ML - Capstone Project

- Project Phases: Data Exploration and Understanding: Dive into the dataset to understand the landscape of laptop specifications. Visualize trends in laptop prices and identify potential influential features. Data

- Preprocessing: Handle missing values, outliers, and encode categorical variables. Ensure the dataset is ready for model training.

- Feature Engineering: Extract meaningful features to enhance model performance. Consider creating new features that capture the essence of laptop pricing. Model Development: Employ machine learning algorithms such as Linear Regression, Random Forest, and Gradient Boosting to predict laptop prices. Evaluate and choose the model that aligns best with the project's objectives.

- Hyperparameter Tuning: Fine-tune the selected model to achieve optimal performance. Real-time Predictions: Implement a mechanism for the model to make predictions for new laptops entering the market.

- Interpretability and Insights: Uncover insights into which features play a pivotal role in pricing decisions. Ensure that SmartTech Co. can interpret and trust the model's predictions.

1. **Introduction**

Objective: Develop a machine learning model to predict laptop prices based on various features.

Scope: Provide insights into factors influencing laptop prices and support SmartTech Co. in market positioning and strategy.

2. **Data Exploration and Understanding**

Dataset Overview: Describe the dataset, including key features and their types (e.g., categorical, numerical).

Exploratory Data Analysis (EDA): Share visualizations and statistics that highlight patterns, correlations, and distributions in the data.

3. **Data Preprocessing**

Handling Missing Values: Explain how missing values were addressed.

Feature Encoding: Detail the one-hot encoding process for categorical features.

Feature Scaling: Discuss any scaling techniques applied.

4. **Model Development**

Algorithms Used:

- **Linear Regression**: Explain its role and how it was implemented.
- **XGBoost**: Describe its usage and advantages over other algorithms.

Model Training: Discuss how the models were trained, including hyperparameter tuning.

5. **Model Performance**

Evaluation Metrics: Present metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared, and any other relevant metrics.

Comparison of Models: Compare the performance of Linear Regression and XGBoost.

6. **Insights**

Feature Importance: Identify which features have the most significant impact on laptop prices.

Brand Influence: Analyze whether the brand significantly influences the price.

Specification Impact: Compare how well the model performs for high-end vs. budget laptops.

Performance on Lesser-Known Brands: Evaluate the model's accuracy for laptops from lesser-known brands.

New Releases: Discuss the model's performance on newly released laptops not in the training dataset.

7. **Challenges and Limitations**

Data Limitations: Address any gaps or limitations in the dataset.

Model Limitations: Discuss any challenges faced during model development and limitations in predictions.

8. **Feedback and Improvements**

Gather Feedback: Ask stakeholders for their insights and concerns about the model's performance and applicability.

Future Improvements: Propose potential improvements based on feedback and observed limitations.

Feature Encoding

```
from sklearn.preprocessing import OneHotEncoder
import pandas as pd
# Example for one-hot encoding categorical features
encoder = OneHotEncoder(drop='first')
encoded_features = encoder.fit_transform(data[['Brand', 'Category']]).toarray()
encoded_df = pd.DataFrame(encoded_features, columns=encoder.get_feature_names_out())
data = data.join(encoded_df).drop(['Brand', 'Category'], axis=1)
```

Model Training

```
•    from sklearn.linear_model import LinearRegression

•    from xgboost import XGBRegressor

•    from sklearn.model_selection import train_test_split

•    from sklearn.metrics import mean_squared_error, r2_score


# Split data

•    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


•    # Train Linear Regression model

•    lr_model = LinearRegression()

•    lr_model.fit(X_train, y_train)


# Train XGBoost model

•    xgb_model = XGBRegressor()

•    xgb_model.fit(X_train, y_train)

# Predictions

•    lr_predictions = lr_model.predict(X_test)

•    xgb_predictions = xgb_model.predict(X_test)


# Evaluation

•    lr_mse = mean_squared_error(y_test, lr_predictions)

•    xgb_mse = mean_squared_error(y_test, xgb_predictions)

•    lr_r2 = r2_score(y_test, lr_predictions)

•    xgb_r2 = r2_score(y_test, xgb_predictions)


•    print(f'Linear Regression MSE: {lr_mse}, R2: {lr_r2}')

•    print(f'XGBoost MSE: {xgb_mse}, R2: {xgb_r2}')
```

Final Presentation Tips

- **Visualizations**: Use charts and graphs to make your data and results more comprehensible.
- **Clarity**: Ensure that your explanations are clear and tailored to the audience's level of technical expertise.
- **Interaction**: Encourage questions and discussions to engage stakeholders and gather valuable feedback.

- This structure should help you deliver a comprehensive and insightful presentation to SmartTech Co.