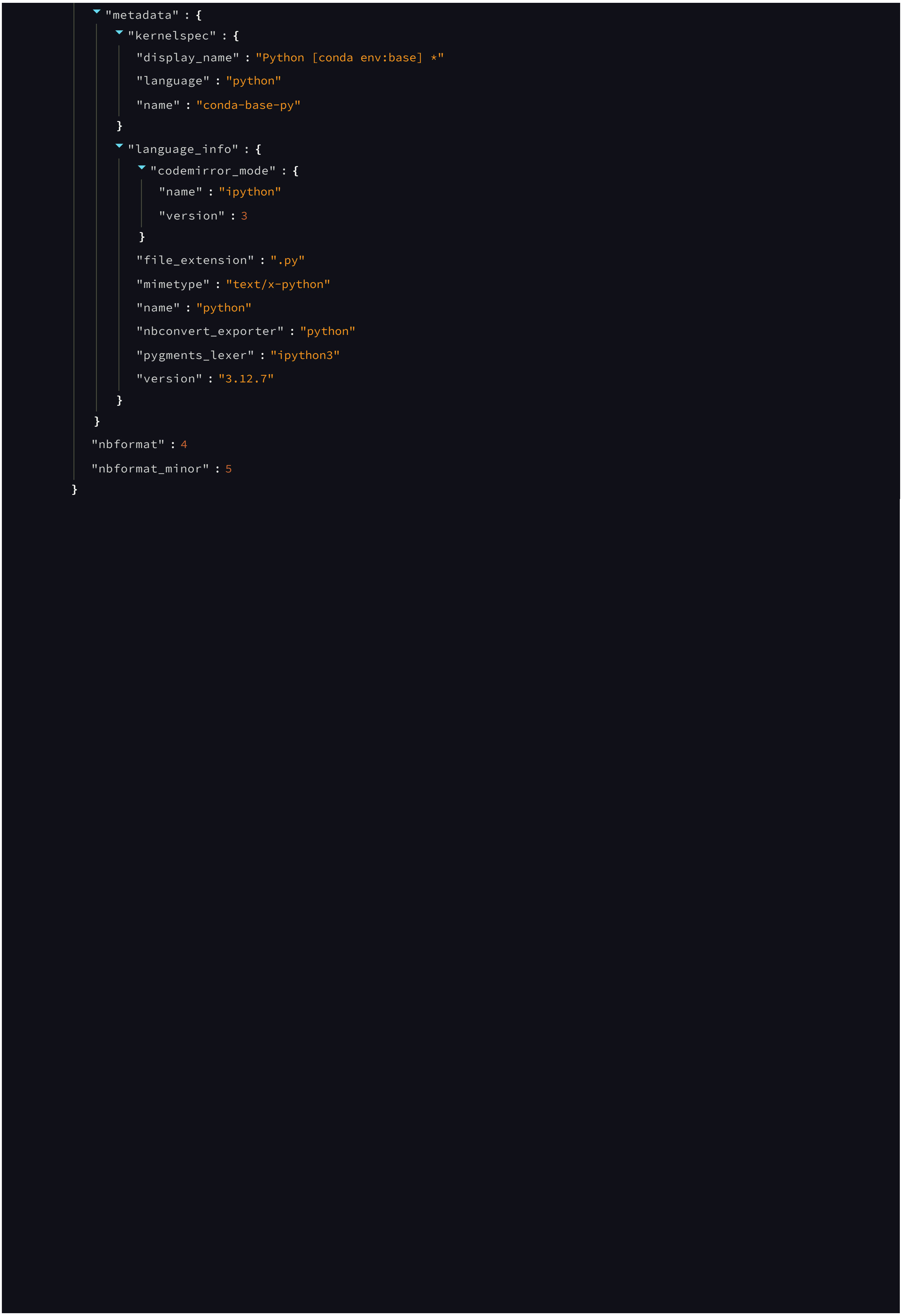
```
{
  "cells" : [
    0 : {
      "cell_type" : "code"
      "execution_count" : 5
      "id" : "86725496-29cf-4ee8-8ad2-bfd6f53026a7"
      "metadata" : {}
      "outputs" : [
        0 : {
          "data" : {
            "text/plain" : [
              0 : "DeltaGenerator()"
            ]
          }
          "execution_count" : 5
          "metadata" : {}
          "output_type" : "execute_result"
        }
      ]
      "source" : [
        0 : "import streamlit as st
            "
        1 : "import pandas as pd
            "
        2 : "import seaborn as sns
            "
        3 : "import matplotlib.pyplot as plt
            "
        4 : "
            "
        5 : "# Load data
            "
        6 : "df = pd.read_csv("product_info.csv")
            "
        7 : "
            "
        8 : "# Drop missing
            "
        9 : "df = df.dropna(subset=["brand_name", "primary_category", "rating", "child_max_price"])
            "
        10 : "
             "
        11 : "# Sidebar filters
             "
        12 : "st.sidebar.title("Filters")
             "
        13 : "brands = st.sidebar.multiselect("Select Brands", options=df["brand_name"].unique(), default=df["branc
             "
        14 : "categories = st.sidebar.multiselect("Select Categories", options=df["primary_category"].unique(), det
             "
        15 : "price_range = st.sidebar.slider("Price Range", float(df["child_max_price"].min()), float(df["child_ma
             "
        16 : "
             "
        17 : "# Filter data
             "
        18 : "filtered_df = df[
             "
19 : "    (df["brand_name"].isin(brands)) &
     "
20 : "    (df["primary_category"].isin(categories)) &
     "
21 : "    (df["child_max_price"].between(price_range[0], price_range[1]))
     "
22 : "]
     "
23 : "
     "
24 : "# Metrics
     "
25 : "st.title("Sephora Product Explorer")
     "
26 : "st.metric("Average Rating", round(filtered_df["rating"].mean(), 2))
     "
27 : "st.metric("Average Price", f"${round(filtered_df['child_max_price'].mean(), 2)}")
     "
28 : "
     "
29 : "# Scatter plot
     "
30 : "st.subheader("Rating vs. Price")
     "
31 : "fig1, ax1 = plt.subplots()
     "
32 : "sns.scatterplot(data=filtered_df, x="child_max_price", y="rating", ax=ax1, color="purple")
     "
33 : "st.pyplot(fig1)
     "
34 : "
     "
35 : "# Rating histogram
     "
36 : "st.subheader("Rating Distribution")
     "
37 : "fig2, ax2 = plt.subplots()
     "
38 : "sns.histplot(filtered_df["rating"], bins=20, kde=True, ax=ax2, color="orange")
     "
39 : "st.pyplot(fig2)
     "
40 : "
     "
41 : "# Show filtered data
     "
42 : "st.subheader("Filtered Products")
     "
43 : "st.dataframe(filtered_df[["product_name", "brand_name", "rating", "child_max_price"]].reset_index(dro
     "
44 : "
     "
   ]
  }
 1 : {
   "cell_type" : "code"
   "id" : "5d2eb2ad-a19b-4ef2-b448-c1f55c315a68"
   "outputs" : []
   "source" : []
  }
]
```



```
    "metadata" : {
      "kernelspec" : {
        "display_name" : "Python [conda env:base] *"
        "language" : "python"
        "name" : "conda-base-py"
      }
      "language_info" : {
        "codemirror_mode" : {
          "name" : "ipython"
          "version" : 3
        }
        "file_extension" : ".py"
        "mimetype" : "text/x-python"
        "name" : "python"
        "nbconvert_exporter" : "python"
        "pygments_lexer" : "ipython3"
        "version" : "3.12.7"
      }
    }
    "nbformat" : 4
    "nbformat_minor" : 5
}
```