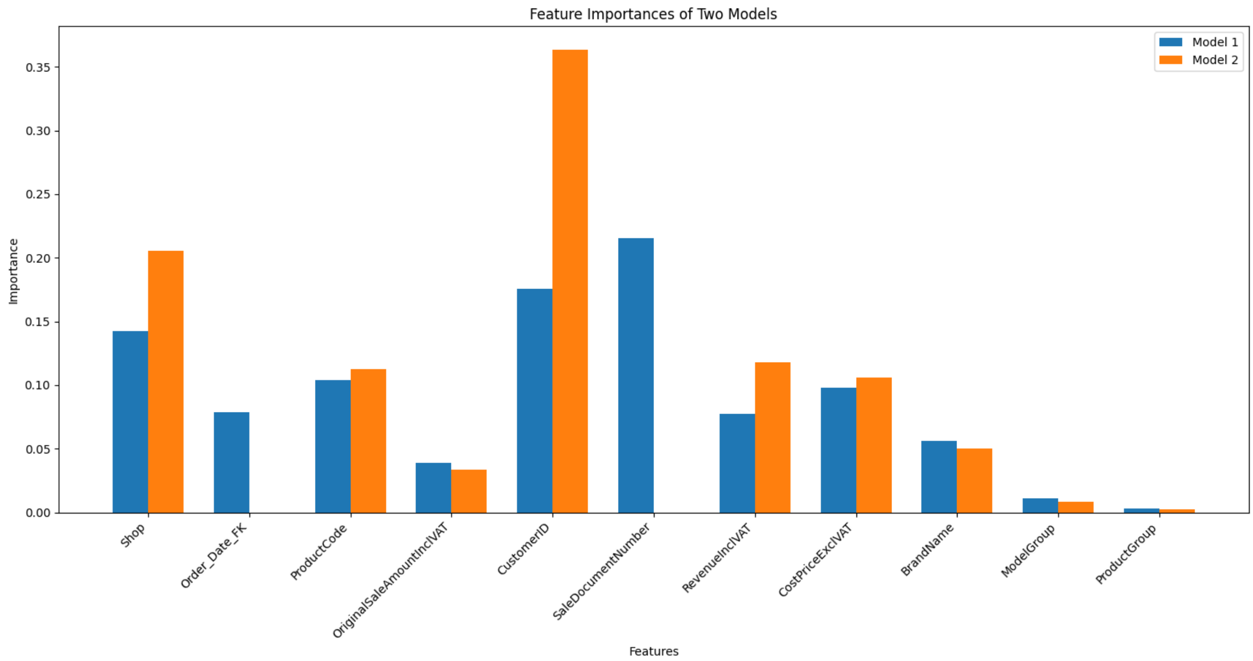
In [ ]:
```
import joblib

model1 = joblib.load('random_forest_model.pkl') #random forest with no no
model2 = joblib.load('random_forest_model2.pkl') #random forest with norm
```

In [ ]:
```
importances1 = model1.feature_importances_
importances2 = model2.feature_importances_

# You might want to plot these for a visual comparison
```

In [ ]:
```
import matplotlib.pyplot as plt
```

In [ ]:
```
import numpy as np

# Number of features (assuming both models have the same number of featur
n_features = len(importances1)

# Create an array with the positions of each bar along the x-axis
x_values = np.arange(n_features)
```

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

feature_names = ['Shop', 'Order_Date_FK', 'ProductCode', 'OriginalSaleAmo

n_features = len(importances1)
x_values = np.arange(n_features)
bar_width = 0.35

plt.figure(figsize=(15, 8))
plt.bar(x_values, importances1, width=bar_width, label='Model 1')
plt.bar(x_values + bar_width, importances2, width=bar_width, label='Model

plt.xlabel('Features')
plt.ylabel('Importance')
plt.title('Feature Importances of Two Models')
plt.xticks(x_values + bar_width / 2, feature_names, rotation=45, ha='righ

# Adjust the bottom margin to make room for the diagonal labels
plt.subplots_adjust(bottom=0.2)

plt.legend()
plt.tight_layout()  # Automatically adjust subplot params to give specifi
plt.show()
```



Comparing the feature importances of two different models, labeled "Model 1" (random forest with no normalization) and "Model 2" (random forest with normalization). Feature importance in this context measures the relative contribution of each feature to the model's predictions. Here are some observations:

1. **Dominant Features**: The feature `RevenueIncVAT` stands out prominently for both models, indicating that it has a significant impact on the model's predictions. This suggests that the revenue including VAT is a strong predictor in your dataset.

2. **Comparative Importance**: There's a visible difference in the importance of `SaleDocumentNumber` between the two models. It's quite important for Model 1 but much less so for Model 2. This could mean that Model 1 relies more on this feature for making predictions or that it has captured a pattern associated with this feature that Model 2 has not.

3. **Consistency Across Models**: Several features, such as `Shop`, `Order_Date_FK`, `ProductCode`, and `CustomerID`, show roughly similar levels of importance in both models, indicating a level of agreement between the models on the relevance of these features.

4. **Less Influential Features**: Towards the right of the chart, features like `BrandName`, `ModelGroup`, and `ProductGroup` have very low importance in both models. This could suggest that these features do not contribute significantly to the predictive performance of the models or that the information they contain is also captured by other features.

5. **Model Differences**: The overall pattern of feature importance between the two

models is similar for some features but differs for others. This could be due to a variety of reasons, such as different hyperparameter settings, different handling of categorical variables, or different interactions being captured between features in each model.

6. **Potential Overfitting**: Without knowing the model specifics, if a feature has a very high importance, it's worth investigating to ensure the model is not overfitting to this particular feature. Overfitting can happen if a feature correlates too well with the target only in the training set but not in unseen data.