→ light, → light, → dark, → light, → light, → light, → dark, → dark,
→ dark, → light, → dark, → dark, → light, → dark, → dark, → light,
→ dark, → light, → dark, → dark, → light, → dark, → dark, → light,
→ dark, → light, → light, → light, → dark, → dark, → light, → dark,
→ dark, → light, → light, → light, → dark, → dark, → dark, → dark,
→ dark, → light, → light, → light, → light, → light, → light, → dark,
→ dark, → dark, → light, → light, → dark, → dark, → light, → dark,
→ light, → light, → light, → light, → dark, → dark, → dark, → dark,
→ light, → dark, → light, → dark, → light, → light, → light, → light,
→ light, → dark, → dark, → light, → light, → light, → dark, → dark,


→ light, → dark, → dark, → light, → dark, → light, → dark, → light,
→ light, → light, → light, → dark, → light, → dark, → light, → light,
→ dark, → light, → light, → light, → light, → dark, → dark, → dark,
→ dark, → dark, → dark, → dark, → light, → light, → dark, → light,
→ light, → light, → light, → dark, → light, → light, → dark, → light,
→ dark, → dark, → dark, → dark, → dark, → dark, → light, → light,
→ light, → light, → dark, → light, → dark, → dark, → dark, → light,
→ dark, → light, → light, → dark, → light, → light, → light, → dark,
→ dark, → dark, → dark, → dark, → dark, → dark, → light, → dark,
→ dark, → dark, → light, → dark, → light, → dark, → dark, → dark,
→ light, → light, → dark, → light, → light, → light, → dark, → light,
→ light, → dark, → dark, → light, → dark, → light, → light, → dark,
→ dark, → dark, → light, → light, → light, → light, → dark, → light,
→ dark, → dark, → dark, → light, → light, → dark, → dark, → light,
→ light, → light, → light, → dark, → light, → light, → dark, → dark,
→ dark, → dark, → dark, → dark, → light, → light, → dark, → dark, → light,
→ light, → dark, → dark, → dark, → dark, → light, → light, → dark, → dark,
→ light, → light, → dark, → light, → light, → dark, → dark, → dark,
→ light, → dark, → light, → dark, → dark, → light, → dark, → dark,
→ light, → light, → dark, → light, → light, → light, → dark, → light, → light,
→ light, → dark, → dark, → light, → light, → dark, → light, → light,
→ light, → light, → dark, → light, → dark, → light, → light, → light,
→ dark, → light, → dark, → dark, → dark, → dark, → dark, → light,
→ light, → dark, → dark, → dark, → light, → light, → dark, → light,
→ dark, → light, → light, → light, → dark, → light, → light, → light,
→ light, → light, → light, → light, → light, → dark, → dark, → light,
→ light, → light, → light, → dark, → light, → dark, → light, → light,
→ light, → dark, → light, → dark, → dark, → dark, → light, → light,
→ dark, → light, → dark, → dark, → dark, → dark, → dark, → dark,
→ dark, → dark, → dark, → light, → dark, → dark, → dark, → light,
→ dark, → dark, → dark, → dark, → dark, → light, → light, → light,
→ dark, → dark, → dark, → light, → dark, → dark, → dark, → dark, → dark,
→ light, → light, → light, → dark, → dark, → dark, → light, → dark,
→ light, → light, → dark, → dark, → light, → light, → dark, → dark,
→ light, → light, → dark, → dark, → light, → dark, → dark, → light,
→ dark, → light, → dark, → dark, → dark, → light, → dark, → light, → dark,
→ light, → light, → light, → dark, → light, → light, → light, → light,
→ light, → light, → light, → dark, → dark, → light, → dark, → dark,
→ light, → light, → light, → dark, → dark, → light, → dark, → light,
→ light, → dark, → light, → dark, → dark, → light, → light, → dark,
→ dark, → light, → dark, → dark, → light, → light, → dark, → light,
→ dark, → light, → light, → dark, → light, → light, → light, → light,

```
■ → dark, ■ → dark, ■ → light, ■ → light, ■ → dark, ■ → dark, ■ → dark, ■ → light,
■ → light, ■ → dark, ■ → dark, ■ → dark, ■ → light, ■ → light, ■ → dark, ■ → light,
■ → light, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → light,
■ → light, ■ → light, ■ → dark, ■ → light, ■ → light, ■ → light, ■ → dark, ■ → light,
■ → dark, ■ → light, ■ → light, ■ → light, ■ → dark, ■ → light, ■ → light, ■ → dark,
■ → dark, ■ → dark, ■ → light, ■ → dark, ■ → dark, ■ → light, ■ → light, ■ → light,
■ → dark, ■ → light, ■ → light, ■ → light, ■ → dark, ■ → dark, ■ → light, ■ → light,
■ → dark, ■ → light, ■ → light, ■ → light, ■ → light, ■ → light, ■ → light, ■ → dark,
■ → light, ■ → light, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → light,
■ → light, ■ → light, ■ → dark, ■ → light, ■ → dark, ■ → light, ■ → light, ■ → dark,
■ → light, ■ → dark, ■ → light, ■ → light, ■ → light, ■ → light, ■ → dark, ■ → dark,
■ → light, ■ → light, ■ → dark, ■ → light, ■ → dark, ■ → dark, ■ → light, ■ → light,
■ → dark, ■ → dark, ■ → light, ■ → dark, ■ → dark, ■ → light, ■ → light, ■ → light,
■ → dark, ■ → dark, ■ → light, ■ → light, ■ → light, ■ → light, ■ → dark, ■ → light}
```

We build a Naive Bayes Classifier with 2 classes.

```
In[337]:= colorClassifier = Classify[trainingSet]

Out[337]= ClassifierFunction[ Input type: Color
                              Classes: dark, light ]
```

Now we create some random colors ...

```
In[338]:= moreColors = RandomColor[50]

Out[338]= {■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■,
           ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■}
```

.. and we test the model on these colors.

```
In[339]:= Map[# → colorClassifier[#] &, moreColors]

Out[339]= {■ → dark, ■ → light, ■ → dark, ■ → dark, ■ → dark, ■ → light, ■ → dark, ■ → dark,
  ■ → light, ■ → light, ■ → dark, ■ → light, ■ → light, ■ → light, ■ → dark, ■ → dark,
  ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark,
  ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark,
  ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → light, ■ → dark, ■ → light,
  ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark, ■ → dark}
```

Here is some Information about the classifier.

In[340]:=
```
Information[colorClassifier]
```

Out[340]=

Classifier information

Data type	Color
Classes	dark, light
Accuracy	(99.5 ± 0.6)%
Method	LogisticRegression
Single evaluation time	1.62 ms/example
Batch evaluation speed	64.1 examples/ms
Loss	0.0213 ± 0.0078
Model memory	235. kB
Training examples used	1000 examples
Training time	1. / s