Sample Python Code Templates

---

1 data_preprocessing.py

```
import pandas as pd

def load_data ():
    inventory = pd.read_csv("data/device_inventory.csv")
    logs = pd.read_csv("data/maintenance_logs.csv")
    requests = pd.read_csv("data/service_requests.csv")
    return inventory, logs, requests

def clean_maintenance_logs(logs):
    logs['date'] = pd.to_datetime(logs['date'])
    logs['downtime_hours'] = logs['downtime_hours']. fillna (0)
    logs.drop_duplicates(inplace=True)
    return logs
```

---

2 kpi_calculations.py

```
import pandas as pd

def calculate_mtbf(logs):
    failure_logs = logs[logs["event_type"] == "Failure"]
    mtbf = failure_logs.groupby("device_id") ["date"]. diff ().dt.days
    return mtbf.mean()

def pm_compliance(logs):
    pm = logs[logs["event_type"] == "PM"]
    completed = pm[pm["status"] == "Completed"]. shape [0]
    total_pm = pm.shape[0]
    return round ((completed / total_pm) * 100, 2)
```

---

3 predictive_model.py

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import pandas as pd

def train_failure_model(df):
    X = df[["age_years", "last_service_days", "usage_hours"]]
    y = df["will_fail"]

    X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2)
```

```
    Model = RandomForestClassifier ()
    Model. Fit (X_train, y_train)

    preds = model.predict(X_test)
    print ("Accuracy:", accuracy score (y_test, preds))

    return model
```

---

dashboard_app.py (Streamlit)

```
import streamlit as st
import pandas as pd
import plotly.express as px
from src.kpi_calculations import pm_compliance

st.title("Medical Device Maintenance Dashboard")

df_logs = pd.read_csv("data/maintenance_logs.csv")

st.subheader("Preventive Maintenance Compliance")
st.metric("PM Compliance", f"{pm_compliance(df_logs)} %")

fig = px.histogram(df_logs, x="device_type", color="event_type")
st.plotly_chart(fig)
```